Eye-Gaze Based Virtual Keyboard and System Interaction for Disabled

PROJECT REPORT

Submitted by

Lekshmi S S

TKM18MCA018

to

the APJ Abdul Kalam Technological University

in partial fulfillment of the requirements for the award of the Degree

of

Master of Computer Applications

Department of Computer Applications

TKM COLLEGE OF ENGINEERING

KOLLAM

DECLARATION

I undersigned hereby declare that the project report EYE-GAZE BASED VIRTUAL KEYBOARD AND SYSTEM INTERACTION FOR THE DISABLED, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Prof. Fousia.M.Shamsudeen. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Kollam

22/06/2021 LEKSHMI S S

DEPARTMENT OF COMPUTER APPLICATIONS
TKM COLLEGE OF ENGINEERING

C E R T I F I C A T E

This is to certify that the report entitled **Eye-Gaze Based Virtual Keyboard and System Interaction forDisabled** is a bonafide record of the work submitted by **LEKSHMI S S (TKM18MCA018)** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications is a bonafide record of the project work carried out by her under my guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Internal Supervisor Head of the Department External Examiner

ACKNOWLEDGEMENT

First and foremost I thank GOD almighty and my parents for the success of this project. I owe sincere gratitude and heart full thanks to everyone who shared their precious time and knowledge for the successful completion of my project.

I am extremely grateful to **Prof. Dr. Nadera Beevi S**, Head of the Department, for providing me with best facilities.

I would like to thank my coordinator, **Prof. Fousia.M.Shamsudeen**, Department of Computer Applications, who motivated me throughout the work of my project.

I profusely thank all other faculty members in the department and all other members of TKM College of Engineering, for their guidance and inspirations throughout my course of study.

I owe my thanks to my friends and all others who have directly or indirectly helped me in the successful completion of this project.

Lekshmi S S

ABSTRACT

In our society we have people who have different physical disability. For example some of them have paralysis and cannot move their body parts except their head or sometimes only eyes. The system developed by us will support them to overcome those disability. With our system they will be able to interact with their eye gaze, eye blinks. They can also operate different tasks like - Keyboard Operating, Going a Video Conference, SMS/email sending, Playing Video and Audio ,Document reading and so on in the same manner. The user can operate the system with a simple GUI and in different mode. User can change the mode as per his/her need. Our system will make them able to interact with the System using their eyes and eye gaze movement.Used this we can intact with our system devices with the the help of eye movement.If only moving parts of any user is his/her eyes then he/she can use our system with only eye gaze and eye blinks.Gaze-based interaction describes the input of user commands by using an eye tracking system.Face Detection,Eye Detection ,Eye Blink Detection,Eye Gaze movement Detection are the initial steps of the project. In this work, we focus on camera-based eye tracking systems as they are less intrusive than other eye tracking approaches. Eye-based input has several advantages over voice-based and sensor-based solutions.

Contents

1 INTRODUCTION **1**

1.1 Problem Defenition 2

1.2 Objective of this Project 3

2 LITERATURE SURVEY **4**

2.1 Purpose of the Literature Review 4

2.2 Literature Review of Eye gaze and Key Board Creation 5

3 METHODOLOGY **9**

3.1 Proposed System 10

3.2 Design Steps 11

3.2.1 Face Detection 13

3.2.2 Eye Region Detection With haar Classifiers In OpenCV 14

3.2.3 Gaze Tracing With Circular Object 17

3.2.4 Deep Learning Methods 20

3.3 Software Requirement and Specification 25

3.3.1 Python 25

3.3.2 Anaconda 26

3.3.3 Jupyter 27

3.4 Python Package Requirements 28

3.4.1 Opencv 28

3.4.2 dlib 28

3.4.3 xlib 29

3.4.4 pyqt 29
3.4.5 imutils 29
3.4.6 keras 29
3.4.7 pyaudio 29
3.4.8 pyQt5 30

4 RESULTS AND DISCUSSIONS **31**
4.1 Testing Methods 31
4.1.1 Output 1 36
4.1.2 Output 2 37
4.1.3 Output 3 38
4.1.4 Output 4 40
4.1.5 Output 5 41

5 CONCLUSION **42**
5.1 Future Enhancement 43

REFERENCES **44**

List of Figures

3.1 Architecture of the system 12

3.2 Haar features 13

3.3 integral image calculation 14

3.4 Extended set of Haar-like features 15

3.5 Equation 1 15

3.6 Equation 2 and Image Representation 16

3.7 The circular object to detect the pupil 17

3.8 Equation 3 and 4 18

3.9 Equation to puplie movement 18

3.10 Left and right movement of the pupil 19

3.11 Up and down movement of the pupil 19

3.12 Eye blink motion detection 19

3.13 A regular 3-layer neural network and a regular convolutional neural network . . . 21

3.14 Eye model 22

3.15 Gaze point Calculation equations 23

3.16 Gaze position in Y-axis 23

3.17 Gaze position in y-axis 23

3.18 VGG architecture 24

4.1 Eye Open 32

4.2 Eye closed 32

4.3 Eye Detection 33

4.4 Left and Right movement of Eye 34

4.5 Blink Detection 34
4.6 The window which can be used to interact with the system 36
4.7 virtual Keyboard operaion 39
4.8 Video Conference Window 40
4.9 document Option 41

List of Tables

Chapter 1

INTRODUCTION

In this project, Designed and implemented a system that will use users' eye gaze, eye blinks movements.Gaze-based interaction describes the input of user commands by using an eye tracking system. In this work, we focus on camera-based eye tracking systems as they are less intrusive than other eye tracking approaches. Eye-based input has several advantages over voice-based and sensor-based solutions. The system has a task manager GUI which has functionality of Keyboard Operating, Going a Video Conference, SMS/email sending, Playing Video and Audio ,Document reading,Sending an Emergency Alert. There are people who have critical physical disability, some of them have paralysis, cannot move any body part except only eyes and/or head.This will make them able to interact with the System using their eyes and eye gaze movement.Used this we can intact with our system devices with the the help of eye movement.Using this we change the thinking of Disabled People.Because they always think that they can do anything without the help of an another person.Using this they can do some operation without depending other person.This is the main aim of the project.
Gaze-based interaction describes the input of user commands by using an eye tracking system. In this work, we focus on camera-based eye tracking systems as they are less intrusive than other eye tracking approaches. Eye-based input has several advantages over voice-based and sensor-based solutions.Gaze-based interaction describes the input of user commands by using an eye tracking system.The system Face Detection ,Eye detection,Eye Gaze Detection and Eye blink Detection will be done.For the present version of our system we developed a single Convolutional Neural Network (CNN) architecture that is a modified version of VGG-8 architecture for classifying all

four classes (left gaze, center gaze, right gaze, and blink). In deep learning, CNNs are a class of deep neural networks that are mostly applied to evaluate visual images . They are inspired by the organization of the visual cortex in the brain .Here used a haarcascade classifier model for eye gaze trained with 1000+ sample and for eye blinks. system we developed a single Convolutional Neural Network (CNN) architecture that is amodified version of VGG-8 architecture for classifying all four classes (left gaze, centergaze, right gaze, and blink).We used two separate model for left eye and right eye, finalresult is integrated with both of the results.A thresholed Value will be calculated here wich is the most important part of this section.Some Python Package Dependencies used here are keras,opencv,dlib,xlib,pyqt etc.We used PyQt5 framework to design our GUI for user. For styling our GUI we have used a library called QdarkStyle.Here first we do the Eye detection which will be done with the the help of opecv and face detection algorithm

1.1 Problem Defenition

- Gaze-based interaction describes the input of user commands by using an eye tracking system.
- The system Face Detection ,Eye detection,Eye Gaze Dtection and Eye blink Detection will be done.
- Face Detection,Eye Detection ,Eye Blink Detection,Eye Gaze movement Detection are the initial steps of the project.
- For the present version of our system we developed a single Convolutional Neural Network (CNN) architecture that is a modified version of VGG-8 architecture for classifying all four classes (left gaze, center gaze, right gaze, and blink). .

1.2 Objective of this Project

The main aim of the project is to will make them able to interact with the System using their eyes and eye gaze movement.

- Face Detection,Eye Detection ,Eye Blink Detection,Eye Gaze movement Detection its direction identification are the initial steps done here.
- We had used a haarcascade classifier model for eye gaze trained with 1000+ sample and for eye blinks .
- system we developed a single Convolutional Neural Network (CNN) architecture that is a modified version of VGG-8 architecture for classifying all four classes (left gaze, center gaze, right gaze, and blink).We used two separate model for left eye and right eye, final result is integrated with both of the results.
- Some Python Package Dependencies used here are keras,opencv,dlib,xlib,pyqt etc.

Chapter 2

LITERATURE SURVEY

Literature review is the comprehensive study and interpretation of literature that relates to a particular topic. When one uses literature review research questions are identified, then one seek to answer this research questions by searching for and analyzing relevant literature. Some importance of literature reviews is that new insights can be developed by the re-analyzing the results of the study. A literature review is both a summary and explanation of the complete and current state of knowledge on a topic as found in academic books and journal articles. There are two kinds of literature reviews you might write at university: one that students are asked to write as a stand-alone assignment in a course, and the other that is written as part of an introduction to, or preparation for, a longer work, usually a thesis or research report. The focus and perspective of your review and the kind of hypothesis or thesis argument you make will be determined by what kind of review you are writing. One way to understand the differences between these two types is to read published literature reviews or the first chapters of theses and dissertations in your own subject area. Analyses the structure of their arguments and note the way they address the issues.

2.1 Purpose of the Literature Review

1. Giving the idea about topics.If we read more papper about a particular topic that will be help us in a positive manner.

2. It provides an excellent starting point for researchers beginning to do research in a new area

by forcing them to summarize, evaluate, and compare original research in that specific area.

3. It ensures that researchers do not duplicate work that has already been done.
4. It can provide clues as to where future research is heading or recommend areas on which to focus.
5. It highlights the key findings.
6. It identifies inconsistencies, gaps and contradictions in the literature.
7. It provides a constructive analysis of the methodologies and approaches of other researchers.

2.2 Literature Review of Eye gaze and Key Board Creation

Here, I take some of the papers related to Eye Gaze detection,Virtual Keyboard Creation and computer interaction and study about these methods,

In[1], C. Li, C. Kim and J. Park et al. [16] This papper will give an idea about the indirect interface system in which general users assign computer instructions just through the gaze tracing, without mouse or keyboard. Here wich specify that aWe Web camera to replace the computer-input system. The face region and the eye region was extracted based on Haar classifier implemented on open source computer vision library (OpenCV). It controls mouse-moving by automatically affecting the position where eyesight focuses on, and simulates mouse-click by affecting blinking action. Also give an idea about virtual keyboard displayed on monitor to simulate the keyboard entry.

In[2],Ms. Saily Bhagat1, Ms. Tanvi Patil2 et al. It will be mainly used to understand the theory and idea about the design of iGaze that will assist the paralyzed people to write on screen by gazing at different directions :up, down, left, right based on the targeted key, which is advantageous in terms of cost as well as robustness. In this papper will employs a robust machine learning algorithm for gaze estimation wherein gaze direction is classified on the basis of datasets. An efficient gaze estimator has been trained, by recording a dataset which consists of eye gaze direction frames.The system introduces an economically feasible method of using the convolution neural networks (CNN) for human-machine interaction. And also enhance the accuracy by selecting the key using eye gaze direction and typing the same. The proposed pipeline shows that, eye gaze direction can be estimated and display content of the screen can be controlled accordingly

In[3], A. Ardakani, C. Condo et al.Here the authors will be the give the concept of the implementations of deep neural networks (DNNs), especially convolutional neural networks (CNNs), has dramatically increased, it is very useful one beacuse it give an idea about the eye and eye gaze will be captures internally. Here real-time action recognition and video/image classification systems, latency is of paramount importance.In this paper, will give an description about efficient computational method, which is inspired by a computational core of fully connected neural networks, to process convolutional layers of state-of-the-art deep CNNs within strict latency requirements. To this end, we implemented our method customized for VGG and VGG-based networks which have shown state-of-the-art performance on different classification/recognition data sets.

In[4], F. M. Sigit, E. M. Yuniarno, R. F. Rachmadi and A. Zaini et al. Here the machine learning to detect blinking eyes based on images running in realtime. Using that collect images of faces and eyes to build a dataset and separate this dataset to two label categories based on our target classifications, these labels are ”opened eyes” and ”closed eyes”.There is a little thing different at 10000 image dataset compared by those two existing datasets particularly at closed eyes class category. In closed eyes class category at 10000 image dataset contains two variations of closed eyes images, there are 4000 perfectly closed eyes images and 1000 half-closed eyes images. All of those datasets are trained to convolutional neural network (CNN) so we have 3 different pretrained CNNs. Those three pretrained CNNs are tested to detect blinking eyes of samples running in realtime.

In[5], H. Cecotti et al.A New portable and noninvasive eye-trackers allow the creation of robust virtual keyboards that aim to improve the life of disabled people who are unable to communicate. A novel multimodal virtual keyboard and evaluates the performance changes that occur with the use of different modalities. The virtual keyboard is based on a menu selection with eight main commands that allow us to spell 30 different characters and correct errors with a delete button. It will will help the Virtual Keyboard Creation time.give an idea about how to set virtual keyboard frame work and buttons on it.

In[6],K. Goyal, K. Agarwal and R. Kumar et al.An application for tracking and detecting faces in videos and in cameras which can be used for multipurpose activities. The intention of the paper is deep study of face detection using open CV. A tabular comparison is performed in order to understand the algorithms in an easier manner. It talks about various algorithms like Adaboost, Haar cascades. This paper aims to help in understanding the best prerequisites for face detection.

In[7], W. Yu, J. Xiu, C. Liu and Z. Yang et al,In this paper, a face detection algorithm based on AdaBoost algorithm combined with Haar classifier depth cascade about OpenCV is proposed.It will be give an deep idea about the algorithm and its internal working. In order to solve the problem that the training depth of AdaBoost algorithm is not enough and the Haar classifier is not accurate because of the lack of depth cascade classification. Before the face feature extraction, AdaBoost algorithm is used to train the face samples to train the strong classifier. Then using OpenCV Haar classifier to depth classification for these strong classifier, it equal to cascade the AdaBoost algorithm, using high efficiency characteristic matrix and integral image method to accelerate the extraction rate of image feature values of the Haar-like.

In[8],H. Qassim, A et al. Deep learning has given way to a new era of machine learning, apart from computer vision. Convolutional neural networks have been implemented in image classification, segmentation and object detection. Despite recent advancements, we are still in the very early stages and have yet to settle on best practices for network architecture in terms of deep design, small in size and a short training time. In this paper, we address the issue of speed and size by proposing a compressed convolutional neural network model namely Residual Squeeze VGG16.

In[9], C. Meng and X. Zhao et al. Due to its low price, webcam has become one of the most promising sensors with the rapid development of computer vision. However, the accuracies of eye tracking and eye movement analysis are largely limited by the quality of the webcam videos.

To solve this issue, a novel eye movement analysis model is proposed based on five eye feature points rather than a single point (such as the iris center). First, a single convolutional neural network (CNN) is trained for eye feature point detection, and five eye feature points are detected for obtaining more useful eye movement information. Subsequently, six types of original time-varying eye movement signals can be constructed by feature points of each frame, which can reduce the dependency of the iris center in low quality videos. Finally, behaviors-CNN can be trained by the timevarying eye movement signals for recognizing different eye movement patterns, which is capable of avoiding the influence of errors from the basic eye movement type detection and artificial eye movement feature construction. To validate the performance, a webcam-based visual activity data set was constructed, which contained almost 0.5 million frames collected from 38 subjects. The experimental results on this database have demonstrated that the proposed model can obtain promising results for natural and convenient eye movement-based applications.

In this proposed work I Will more study about the eye gaze detection,threshold value calculation,EAR(Eye Aspect Ratio) calculation.Because We had used a haarcascade classifier model for eye gaze trained with 1000+ sample and for eye blinks we got an EAR value from pre-trained facial landmark recognizer with dlib and determined left blink, right blink and both blink. For the present version of our system we developed a single Convolutional Neural Network (CNN) architecture that is a modified version of VGG-8 architecture for classifying all four classes (left gaze, center gaze, right gaze, and blink).

Chapter 3

METHODOLOGY

In our society, there are people who have critical physical disability, some of them have paralysis, cannot move any body part except only eyes and/or head. They will always think that they can do anything without a another person.In this system will make those people to intract the system without the help of a third party.

Before we goes through our system operations we will first check what are the normal ways to interact with the system.Some of them are

Mouse based Interaction: It is the most comment type of interaction We all are used to interact with the system. For normal people it is possible but for physically disabled people it is a difficult one.

Voice based Interaction: Voice-based interaction is applied in a small set of use cases. The most prominent example is a situation where car drivers cannot or must not control a mobile device with their fingers but need an alternative way to answer a phone call or give instructions to a navigation system. In this situation, voice input seems to be the optimal choice as it does not force the driver to look away from the street. However, for scenarios except driving a car, voice-input often is not a feasible option:

Some of the disadvantages are

1. Background Noice
2. Quiet Zones

3. Privacy
4. Disabilities

Sensor based Interaction : Nowadays, mobile devices come with multiple sensors on board that provide information about status changes of the device. Received signals can be interpreted as user input. However, the usability of these approaches as primary input for people with disabilities is limited as they require full control over hand movements and are restricted to a finite set of gestures

3.1 Proposed System

In this proposed system developed a virtual keyboard using eye gaze system that is displayed on monitor, so user could type some words or simple sentence. Eye gaze is a natural interaction for every person that is so easily to do. The system consist of a virtual keyboard application that is operated using eye gaze of the user. So, with this system they can give some information to the others easily. In this system, there is a pointer that is moved using eye gaze of the users. So, the users should moves their eye gaze to choose the letter that they want. Our system will make them able to operate the Virtual Keyboard and some system operations using the gaze movement and eye blinking Work

In the proposed system, introducing the eye based interaction that is
Gaze-based interaction : describes the input of user commands by using an eye tracking system. In this work, we focus on camera-based eye tracking systems as they are less intrusive than other eye tracking approaches. Eye-based input has several advantages over voice-based and sensor-based solutions.
Some of the advantages are

1. Availability

   Eye-based input does not require any additional hardware. Most modern mobile devices come with a front camera that provides images in several mega-pixel resolution which is sufficient to precisely detect the pupil. Eye input is available in most situations as - in contrast to speech-input other people in close proximity are not disturbed.

2. Accessibility

   Physical disabilities often also affect hands and the articulation of a person. However, eye muscles are less often affected. Thus, eye-based input is an option for persons with spinal chord injuries and speech disorders.

3. Privacy

   Eye-based input protects the user's privacy. Even though eye-movements could be observed by a third person, they do not have a meaning by itself. An observer would require both, the eye-movements and the current menu options on the display to make sense of eye-based input.

4. Performance

   Even though the performance of smartphones and tablets has multiplied over the past years, they fall short of the computational capabilities of static devices. Especially as the computation intensive image processing in eye-tracking needs to happen in real-time, mobile devices easily reach their limits. This results in a lower frame rate which causes less accurate eye-tracking and eventually a less responsive interaction system.

5. Position:

   As mobile devices can be carried around, held in different positions, and be used during motion there is a large variety in the alignment of the device and the user's face. This makes it harder for eye-tracking algorithms to estimate the correct gaze position. Moreover, a calibration phase might not solve the problem as distance and angle between face and device might change at any time.

3.2 Design Steps

In this system we designed a system to provide the writing power without hand. Users will be able to write by their eyes. To construct the system we have to use a webcam to capture live video.

From the video, the system will detect face and eye using facial landmark structure. The system will be built on several parts as dace detection, eye detection, eye gaze and movement detection, eye blinking detection, virtual keyboard on screen, select left, right part of the keyboard and finally write using eye blinking.
Some of the Steps will be done wich lead to the completion of the entire System

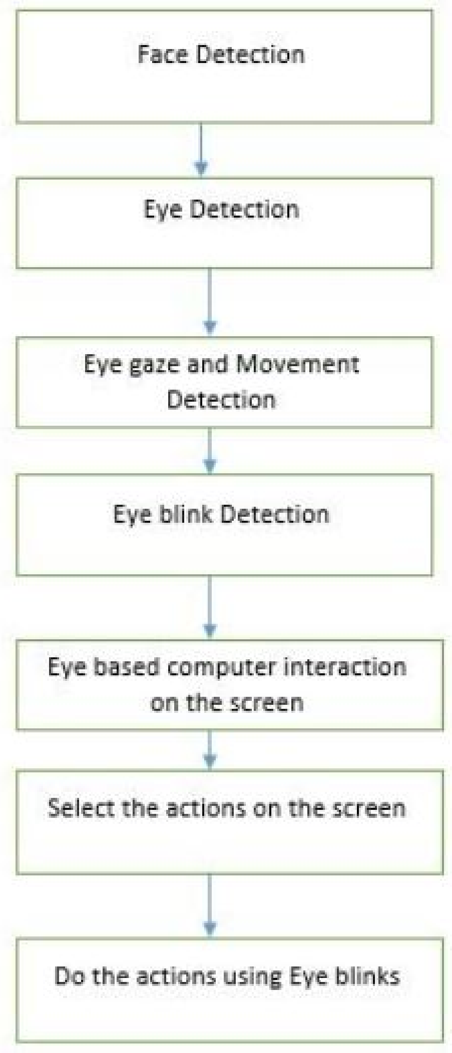

Face Detection
Eye Detection
Eye gaze and Movement Detection
Eye blink Detection
Eye based computer interaction on the screen
Select the actions on the screen
Do the actions using Eye blinks

Figure 3.1: Architecture of the system

3.2.1 Face Detection

Face detection is a process of locatingaa face inside an image frame, regardless of the identity of that face. Before recognizing a face,its first essential to detect and extract the faces from the original pictures. Face Detection target on finding the faces (area and size) in an image and probably extract them to be used by the face recognition algorithm.

Viola-Jones Algorithm

Developed by Paul Viola and Michael Jones back in 2001, the Viola-Jones Object Detection Framework can quickly and accurately detect objects in images and works particularly well with the human face.

The Viola Jones algorithm has four main steps

- Selecting Haar-like Features

  A Haar-like feature consists of dark regions and light regions. It produces a single value by taking the sum of the intensities of the light regions and subtract that by the sum of the intensities of dark regions. There are many different types of Haar-like features but the Viola-Jones Object Detection Framework only uses the ones in Figure 1

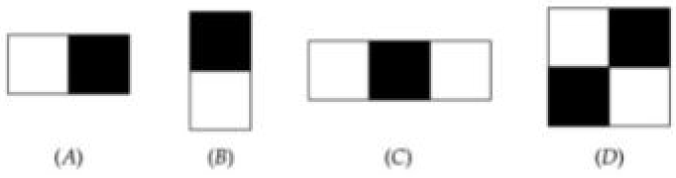


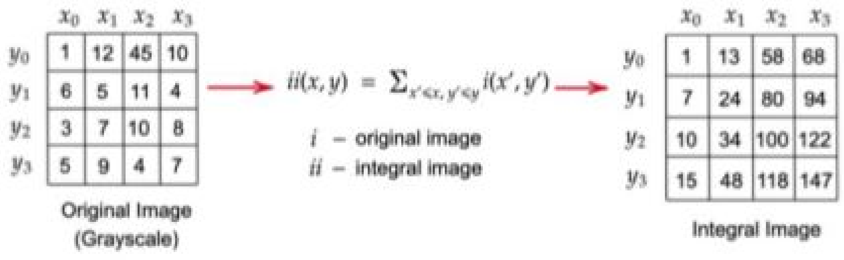
Figure 3.2: Haar features

- Creating an integral image

  An Integral Image is an intermediate representation of an image where the value for location (x, y) on the integral image equals the sum of the pixels above and to the left (inclusive) of the (x, y) location on the original image.This intermediate representation is essential because it allows for fast calculation of rectangular region,thus integral image reduces thet imeneeded to calculate the Haar-Feature values.

- Running AdaBoost traing
  The number of features that are present in the 24×24 detector window is nearly 160,000, but only a few of these features are important to identify a face. So we use the AdaBoost algorithm to identify the best features in the 160,000 features.

- Creating classifier cascades
  A Cascade Classifier is a multi-stage classifier that can perform detection quickly and accurately. Each stage consists of a strong classifier produced



	x_0	x_1	x_2	x_3
y_0	1	12	45	10
y_1	6	5	11	4
y_2	3	7	10	8
y_3	5	9	4	7

$$ii(x,y) = \Sigma_{x'<x,\, y'<y} i(x',y')$$

	x_0	x_1	x_2	x_3
y_0	1	13	58	68
y_1	7	24	80	94
y_2	10	34	100	122
y_3	15	48	118	147

Figure 3.3: integral image calculation

by the AdaBoost Algorithm. From one stage to another, the number of weak classifiers in a strong classifier increases. An input is evaluated on a sequential(stagebystage)basis. Ifaclassifier-foraspecificstageoutputsanegative result, the input is discarded immediately. In case the output is positive, the input is forwarded onto the next stage.

3.2.2 Eye Region Detection With haar Classifiers In OpenCV

Haar cascade classifier:

The core basis for Haar classifier object detection is the Haar-like features. These features, rather than using the intensity values of a pixel, use the change in contrast values between adjacent rectangular groups of pixels. The contrast variances between the pixel groups are used to determine relative light and dark areas. Two or three adjacent groups with a relative contrast variance formed the Haar-like features. Haar-like features, as shown in Figure 1 are used to detect an image. Haar features can easily be scaled by increasing or decreasing the size of the pixel group being examined. This allows features to be used to detect objects of various sizes.

The simple rectangular features of an image are calculated using an intermediate representation

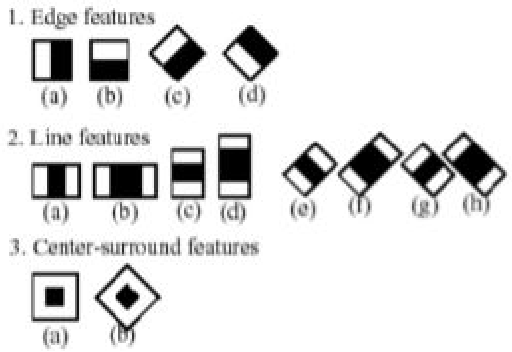


Figure 3.4: Extended set of Haar-like features

of an image, called the integral image. The integral image is an array containing the sums of the pixels' intensity values located directly to the left of a pixel and directly above the pixel at location (x, y) inclusive. So if AI[X, Y] is the original image and A[x, y] is the integral image then the integral image is computed as shown in equation 1 and illustrated in Figure 2.

$$AI[X,Y] = \sum_{x \le X, y \le Y} A(x,y) \qquad (1)$$

Figure 3.5: Equation 1

The features rotated by forty-five degrees, like the line feature, require another intermediate representation called the rotated integral image or rotated sum auxiliary image. The rotated integral image is calculated by finding the sum of the pixels' intensity values that are located at a forty five degree angle to the left and above for the x value and below for the y value. So if A[x, y] is the original image and AR[X, Y] is the rotated integral image then the integral image is computed as shown in equation 2 an illustrated in Figure 3. It only takes two passes to compute both integral

$$AR[X,Y] = \sum_{x \le X, y \le Y - |Y - y|} A(x,y) \,. \qquad (2)$$

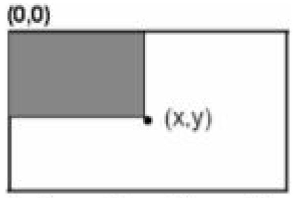


Figure 2. Summed area of integral image

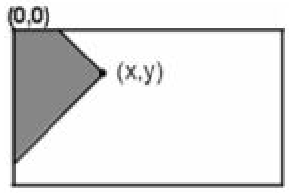


Figure 3. Summed area of rotated integral image

Figure 3.6: Equation 2 and Image Representation

image arrays, one for each array. Using the appropriate integral image and taking the difference between six to eight array elements forming two or three connected rectangles, a feature of any scale can be computed. Thus calculating a feature is extremely fast and efficient. It also means calculating features of various sizes requires the same effort as a feature of only two or three pixels. The detection of various sizes of the same object requires the same amount of effort and time as objects of similar sizes since scaling requires no additional effort.

Customize the Harr cascade classifier in OpenCV:

The detection of human facial features, such as the mouth, eyes, and nose require that Haar classifier cascades to be trained first. In order to train the classifiers, this gentle AdaBoost algorithm and Haar feature algorithms must be implemented. Fortunately, Intel developed an open source library devoted to easing the implementation of computer vision related programs called Open Computer

Vision Library (OpenCV). The OpenCV library is designed to be used in conjunction with applications that pertain to the field of HCI, robotics, biometrics, image processing, and other areas where visualization. It also includes an implementation of Haar classifier detection and training.

3.2.3 Gaze Tracing With Circular Object

The adjustment of the pupil size and position:

Because of the upper and lower eyelid, etc., it is difficult to find the complete circular shape of the pupil at first. The circular object is defined to detecting the pupil as shown in Figure 5. The diameter of the circle is corresponding to the width of the pupil. Across the surrounding of the circular object, eight rectangular windows are aligned every 45 degrees. The values obtained by these windows are used for adjusting the position and size of the circular object. Extracted circular object is determined by the central coordinate and the diameter of a circle.

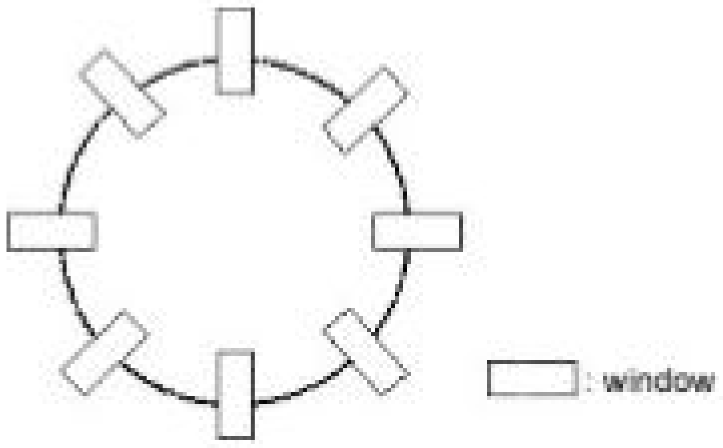


Figure 3.7: The circular object to detect the pupil

First, the energy formula of the windows is written as follows:

$$f_{\omega} = \frac{\text{sum of all pixel in window}}{\text{The number of pixel in window}} - \frac{255}{2} \quad (3)$$

The adjusting formula of the pupil size is written as follows:

$$f_{resize} = \int_0^{2\pi} f_{\omega}(\theta) d\theta \quad (4)$$

Figure 3.8: Equation 3 and 4

The formula takes a positive value from 0 to 180 degrees, and other values are negative. Therefore, the upper and lower position of the pupil is determined.

$$f_{move} = \int_0^{2\pi} f_{\omega}(\theta) \sin\theta d\theta .$$

Figure 3.9: Equation to puplie movement

The position control of the mouse pointer :

In order to use the pupil to control the mouse pointer (cursor) on the screen, the central coordinate of the screen is set as a start point. This position is used as the base for gaze tracing, and the initial position of the mouse pointer is set as the center of the screen. The moving position of the cursor takes the initial position as the base. As the pupil move to some direction, the coordinate of the mouse pointer on screen change according to the movement of the pupil. When the pupils return to the original position, the cursor stops moving.

The horizontal movement of the pupil can be fully grasped by the movement of the circular objects as shown in Figure 6. The vertical movement of the circular object is more subtle than the horizontal movement, so the size of the pupil is used for control as shown in Figure 7. When people look

Figure 3.10: Left and right movement of the pupil

upwards, the eyes are getting bigger. When looking downwards, the eyes are in slightly half-closed state. This phenomenon can be used for controlling the mouse pointer to move from top to bottom.

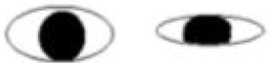


Figure 3.11: Up and down movement of the pupil

The mouse click is treated by perceiving the blink of an eye. Namely, when one person's eyes are recognized in closed state, we click the position of the current mouse pointer. The Figure 8 shows the calculus of the closed state of the eyes. In a certain region containing the pupils, we consider several vertical masks and measure the run length of the black and white pixels.

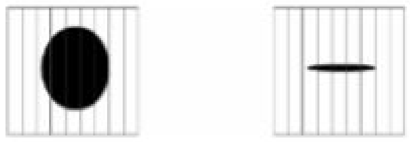


Figure 3.12: Eye blink motion detection

Namely, compared the black pixel with the white pixel on the mask and, when the black pixel is far less than the white pixel, we deem the eyes lying in closed state. When the left eye blinks, we click the left key of the mouse. When the right eye blinks, we click the right key of the mouse. Normally, when the errors of the human blinks occur, both eyes are closed together. At this time, we do not perform any actions. The action is only carried out with the closure of a single eye. IV

3.2.4 Deep Learning Methods

Deep learning is part of a broader family of machine learning methods based on artificial neural networks with representation learning.In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep neural network, most commonly applied to analyze visual imagery. ... CNNs are regularized versions of multilayer perceptrons.

For the present version of our system we developed a single Convolutional Neural Network (CNN) architecture that is a modified version of VGG-8 architecture for classifying all four classes (left gaze, center gaze, right gaze, and blink).We used two separate model for left eye and right eye, final result is integrated with both of the results.

convolutional neural network : In deep learning, CNNs are a class of deep neural networks that are mostly applied to evaluate visual images . They are inspired by the organization of the visual cortex in the brain . The visual cortex is the main region of the brain that receives and processes visual information transmitted from the eye [26]. As shown in Figure 2, CNNs are very similar to regular neural networks made up of different neurons with learnable weights and biases. Each neuron in the network accept an input, performs a dot product operation, and optionally follow the dot operation with a non-linearity operation (such as Rectified Linear Unit (ReLU)). CNN architectures make the explicit assumption that all inputs are images and thus allow users to encode certain properties into the architecture . Typically, a CNN consists of multiple convolutional layers followed by pooling, non-linearity, and finally fully connected layer(s) plus output layer. The first three layers are responsible for feature extraction, while the fully connected layer is responsible for classification.

With the recent rise of deep learning, Convolutional Neural Network (CNN) based gaze estimation models are becoming very popular and prevalent.CNN models are also suitable for handling large scale datasets, and they have been successfully applied to many different domains, such as computer vision , speech recognition , and language modelling . CNN models are capable of directly mapping image features, such as pupil and glint locations, to gaze points without the use of hand-engineered features.

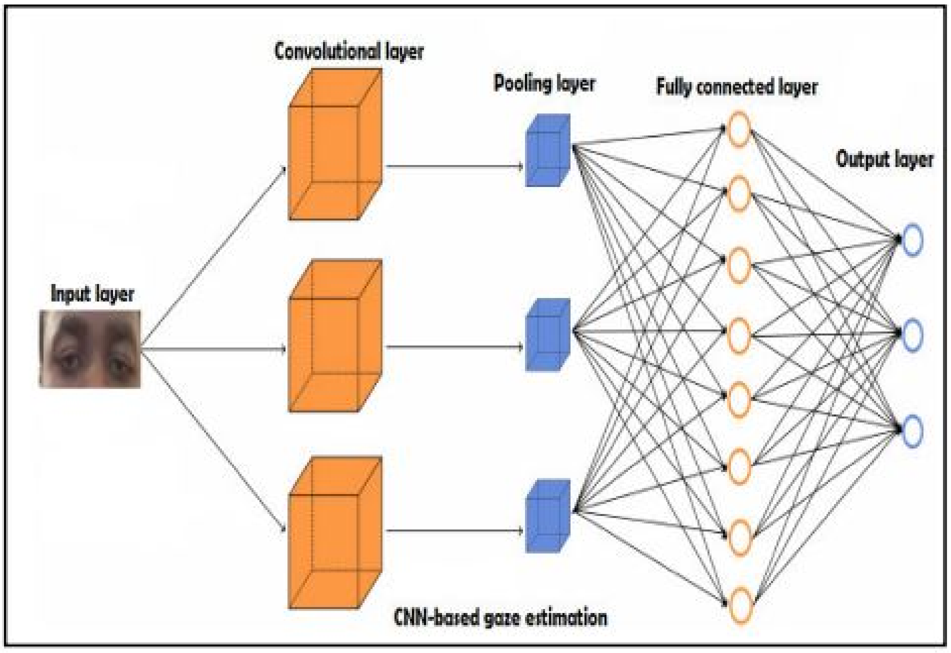


Figure 3.13: A regular 3-layer neural network and a regular convolutional neural network

Gaze estimation is a technology that aims to understand the intent and interest of users . Gaze estimation techniques focus on the relationship between the image data and gaze direction . Gaze directions are estimated based on specific eye features (such as pupil and corneal reflection) extracted from the eye regions of image data collected from single or multiple cameras . Generally, gaze estimation techniques can be grouped into two: model-based and appearance-based techniques . More information on the two techniques is provided in the next sub-sections.

APPEARANCE-BASED GAZE ESTIMATION:

Appearance-based techniques rely on the photometric appearance of the eye to perform gaze estimation. They generally need a single camera to capture eye images which are then used to build gaze estimation models that can map the appearances of the captured images to certain gaze directions. These models do not require hand-engineered features for training; they can extract image features implicitly from the data. Compared to model-based techniques, appearancebased techniques require a larger number of eye images for training, and this gives them the capacity to learn invariance in appearance disparity .

MODEL-BASED GAZE ESTIMATION:

Model-based techniques perform gaze estimation by combining the geometric model of the eye with eye features, such as cornea reflection and pupil centre.The gaze direction is determined by the visual axis, which goes through the cornea centre and the fovea. The point of regard (PoR) can be defined as the intersection between the visual axis and the display surface [1]. The angle kappa is defined as the angle between the visual axis and the optical axis.

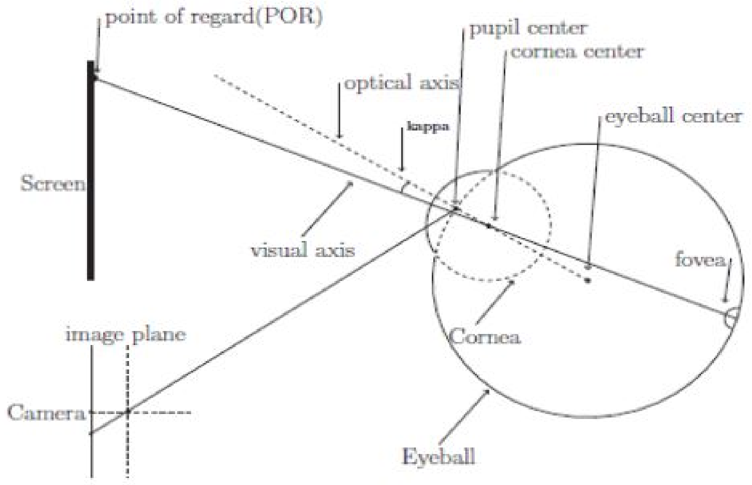


Figure 3.14: Eye model

GAZE POINT COORDINATES IN PIXELS:

The calculation for the gaze point coordinates in pixels can be obtained from (1) and (2) where

$$Gaze\_X = mean\left(\frac{x_{left} + x_{right}}{2}\right) \qquad (1)$$

$$Gaze\_Y = mean\left(\frac{Y_{left} + Y_{right}}{2}\right) \qquad (2)$$

Figure 3.15: Gaze point Calculation equations

(Xleft, Yleft, Xright, Yright) represents the actual gaze coordinates of the left and right eye as obtained from the eye tracker.

GAZE POSITION IN mm:

The gaze position in mm of on-screen distance can be calculated by using (3) to (4):

$$X\_Position(mm) = \mu * Gaze\_X \qquad (3)$$

Figure 3.16: Gaze position in Y-axis

$$Y\_Position(mm) = \mu * Gaze\_Y \qquad (4)$$

Figure 3.17: Gaze position in y-axis

VGG Architecture VGGNets and VGG-like networks consist of a stack ofconvolutional layers followed by a small number of fully-connected layers. The size of filters and stride are fixed to the receptive field of 3×3 and one pixel for the convolutional processes, respectively. A one-pixel convolutional padding is used to preserve the spatial resolution after convolution and all layers are

equipped with the linear rectifier (i.e. ReLU) as their non-linear activation function.
For the present version of our system we developed a single Convolutional Neural Network (CNN) architecture that is a modified version of VGG-8 architecture for classifying all four classes (left gaze, center gaze, right gaze, and blink).We used two separate model for left eye and right eye, final result is integrated with both of the results. We trained our model with 6000+ sample and got 97 percentage.

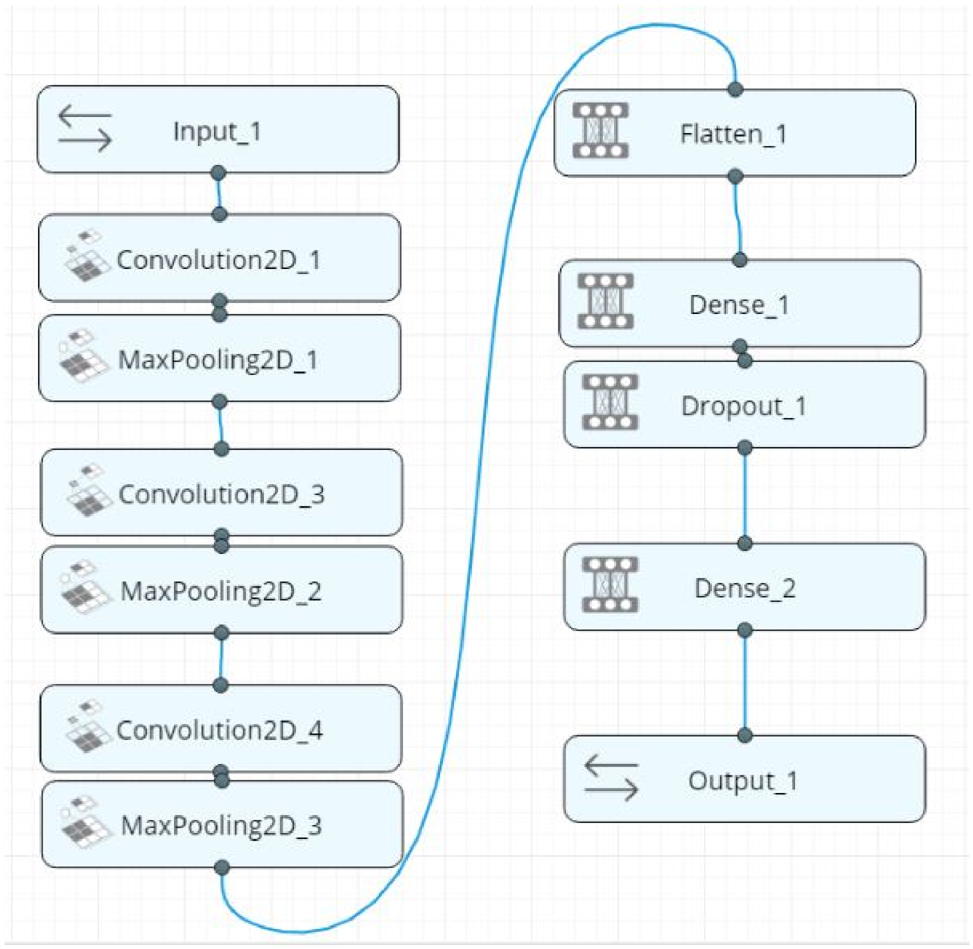


Figure 3.18: VGG architecture

3.3 Software Requirement and Specification

The software used for the project:

- Python
- Anaconda
- Jupyter

3.3.1 Python

Python is an object-oriented programming language created by Guido Rossum in 1989. It's ideally designed for fast prototyping of complicated applications. It has interfaces to several OS system calls and libraries and is protractile to C or C++. several massive corporations use the Python programming language embody NASA, Google, YouTube, BitTorrent, etc. Python programming is widely utilized in AI, natural language Generation, Neural Networks and other advanced fields of computer science. Python is programming language open supply, high-level artificial language developed by Guido van Rossum within the late Eighties and presently administered by Python Software Foundation. It came from the ABC language that he helped produce early on in his career. Python is a powerful language that you can use develop games, write GUIs, and develop web applications. It's a high-level language. Reading and writing codes in Python is far like reading and writing regular English statements. As a result, they're not written in the machine-readable language, Python programs got to be processed before machines can run them. Python is an understood language. This implies that each time a program is run, its interpreter runs through the code and interprets it into machine-readable byte code. Python is an object-oriented language control users to manage and management data structures or objects to make and run programs. Everything in Python is, in fact, top-notch. All objects, data types, functions, methods, and classes take an equal position in Python. Programming languages are created to satisfy the requirements of programmers and users for an efficient tool to develop applications that impact lives, lifestyles, economy, and society. they assist build lives better by increasing productivity, enhancing communication, and rising potency. Languages die and become obsolete once they fail to live up to expectations and are replaced and superseded by languages that are more powerful. Python programming language artificial language that has stood the test of time and has remained relevant across industries and businesses and among programmers, and individual users. It's a living, thriving, and extremely helpful language that's extremely recommended as a primary programming language for those that want to dive into and experience programming.

3.3.2 Anaconda

Anaconda is a Python distribution (prebuilt and preconfigured collection of packages) that is commonly used for data science. The Anaconda distribution includes the Conda package manager in addition to the preconfigured Python packages and other tools. Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Package versions are managed by the package management system conda. The Anaconda distribution includes data-science packages suitable for Windows, Linux, and MacOS. Anaconda distribution comes with more than 1,500 packages as well as the conda package and virtual environment manager. It also includes a GUI, Anaconda Navigator], as a graphical alternative to the command line interface (CLI). Big difference between conda and the pip package manager is in how package dependencies are managed, which is a significant challenge for Python data science and the reason conda exists. When pip installs a package, it automatically installs any dependent Python packages without checking if these conflict with previously installed packages. It will install a package and any of its dependencies regardless of the state of the existing installation. Because of this, a user with a working installation of, for example, Google Tensorflow, can find that it stops working having used pip to install a different package that requires a different version of the dependent numpy library than the one used by Tensorflow.

In some cases, the package may appear to work but produce different results in detail.In contrast, conda analyses the current environment including everything currently installed, and, together with any version limitations specified, works out how to install a compatible set of dependencies, warning if this cannot be done.Open source packages can be individually installed from the Anaconda repository, Anaconda Cloud (anaconda.org), or your own private repository or mirror, using the conda install command. Anaconda Inc compiles and builds all the packages in the Anaconda repository itself, and provides binaries for Windows 32/64 bit, Linux 64 bit and MacOS 64-bit. Anything available on PyPI may be installed into a conda environment using pip, and conda will keep track of what it has installed itself and what pip has installed.Custom packages can be made using the conda build command, and can be shared with others by uploading them to Anaconda Cloud,PyPI or other repositories.The default installation of Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7. However, it is possible to create new environments that include any version of Python packaged with conda. Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them.

3.3.3 Jupyter

Anaconda Navigator is a GUI tool that is included in the Anaconda distribution and makes it easy to configure, install, and launch tools such as Jupyter Notebook. A Conda Python environment is an isolated environment. It allows you Project to install package swithout modifying your system's Python installation. .Jupyter is a suite of software products used in interactive computing. IPython was originally developed by Fernando Perez in 2001 as an enhanced Python interpreter. A web based interface to IPython terminal in the form of IPython notebook was introduced in 2011. In 2014, Project Jupyter started as a spin-off project from IPython. Some of the Packages under Jupyter project include

Jupyter notebook : A web based interface to programming environments of Python, Julia, R and many others.

QtConsole : Qt based terminal for Jupyter kernels similar to IPython

nbviewer : Facility to share Jupyter notebooks

JupyterLab : Modern web based integrated interface for all products.

3.4 Python Package Requirements

Some of the Python Packages Requirements are need in this project.They are:

- opencv
- dlib
- xlib
- pyqt
- imutils
- pyaudio
- qdarkstyle
- keras

3.4.1 Opencv

OpenCV (Open Source Computer Vision Library) includes several hundreds of computer vision algorithms. It has a modular structure,which means that the package includes several shared or static libraries.OpenCV is a popular library for Image processing and Computer Vision. Using python with OpenCV combines the simplicity of python with the capabilities of the versatile OpenCV library. ... Anaconda is the first choice distribution for scientific python particularly on Windows.

3.4.2 dlib

DLib is an open source C++ library implementing a variety of machine learning algorithms, including classification, regression, clustering, data transformation, and structured prediction. ... K-Means clustering, Bayesian Networks, and many others.

3.4.3 xlib

The Python X Library is intended to be a fully functional X client library for Python programs. It is written entirely in Python, in contrast to earlier X libraries for Python (the ancient X extension and the newer plxlib) which were interfaces to the C Xlib.

3.4.4 pyqt

PyQt is a module to make desktop software with Python. This works on all desktop systems including Mac OS X, Windows and Linux.

If you want to make desktop apps with Python, PyQt is the module you need to make them. After creating your app, you can create an installation program with fbs. You can easily make desktop software with PyQt. There are two ways to install PyQt: with an installer and from code.

Compling PyQt from source can be a tedious process, recommend you to install using the installer or package manager. (an end-user can simply run a setup program to install your software)

3.4.5 imutils

Imutils are a series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, and displaying Matplotlib images easier with OpenCV and both Python 2.7 and Python 3.

3.4.6 keras

Keras is an Open Source Neural Network library written in Python that runs on top of Theano or Tensorflow. It is designed to be modular, fast and easy to use. It was developed by François Chollet, a Google engineer. Keras doesn't handle low-level computation.

3.4.7 pyaudio

PyAudio provides Python bindings for PortAudio, the cross-platform audio I/O library. With PyAudio, you can easily use Python to play and record audio on a variety of platforms. PyAudio is inspired by: tkSnack: cross-platform sound toolkit for Tcl/Tk and Python.

3.4.8 pyQt5

Qt is set of cross-platform C++ libraries that implement high-level APIs for accessing many aspects of modern desktop and mobile systems. These include location and positioning services, multimedia, NFC and Bluetooth connectivity, a Chromium based web browser, as well as traditional UI development.

PyQt5 is a comprehensive set of Python bindings for Qt v5. It is implemented as more than 35 extension modules and enables Python to be used as an alternative application development language to C++ on all supported platforms including iOS and Android.

PyQt5 may also be embedded in C++ based applications to allow users of those applications to configure or enhance the functionality of those applications.

Chapter 4

RESULTS AND DISCUSSIONS

Testing is the major quality measures employed during the software development. After the coding phase, computer programs available are executed for testing purpose. Testing not only has to uncover errors introduced during coding, but also locates errors committed during the previous phase. Thus the aim of testing is to uncover requirements, design or coding errors in the program.

- Testing is a process of executing a program with the intention of finding an error.
- A good test case is one that has a highest probability of finding an as yet undiscovered error.
- A successful test is one that uncovers an as yet undiscovered error.

My objective is to design tests that systematically uncover different classes of errors and to do so with minimum amount of time and effort. Testing demonstrate that software functions appear to be working according to specification, that performance requirements appears to have been met. Data collected as testing is conducted provide a good indication of software reliability and some indication of software quality as a whole. But there is one thing that testing cannot do: Testing cannot show the absence of defects it can only show that software defects as present.

4.1 Testing Methods

There are different types of testing methods available.

1 **Unit Testing**

In this testing we test each module individually and integrate the overall system. Unit testing focuses verification efforts on the smaller unit of software design in the module. This is also known as “module” testing. The modules of the system are tested separately. The testing is carried out during programming stage itself.

we have 4 stages of tesing here,

- Stage 1:Here our main aim is to detect the eye,eye gaze and eye blinks.Becase based on the eye gaze movement we will we can choose different operation. After running the program web camera of the pc is opened. The first step of the system is face detection. So, face detection in real-time implemented using Haar like feature descriptor and to detect face and eye method with the help of dlib.
- Stage 2: After successfull completion of face detection To detect eye gaze we calculate eye gaze ratio from both left and right eye and calculate the Eye Aspect Ratio(EAR). We detect the screening point at which eyeball is looking. We can only define eye gaze as the center, right and left.

We draw Horizondal and Vertical line .When the eye open the lengh of the Vertical line is greater.When the eye will be closed the lengh of the vertical line become smaller

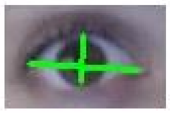


Figure 4.1: Eye Open

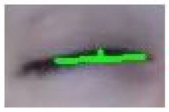


Figure 4.2: Eye closed

- Stage 3: The circular object is defined to detecting the pupil The diameter of the circle is corresponding to the width of the pupil. Across the surrounding of the circular object, eight rectangular windows are aligned every 45 degrees. The values obtained by these windows are used for adjusting the position and size of the circular object. Extracted circular object is determined by the central coordinate and the diameter of a circle.

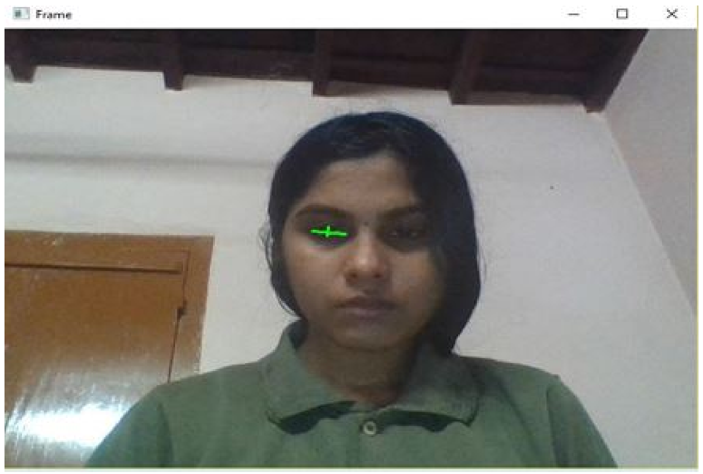


Figure 4.3: Eye Detection

- Stage 4:The next step will be detect the Eye blink. The main task of the system is typing which is done by eye blinking. Here the eye area is taken from the video stream and then we apply a threshold to detect eye-ball more accurately. In this system to detect eye blinking, we create two lines. One line crossing the eye horizontally and another line in vertically. In a certain region containing the pupils, we consider several vertical masks and measure the run length of the black and white pixel.compared the black pixel with the white pixel on the mask and, when the black pixel is far less than the white pixel, we deem the eyes lying in closed state. When the left eye blinks, we click the left key of the mouse. When the right eye blinks, we click the right key of the mouse. Normally, when the errors of the human blinks occur, both eyes are closed together. At this time, we do not perform any actions. The action is only carried out with the closure of a single eye.

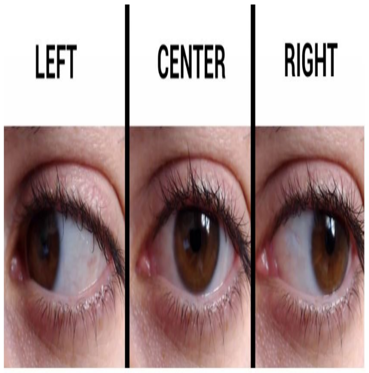


Figure 4.4: Left and Right movement of Eye

- **Integration Testing** Data can be lost across an interface. One module can have an adverse effect on the other sub functions when combined may not produce the desired major functions. Integrated testing is the systematic testing for constructing the uncover errors within the interface. This testing was done with sample data. The need for integrated test is to find the overall system performance.

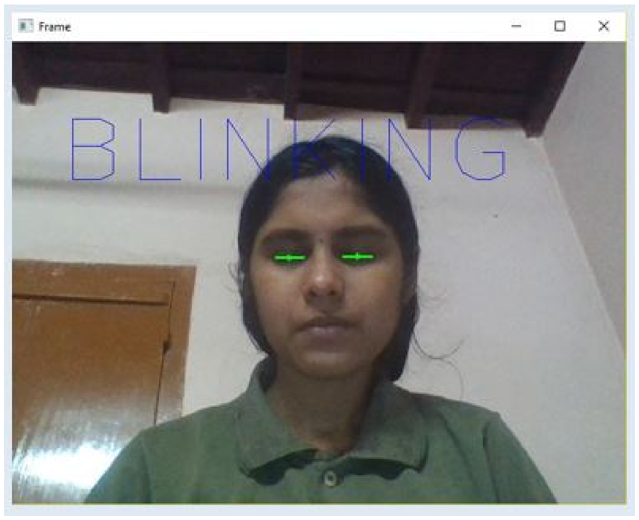


Figure 4.5: Blink Detection

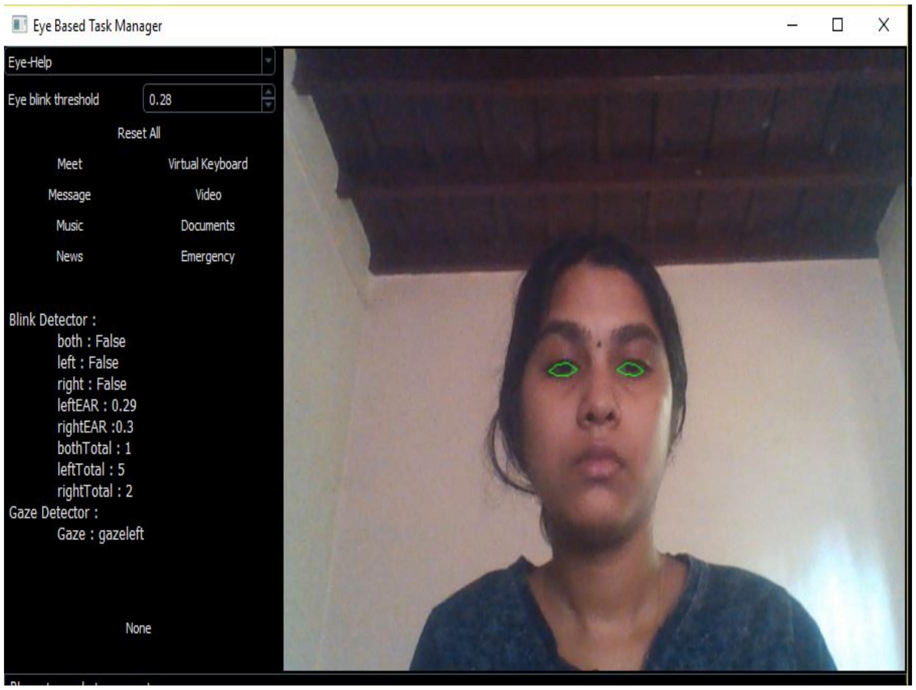
- Here sucessfully done face,eye,eye gaze and eye blink detection our main aim is to integrate it with Eye based computer interaction window.And perfome the operation.In that window we can perfoming the operation.in that we using the We used PyQt5 framework to design our GUI for user. For styling our GUI we have used a library called QdarkStyle

- **Validation Testing**

  At the culmination of black box testing, software is completely assembled as a pack- age, interface errors have been uncovered and corrected and final series of software

  test, validation test begins. Validation testing can be defined in many ways but a simple.definition is the validation succeeds when the software functions in a manner that can be reasonably accepted by the customer.After validation test have been conducted one of the two possible conditions exists. The function or performance characteristics confirmed to the specification and are accepted. A deviation from specification is uncovered and a deficiency list is created.

4.1.1 Output 1



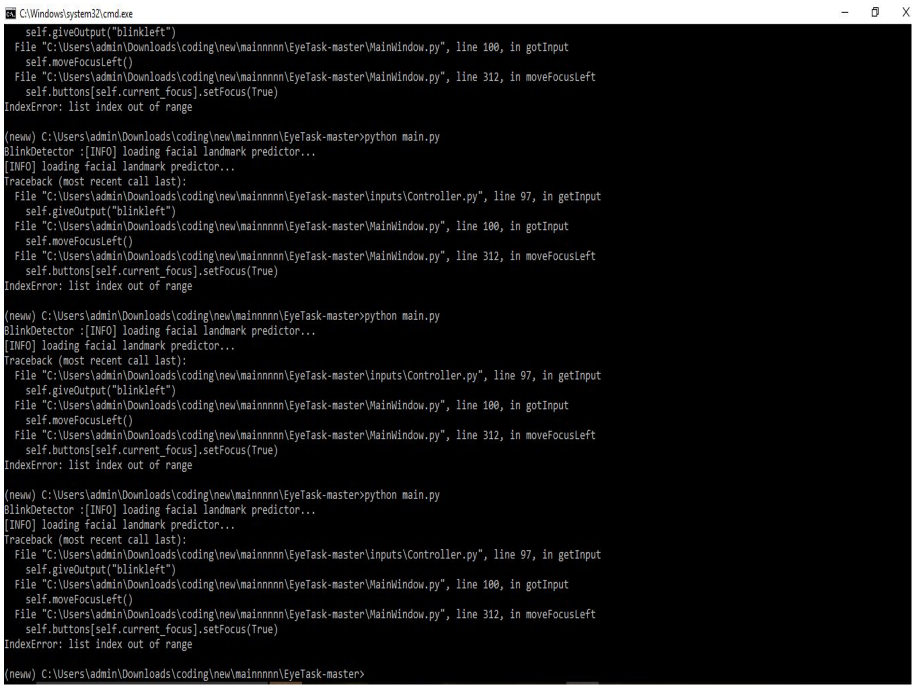
Eye blink threshold 0.28

```
Blink Detector :
        both : False
        left : False
        right : False
        leftEAR : 0.29
        rightEAR :0.3
        bothTotal : 1
        leftTotal : 5
        rightTotal : 2
Gaze Detector :
        Gaze : gazeleft
```

Figure 4.6: The window which can be used to interact with the system

4.1.2 Output 2

In the fig shown that the output of what happen inside the System. [h]

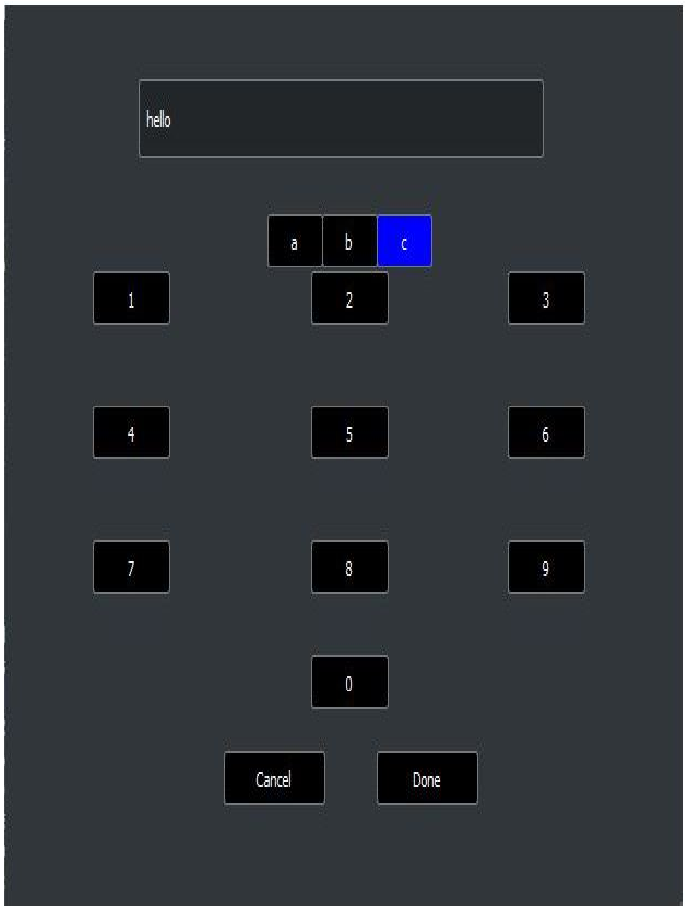


```
C:\Windows\system32\cmd.exe
    self.giveOutput("blinkleft")
  File "C:\Users\admin\Downloads\coding\new\mainnnnn\EyeTask-master\MainWindow.py", line 100, in gotInput
    self.moveFocusLeft()
  File "C:\Users\admin\Downloads\coding\new\mainnnnn\EyeTask-master\MainWindow.py", line 312, in moveFocusLeft
    self.buttons[self.current_focus].setFocus(True)
IndexError: list index out of range

(neww) C:\Users\admin\Downloads\coding\new\mainnnnn\EyeTask-master>python main.py
BlinkDetector :[INFO] loading facial landmark predictor...
[INFO] loading facial landmark predictor...
Traceback (most recent call last):
  File "C:\Users\admin\Downloads\coding\new\mainnnnn\EyeTask-master\inputs\Controller.py", line 97, in getInput
    self.giveOutput("blinkleft")
  File "C:\Users\admin\Downloads\coding\new\mainnnnn\EyeTask-master\MainWindow.py", line 100, in gotInput
    self.moveFocusLeft()
  File "C:\Users\admin\Downloads\coding\new\mainnnnn\EyeTask-master\MainWindow.py", line 312, in moveFocusLeft
    self.buttons[self.current_focus].setFocus(True)
IndexError: list index out of range

(neww) C:\Users\admin\Downloads\coding\new\mainnnnn\EyeTask-master>python main.py
BlinkDetector :[INFO] loading facial landmark predictor...
[INFO] loading facial landmark predictor...
Traceback (most recent call last):
  File "C:\Users\admin\Downloads\coding\new\mainnnnn\EyeTask-master\inputs\Controller.py", line 97, in getInput
    self.giveOutput("blinkleft")
  File "C:\Users\admin\Downloads\coding\new\mainnnnn\EyeTask-master\MainWindow.py", line 100, in gotInput
    self.moveFocusLeft()
  File "C:\Users\admin\Downloads\coding\new\mainnnnn\EyeTask-master\MainWindow.py", line 312, in moveFocusLeft
    self.buttons[self.current_focus].setFocus(True)
IndexError: list index out of range

(neww) C:\Users\admin\Downloads\coding\new\mainnnnn\EyeTask-master>python main.py
BlinkDetector :[INFO] loading facial landmark predictor...
[INFO] loading facial landmark predictor...
Traceback (most recent call last):
  File "C:\Users\admin\Downloads\coding\new\mainnnnn\EyeTask-master\inputs\Controller.py", line 97, in getInput
    self.giveOutput("blinkleft")
  File "C:\Users\admin\Downloads\coding\new\mainnnnn\EyeTask-master\MainWindow.py", line 100, in gotInput
    self.moveFocusLeft()
  File "C:\Users\admin\Downloads\coding\new\mainnnnn\EyeTask-master\MainWindow.py", line 312, in moveFocusLeft
    self.buttons[self.current_focus].setFocus(True)
IndexError: list index out of range

(neww) C:\Users\admin\Downloads\coding\new\mainnnnn\EyeTask-master>
```

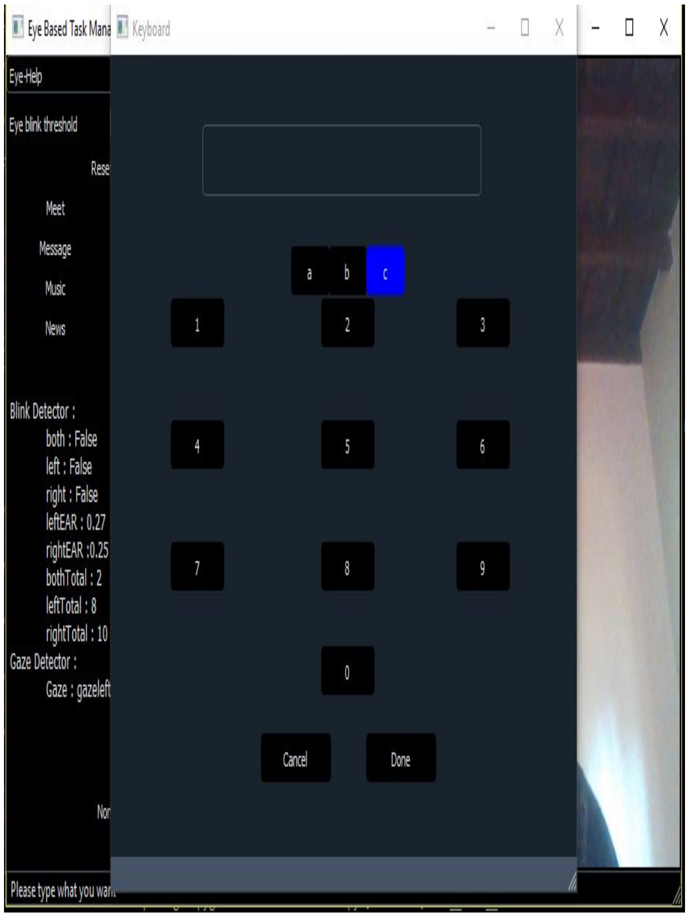
Checking for input phase

4.1.3 Output 3

User can write any message using a specially designed virtual keyboard which can be controlled using eye gaze and eye blinks. The keyboard has 12 main keys and every main keys has 2-4 floating keys. Main keys are navigated using eye blinks and the floating keys are navigated using eye gaze. [h]



virtual Keyboard

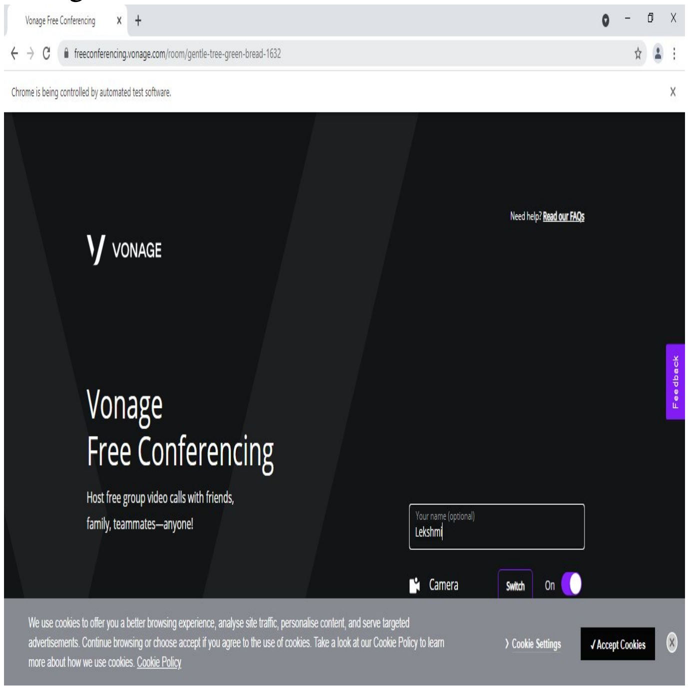


```
Blink Detector :
        both : False
        left : False
        right : False
        leftEAR : 0.27
        rightEAR :0.25
        bothTotal : 2
        leftTotal : 8
        rightTotal : 10
Gaze Detector :
        Gaze : gazeleft
```

Figure 4.7: virtual Keyboard operaion

4.1.4 Output 4

When eye clicking Meeting button a Video Conference window will be automatically opened.



Chrome is being controlled by automated test software.

Vonage
Free Conferencing

Host free group video calls with friends, family, teammates—anyone!

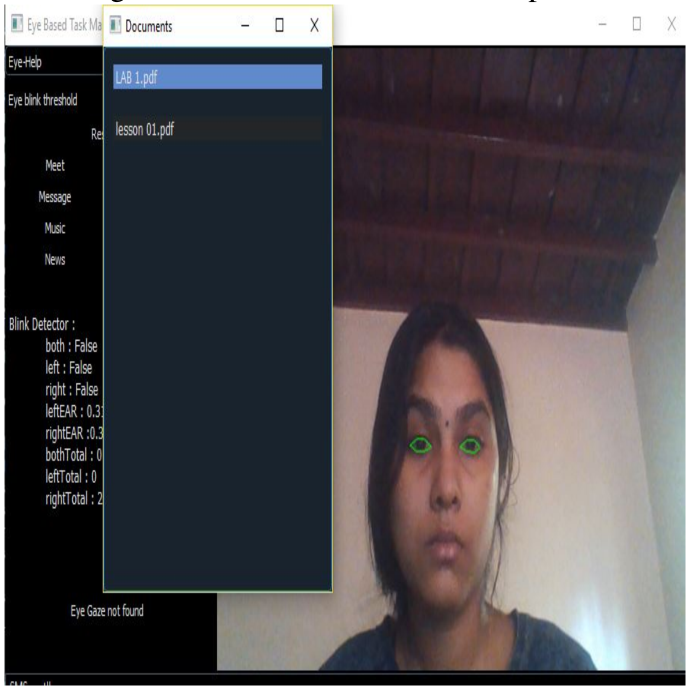
Your name (optional)
Lekshmi

We use cookies to offer you a better browsing experience, analyse site traffic, personalise content, and serve targeted advertisements. Continue browsing or choose accept if you agree to the use of cookies. Take a look at our Cookie Policy to learn more about how we use cookies. Cookie Policy

Figure 4.8: Video Conference Window

4.1.5 Output 5

When eye clicking document button a document option will be showed.



```
Blink Detector :
        both : False
        left : False
        right : False
        leftEAR : 0.3
        rightEAR :0.3
        bothTotal : 0
        leftTotal : 0
        rightTotal : 2
```

Figure 4.9: document Option

Chapter 5

CONCLUSION

In our system,eye gaze is used to interact with the system Eye gaze movement identification will be used to interact with the system.Gaze-based interaction describes the input of user commands by using an eye tracking system. In this work, we focus on camera-based eye tracking systems as they are less intrusive than other eye tracking approaches. Eye-based input has several advantages over voice-based and sensor-based solutions.Some of the advantages are present in the eye gaze based computer interaction.The eye gaze/blink classifier model is working pretty well but still we want to train more sample data to this and make the model more accurate. We will also experiment our system with all other feasible model if they work better. Finally, we would like to add more real life feature to our system and make it more usable and robust for our targeted users.

The key Features are:

– Some advantages of eye gaze based interaction will be Availability,Accessibility,position,performance etc.

– Face Detection,Eye Detection ,Eye Blink Detection,Eye Gaze movement Detection its direction identification are the initial steps done here.

– The system provide more accuracy and efficiency for eye detection detection.

– Using PyQt5 framework to design our GUI for user. For styling our GUI we have used a library called QdarkStyle.

5.1 Future Enhancement

In the future, the proposed method can be Changed using different algorithms.Use different method to find location and movement of eyes.the This study will likely lead to the development of better algorithms for detecting eyes.We can also include more features to in this work.Include more interactive features for the disabled people.

REFERENCES

[1] C. Li, C. Kim and J. Park, ”The Indirect Keyboard Control System by Using the Gaze Tracing Based on Haar Classifier in OpenCV,” 2009 International Forum on Information Technology and Applications, 2009, pp. 362-366, doi: 10.1109/IFITA.2009.276.

[2] A. Ardakani, C. Condo, M. Ahmadi and W. J. Gross, ”An Architecture to Accelerate Convolution in Deep Neural Networks,” in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 65, no. 4, pp. 1349-1362, April 2018, doi: 10.1109/TCSI.2017.2757036.

[3] A. A. Akinyelu and P. Blignaut, ”Convolutional Neural Network-Based Methods for Eye Gaze Estimation: A Survey,” in IEEE Access, vol. 8, pp. 142581-142605, 2020, doi: 10.1109/ACCESS.2020.3013540.

[4] Patil, Tanvi et al. “iGaze-Eye Gaze Direction Evaluation to Operate a Virtual Keyboard for Paralyzed People.” (2019).

[5] H. Cecotti, ”A Multimodal Gaze-Controlled Virtual Keyboard,” in IEEE Transactions on Human-Machine Systems, vol. 46, no. 4, pp. 601-606, Aug. 2016, doi: 10.1109/THMS.2016.2537749.

[6] F. M. Sigit, E. M. Yuniarno, R. F. Rachmadi and A. Zaini, ”Blinking Eyes Detection using Convolutional Neural Network on Video Data,” 2020 3rd International Conference on Information and Communications Technology (ICOIACT), 2020, pp. 291-296, doi: 10.1109/ICOIACT50329.2020.9331967.

[7] K. Goyal, K. Agarwal and R. Kumar, ”Face detection and tracking: Using OpenCV,” 2017 International conference of Electronics, Communication and Aerospace Tech-

nology (ICECA), 2017, pp. 474-478, doi: 10.1109/ICECA.2017.8203730.

[8] Yamoto, M., Monden, A., Matsumoto, K., Inoue, K., and Torii, K, ”Quick Button election with Eye Gazing for General GUI Environments,” International Conference on Software: Theory and Practice, August 2000.

[9] W. Yu, J. Xiu, C. Liu and Z. Yang, ”A depth cascade face detection algorithm based on adaboost,” 2016 IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC), 2016, pp. 103-107, doi: 10.1109/ICNIDC.2016.7974544.

[10] H. Qassim, A. Verma and D. Feinzimer, ”Compressed residual-VGG16 CNN model for big data places image recognition,” 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), 2018, pp. 169-175, doi: 10.1109/CCWC.2018.8301729.

[11] C. Meng and X. Zhao et al.”Webcam-Based Eye Movement Analysis Using CNN,” in IEEE Access, vol. 5, pp. 19581-19587, 2017, doi: 10.1109/ACCESS.2017.2754299.

[12] A. Poole and L. J. Ball, “Eye tracking in human computer interaction and usability research: current status and future prospects,” Encyclopedia of human computer interaction, pp. 211–219, 2006.

[13] EyeTrackingUpdate. (2011, Jun.) Eye tracking astigmatism correction. [Online]. Available: http://eyetrackingupdate.com/2011/06/27/eye- tracking-astigmatism-correction/.

[14] P. Turaga and R. Chellappa, “Age estimation and face verification across aging using landmarks,” IEEE Trans Inf. Forensic Secure, no. 7(6), pp. 1780– 1788, 2012. [Online]. Available: https://doi.org/10.1109/TIFS.2012.2213812.CrossRefGoogle Scholar. 4

[15] T. Devries, K. Biswaranjan, and G. W. Taylor, “Age estimation and face verification across aging using landmarks,” Canadian Conference on Computer and Robot Vision, IEEEMontreal, 2014, p. 98–103. [Online]. Available: http://ieeexplore.ieee.org/document/6816830/. 5

[16] Bo WU, Haizhou AI, Chang HUANG, Shihong LAO ”Fast Rotation Invariant Multi-View Face Detection Based on Real AdaBoost” Computer Society, 2004.