

# EYE CONTROLLED VIRTUAL MOUSE

NAVYASHREE D M , PUSHPA H R , REETHU S , VARSHITHA N

**Mrs. ARPITHA K**

Assistant Professor

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING BGS  
INSTITUTE OF TECHNOLOGY**

**B G NAGARA, NAGAMANGALA TALUK, MANDYA DISTRICT-571448**



## ABSTRACT

In this paper, an individual human computer interface system using eye motion is introduced. Traditionally, human computer interface uses mouse, keyboard as an input device. This paper presents hands free interface between computer and human. This technology is intended to replace the conventional computer screen pointing devices for the use of disabled. The paper presents a novel idea to control computer mouse cursor movement with human eyes. It controls mouse-moving by automatically affecting the position where eyesight focuses on, and simulates mouse-click by affecting blinking action. However, the proposed vision-based virtual interface controls system work on various eye movements such as eye blinking.

We describe a system that presents a hands-free interface between human and computer. Our system replaces conventional mouse in a new way that makes use of human facial features. It uses various image processing methods such as face detection, eye extraction. It uses a typical webcam to capture an input image. Controlling of mouse cursor is obtained by face movement as moving face up, down, left and right and mouse events are controlled through eye blinks, and keyboard events are controlled through hand gestures. To perform these operations different algorithms like Haar Cascade algorithm, Dlib. Our system is mainly aimed for disabled people to have effective communication with computer.

## CHAPTER 1

### INTRODUCTION

Recently there has been a growing interest in developing natural interaction between human and computer. Several studies for human-computer interaction in universal computing are introduced.

[1] The vision-based interface technique extracts motion information without any high cost equipments from an input video image. Thus, vision-based approach is taken into account an effective technique to develop human computer interface systems. For vision-based human computer interaction, eye tracking is a hot issue. Eye tracking research is distinguished by the emergency of interactive applications. However, to develop a vision-based multimodal human computer interface system, an eye tracking and their recognition is done. Real- time eye input has been used most frequently for disabled users, who can use only their eyes for input.

#### 1.1. OVERVIEW

Personal computers were initially used for solving mathematical problems and word processing. In recent years, however, computers have become necessary for every aspect of our daily activities. These activities range from professional applications to personal uses such as internet browsing, shopping, socializing and entertainment. Computers are designed to be readily accessible for normal individuals. However, for individuals with severe physical disabilities such as cerebral palsy or amyotrophic lateral sclerosis, usage of computers is a very challenging task. There have been many research studies on human computer interface (HCI) to improve the interaction between the user and the computer system. Most of these are applicable only to normal individuals. These interfacing methods include a touch sensitive screens, speech recognition methods and many others. Despite the success of these techniques, they were not suitable for the physically disabled individuals. Many researchers have tried to develop methods to help the disabled to interact with computers by using signals such as electroencephalography (EEG) from the brain, facial muscles signals (EMG) and electro-oculogram (EOG). Other methods include limbus, pupil and eye/eyelid tracking [4-5], contact lens method, corneal, pupil reflection relationship [6] and head movement measurement [7]. These methods require the use of attachments and electrodes to the head, which makes them impractical.

## **1.2. PROBLEM STATEMENT**

Eye tracking technology has become one of the most popular techniques within the human and computer interaction (HCI) this is very important for the people who have difficulty with speech and movement disabilities, especially for the paralyzed and amputees person. The idea of controlling the computers with the eyes will serve a great use for handicapped and disabled person. Also this type of control will eliminate the help required by other person to handle the computer. This measure will be the most useful for the person who is without hands through which they can operate with the help of their eye and facial movements.

## **1.3. THE SOLUTION**

In this paper, an individual human computer interface system using eye motion is introduced. Traditionally, human computer interface uses mouse, keyboard as an input device. This paper presents hands free interface between computer and human. This technology is intended to replace the conventional computer screen pointing devices for the use of disabled. The paper presents a novel idea to control computer mouse cursor movement with human eyes It controls mouse-moving by automatically affecting the position where eyesight focuses on, and simulates mouse-click by affecting blinking action. However, the proposed vision-based virtual interface controls system work on various eye movements such as eye blinking.

## **1.4. OBJECTIVES**

- Easy interaction with computer without using mouse.
- Limitation of stationary head is eliminated.
- Pointer of the mouse will move on screen where the user will be looking & the clicks will be performed by blinking.
- To develop a system which will only use Webcam, and to use human eyes as a pointing device for computer device.
- To provide user friendly human-computer interaction.Designing a System for tracking Face and Eye using Camera.

- Using Facial landmark's technique to detecting the movement of the face, eyes and calculating it cursor position and mouse clicks.

## 1.5. PROJECT SCOPE

- By implementing this process it is cleared that the cursor can be controlled by the eyeball movement i.e., without using hands on the computers.
- This will be helpful for the people having disability in using the physical parts of the computers to control the cursor points.
- Hands – free computing.
- Facilitating the handicapped in using the computer.
- Controlling the mouse pointer through eye movement. Eye based human computer interaction provides real time eye tracking and eye-gaze estimation.

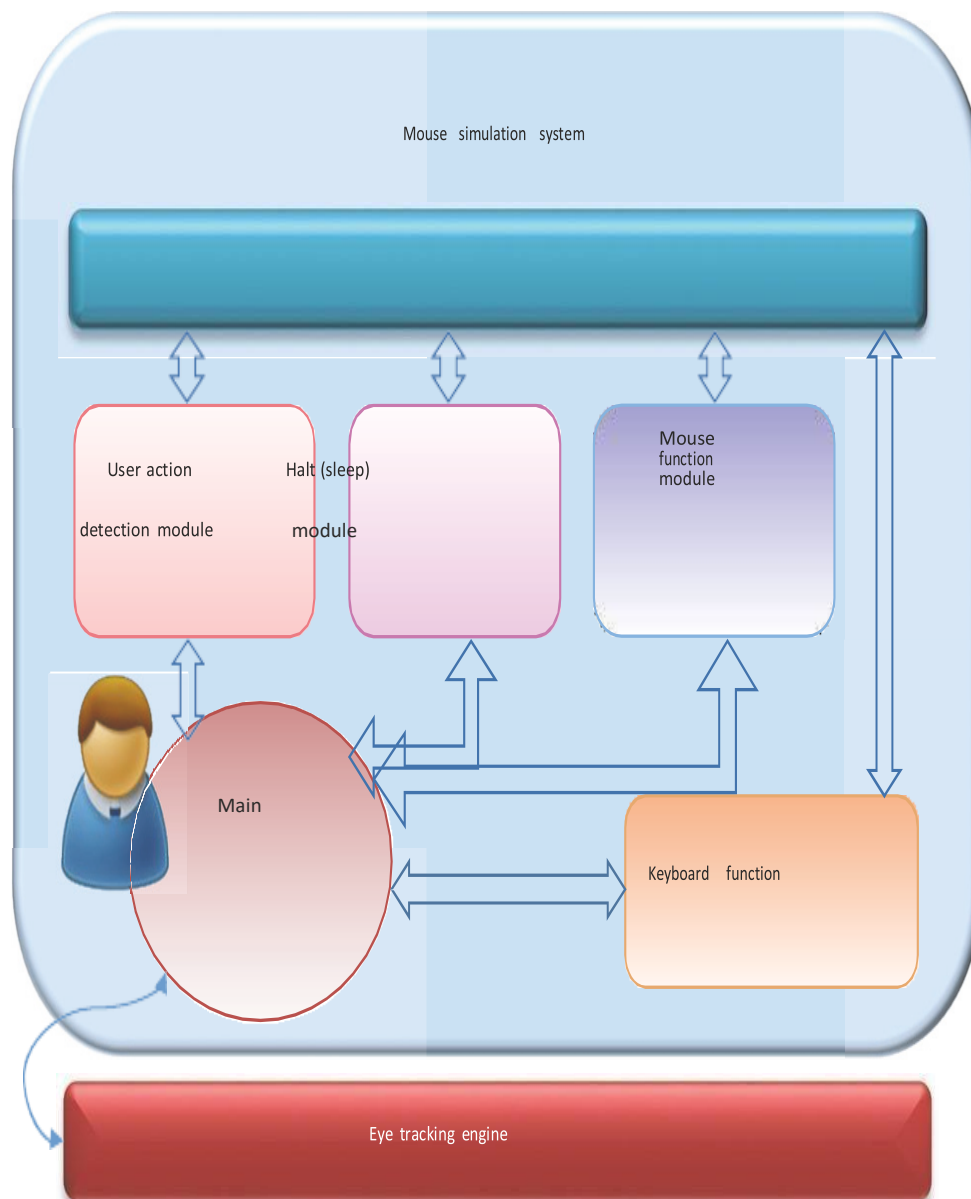
## 1.6. EXISTING SYSTEM

Many researchers have tried to develop methods to help the disabled to interact with computers by using signals such as electroencephalography (EEG) from the brain, facial muscles signals (EMG) and electro oculogram (EOG). The camera mouse was proposed by MargritBetkefor people who are quadriplegic and nonverbal. The movements of the user are tracked using a camera and these can be mapped to the movements of the mouse pointer which is visible on the screen. Yet another method was proposed by Robert Gabriel Lupu, for human computer interaction that made use of head mounted device to track eye movement and to translate it on screen. Another technique by Prof. Prashant salunke presents a techniques of eye tracking using Hough transform.

- Limitations:
  - i. Other methods include limbus, pupil and eye/eyelid tracking, contact lens method, corneal, pupil reflection relationship and head movement measurement.
  - ii. These methods require the use of attachments and electrodes to the head, which makes them impractical. Other high end techniques that are based on infrared tracking of the eye movements to control computers were exceptionally expensive and were not affordable for those who need them.

## 1.7. PROPOSED SYSTEM

In the proposed system, we have included the face detection, face tracking, eye detection and interpretation of a sequence of eye blinks in real time for controlling a nonintrusive human computer interface. Conventional method of interaction with the computer with the mouse is replaced with the human eye movements.



**FIG 1.1 THE ARCHITECTURE OF THE PROPOSED SYSTEM**

The work flow of the proposed system includes the following steps:

- The architecture of the proposed system design and the detail workflow of the system functionalities fig 1.1 will be described developed a prototype of eye control system that is based on low- cost eye trackers and the released Application Program Interface (API).
  - This system was composed of all of the functions of mouse and keyboard, and user should choose the desired function first and then do the real interaction with computer.
  - The accuracy of the eye tracking control system will depend on the eye tracker employed. However, the usage flow of this system goes against user's intuition and it is not natural for user to communicate with computer smoothly.
  - Therefore, in this paper we present an eye tracking based control system that not only supports the commercial low-cost eye trackers but also provides more natural and more convenient communication mechanisms for user-computer dialogue.
- Advantages:
    - i. This technique will help the paralyzed person, physically challenged people especially person without hands to compute efficiently and with the ease of use.
    - ii. Firstly, camera captures the image and focuses on the eye in the image using OpenCV code for face and eye detection.
    - iii. This results the center position of the human face. Then the center position of the face is taken as a reference and based on that the human or the user will control the cursor by moving left and right.

## **1.8. ORGANIZATION OF PROJECT REPORT**

The report is organized in the following manner

- The chapter 1 gives the “Overview of the project”, “Problem Statement”- What the problem is that we are trying to find the answer to, why it's a problem. “The Solution of the project”- how our project will help answer it. “Objectives”, “Project Scope”- Where this project is useful. “Existing System & Proposed System”.
- The chapter 2 gives the “LITERATURE SURVEY”- For your literature survey review information from various sources (books, journals, conference proceedings, web sites, product

literature, etc.). The goal is to show that you have read the information about the system from various sources and understand the problems/issues of the system.

- The chapter 3 gives the “SOFTWARE REQUIREMENT SPECIFICATIONS”, “Functional requirement”, “Non-functional requirements”, “System Requirements”.
- The chapter 4 gives the “SYSTEM ANALYSIS”- Analysis done for planning the design will be explained. “High Level Design”, “System Architecture”, “ DB Design”, “ Low Level Design”, “User Interface Design”.
- The chapter 5 gives the “ IMPLEMENTATION”, “Control Flow”, “Pseudo code or Algorithm”- The detail pseudo code upto sub function level should be detailed in this section.
- The chapter 6 gives the “TESTING”, “Design of Test Cases”- Different Testing approaches used for testing the “application” of the project will be explained. Testing approaches will be explained in the “context of the project”. “Test Cases”.
- The chapter 7 gives the “RESULTS AND SNAPSHOTS”- Important screenshots of the project will be shown, along with the explanation of each screenshot with name.

## CHAPTER 2

### LITERATURE SURVEY

#### [1] "Design and Development of Hand Gesture Based Virtual Mouse,"

**Author name: K. H Shibly, S. Kumar Dey, M .A Islam and S.IftekharShowrav.**

**Published in: 2019**

For human eye (Iris) detection, batch mode is employed. Iris tracking technique is implemented on static images. This technique simply works when the direction of iris is left, right or center. If the position of iris is up or down, it does not work. The system not works in real time. It isnot expert to handle blinks and close eyes.

This paper is aimed for designing and implementing a human computer interface system that tracks the direction of the human eye. The particular motion as well as direction of the iris is employed to drive the interface by positioning the mouse cursor consequently. The location of the iris is completed in batch mode. This means that the frames are stored in a permanent storage device and are retrieved one by one. Each of the frames is processed for finding the location of the iris and thereby placing the mouse cursor consequently. Such a system that detects the iris position from still images provides an alternate input modality to facilitate computer users with severe disabilities.

#### **Merits:**

- With eye mouse we can implement WSN which can detect any event at a desired position in sensing field without human interaction directly.
- By this project we can implement low cost based surveillance system which in user friendly and easily recognized places in any real world system

#### **Demerits:**

- Real time limitations.
- Camera resolution

#### [2] "Eye-controlled mouse cursor for physically disabled individual"

**Author's name: Mohamed Nasor, K KMujeeb Rahman, Maryam, Mohamed Zubair**

**Published in: 2018**



This paper presents a novel algorithm for controlling the movement of a computer screen cursor using the iris movement. By accurately detecting the position of the iris in the eye and mapping that to a specific position on the computer screen, the algorithm enables physically disabled individuals to control the computer cursor movement to the left, right, up and down. The algorithm also enables the person to open and close folders or files or applications through a clicking mechanism.

**Merits:**

- Eye mouse is a very helpful tool which can replace a normal mouse device.
- A gift for physical challenged people.

**Demerits:**

- It is difficult to control eye position accurately all the times unlike mouse. Eye tracker provides instable output when it does not get appropriate image of the eye in consecutive frames.

**[3] “An image-based eye controlled assistive system for paralytic patients”**

**Author’s name: Neil Castellino and Michelle Alva**

**Published in: 2017.**

Communication is an essential part of human life which paralytic patients with locked-in syndrome are deprived of. In locked-in syndrome, the patient cannot move any of his voluntary muscles except the eyes. Taking this into consideration, the proposed system is designed to detect the face and pupil of the patient through a standard webcam using Haar cascade classifiers and Circular Hough Transform algorithm respectively. The proposed system displays different images of daily activities. The patient will have to look at an image for a period above a pre-decided threshold time in order to select it. Subsequently, the system will track the point of gaze of the patient and will select the image accordingly after a confirmation from the patient. Based on this confirmation, the aide will be notified via text or audio. Successful implementation of the system will help the paralytic patient to easily communicate his needs to the aide.

**Merits:**

- It helps to communicate with machines in order to automate manual tasks.
- It increases user experience and performance in playing games.

**Demerits:**

- It does not work with few users who wear contact lenses or have long eye lashes.
- Improper usage may lead to eye problems.

**[4] "Eye Tracking Based Control System for Natural Human-Computer Interaction"****Author's name: Shu-Fan Lin, Xuebai Zhang, Shyan-Ming Yuan****Published in: 2017**

Eye movement can be regarded as a pivotal real-time input medium for human-computer communication, which is especially important for people with physical disability. In order to improve the reliability, mobility, and usability of eye tracking technique in user-computer dialogue, a novel eye control system with integrating both mouse and keyboard functions is proposed in this paper. The proposed system focuses on providing a simple and convenient interactive mode by only using user's eye. The usage flow of the proposed system is designed to perfectly follow human natural habits. Additionally, a magnifier module is proposed to allow the accurate operation. In the experiment, two interactive tasks with different difficulty (searching article and browsing multimedia web) were done to compare the proposed eye control tool with an existing system. The Technology Acceptance Model (TAM) measures are used to evaluate the perceived effectiveness of our system.

**Merits:**

- It helps to learn from experts delivering skills.
- It makes technology more intuitive.

**Demerits:**

- It does not work with few users who wear contact lenses or have long eye lashes.
- Eye movements of some users are often un-intentional. This results into unwanted responses by the system.

**[5] "Hardware and software implementation of real time electrooculogram (EOG) acquisition system to control computer cursor with eyeball movement".****Author's name: Hossain, Zakir, MdMaruf Hossain Shuvo, and PrionjitSarker****Published in: 2017**

Human computer interface (HCI) is an emerging technology of neuroscience and artificial intelligence. Development of HCI system using bio signal e.g. Electrooculogram (EOG), Electromyogram (EMG), Electroencephalogram (EEG), Functional near-infrared spectroscopy (fNIRS) etc. are attracted more and more attention of researchers all over the world in recent years because through this it is possible to get acquainted with advanced technologies of artificial intelligence. This paper presents the design and implementation of a fully functional Electrooculogram (EOG) based human computer interface. In this work we have designed and implemented necessary hardware and software for EOG signal acquisition along with controlling hardware such as wheelchair, robotic arm, mobile robot etc., and move computer mouse cursor simultaneously using EOG signal.

**Merits:**

- This interface has three portion: EOG signal acquisition and amplification, analog to digital conversion, and real time hardware and mouse cursor movement. Eye movement is detected by measuring potential difference between cornea and retina using five Ag-AgCl disposable electrodes .
- EOG is much more reliable, cost efficient and non-invasive to implement.

**Demerits:**

- It reduces the number of potential users of the system and its applications since it increases the cost of the system and reduces its flexibility, portability, and autonomy.

**[6] “Hand Gesture Recognition System Using Image Processing”****Author’s name: Sagar P. More and Abdul Sattar,****Published in: 2016**

A gesture is a form of nonverbal communication or non-vocal communication in which visible bodily actions communicate particular messages, either in place of, or in conjunction with, speech. Gestures include movement of the hands, face, or other parts of the body.

Gestures differ from physical non-verbal communication that does not communicate specific messages, such as purely expressive displays, proxemics, or displays of joint attention. Gestures allow individuals to communicate a variety of feelings and thoughts, from contempt and hostility to approval and affection, often together with body language in addition to words when they speak.

Gesture processing takes place in areas of the brain such as Broca's and Wernicke's areas, which are used by speech and sign language. In fact, language is thought to have evolved from manual gestures. The theory that language evolved from manual gestures, termed Gestural Theory.

**Merits:**

- Helpful in commercial interactive game and advertisement.
- Easier computer control.
- Ideal for use with desktop computers.

**Demerits:**

- Improper usage may lead to eye problems.
- Frames captured per second.

**[7] “Real-Time Eye Blink Detection using Facial Landmarks”**

**Author’s name: Čech and Soukupová**

**Published in: 2016.**

A real-time algorithm to detect eye blinks in a video sequence from a standard camera is proposed. Recent landmark detectors, trained on in-the-wild datasets exhibit excellent robustness against face resolution, varying illumination and facial expressions. We show that the landmarks are detected precisely enough to reliably estimate the level of the eye openness. The proposed algorithm therefore estimates the facial landmark positions, extracts a single scalar quantity eye aspect ratio (EAR) – characterizing the eye openness in each frame. Finally, blinks are detected either by an SVM classifier detecting eye blinks as a pattern of EAR values in a short temporal window or by hidden Markov model that estimates the eye states followed by a simple state machine recognizing the blinks according to the eye closure lengths. The proposed algorithm has comparable results with the state-of-the-art methods on three standard datasets.

**Merits:**

- It increases computing and resource efficiency.
- It helps to assess human conditions and behaviours

**Demerits:**

- It requires some calibration time before it gives satisfactory results. Hence few users deviate themselves from using it.
- It is expensive technology due to costly hardware requirements.

**[8] “Using for a videobased mouse-Kernels replacement interface”.****Author’s name: Samuel Epstein-Eric MissimerMargritBetke****Published in: 2012**

Some people cannot use their hands to control a computer mouse due to conditions such as cerebral palsy or multiple sclerosis. For these individuals, there are various mouse-replacement solutions. One approach is to enable them to control the mouse pointer using head motions captured with a web camera. One such system, the Camera Mouse, uses an optical flow approach to track a manually selected small patch of the subject’s face, such as the nostril or the edge of the eyebrow. The optical flow tracker may lose the facial feature when the tracked image patch drifts away from the initially-selected feature or when a user makes a rapid head movement.

**Merits:**

- Ideal for use with desktop computers.
- Usually supplied as part of a new computer system

**Demerits:**

- They need a flat space close to the computer.
- Older style mice which have roller balls can become clogged with grease and grime and lose their accuracy until cleaned.

**[9] “Real-time eye-gaze estimation using a low-resolution webcam”****Author’s name: Yu-Tzu Lin Ruei-Yan Lin Yu-Chih Lin Greg C Lee****Published in: 2012**

Eye detection and gaze estimation play an important role in many applications, e.g., the eye-controlled mouse in the assisting system for disabled or elderly persons, eye fixation and saccade in psychological analysis, or iris recognition in the security system. Traditional research usually achieves

eye tracking by employing intrusive infrared-based techniques or expensive eye trackers. Nowadays, there are more and more needs to analyze user behaviors from tracking eye attention in general applications, in which users usually use a consumer-grade computer or even laptop with an inexpensive webcam.

**Merits:**

- Eye detection and gaze estimation play an important role in many applications, e.g., the eye-controlled mouse in the assisting system for disabled or elderly persons, eye fixation and saccade in psychological analysis, or iris recognition in the security system.
- Nowadays, there are more and more needs to analyze user behaviors from tracking eye attention in general applications, in which users usually use a consumer-grade computer or even laptop with an inexpensive webcam

**Demerits:**

- It requires some calibration time before it gives satisfactory results. Hence few users deviate themselves from using it.
- It is expensive technology due to costly hardware requirements

**[10] “Facial Feature Based Method For Real Time Face Detection and Tracking I-CURSOR”**

**Author’s name: International Journal of EnggResearchand App., Vol. 2, pp**

**Published in: 2012**

In the proposed system, we have included the face detection, face tracking, eye detection and interpretation of a sequence of eye blinks in real time for controlling a nonintrusive human computer interface. Conventional method of interaction with the computer with the mouse is replaced with the human eye movements. This technique will help the paralyzed person, physically challenged people especially person without hands to compute efficiently and with the ease of use. Firstly, camera captures the image and focuses on the eye in the image using OpenCV code for pupil detection. This results the center position of the human eye (pupil). Then the center position of the pupil is taken as a reference and based on that the human or the user will control the cursor by moving left and right.

**Merits:**

- Facial features (nose tip and eyes) are detected and tracked in real-time to use their actions as mouse events. In our work we were trying to compensate people who have hands disabilities that prevent them from using the mouse by designing an application that uses facial features (nose tip and eyes) to interact with the computer.
- It can be applied to a wide range of face scales. Our basic strategy for detection is fast extraction of face candidates with a Six-Segmented Rectangular (SSR) filter and face verification by a support vector machine.

**Demerits:**

- Real time limitations.
- Camera resolution

## **CHAPTER 3**

# **SOFTWARE REQUIREMENT SPECIFICATIONS**

### **3.1. Functional Requirements**

#### **➤ Face Detection Methods**

Face Detection Methods Face detection is the step stone for all facial analytical algorithms, including facial orientation, facial mapping, face recognition, facial recognition / authentication, facial expression tracking / recognition, gender recognition.

#### **➤ FEATURE BASED METHODS**

The characteristic-based approach is to identify faces by extracting facial structural features. It is first classifier and then used to distinguish between facial and non-facial regions. The aim isto transcend the limitations of our instructive awareness of faces. This method is divided into many phases and also images of several faces.

#### **➤ EYE DETECTION APPROACHES**

There are some detection methods that undergo various stages, such as the positioning of the face and eyes from different locations.

Regression Approach- This minimizes the difference between expected and real eye positions.

#### **➤ EYE TRACKING TECHNIQUE**

Pupil Tracking- Pupil tracking is a gaze-detection technique that is often used in combination with other methods. The eye is much more than a tool for moving the cursor at high speeds. Input of eye movement is obviously faster than input of any factors. Until any mechanical pointing system is controlled, the user typically looks at the destination to which he or she wants to travel.

#### **➤ CONTROLLING THE MOUSE POINTER**

The input of eye movement is taken from the individual's pupil. If a person looks at a center mouse pointer, that point would be taken as the input point and it sets that location as the basis for gaze tracking and it begins moving in the direction of the person's eye movement and the cursor stops moving when the person's eye hits its initial place.



- Left and right movement of the pupil Horizontal eye pupil movement can be achieved using circular artifacts. If the pupil moves in the left direction, the mouse pointer moves in the left direction as well, and the same happens in the right direction as well.

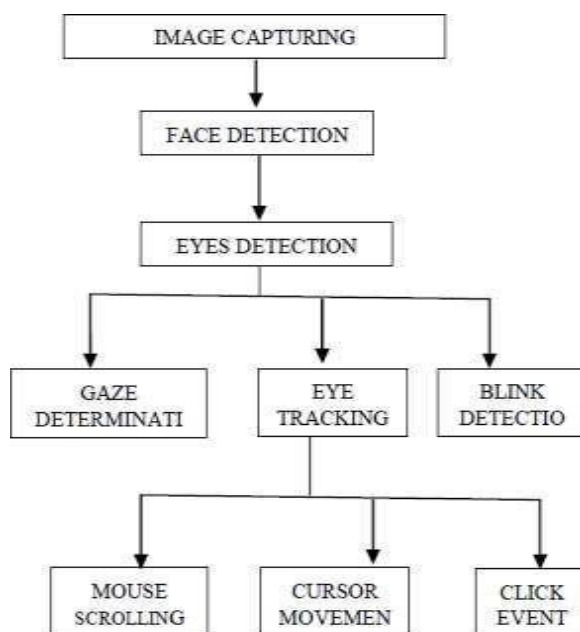


- Up and down movement of the pupil Vertical eye pupil movement can be achieved by using pupil scale. The eyes are in slightly half-closed state when gazing downwards. This phenomenon can be used to guide the step from top to bottom of the mouse pointer.



## ➤ CONTROLLING THE MOUSE CLICK EVENTS

A blink of an eye can be done to the mouse click case. As the mouse pointer begins to shift the eye pupil's path if a person decides to perform a chosen event then the person blinks the eye mouse pointer executes the click event.



**FIG 3.1 WORKING OF CONTROLLING THE MOUSE CLICK EVENTS**

The user will sit in front of the personal computer or laptop screen, a small video camera positioned above the panel to monitor the eyes of the user. The computer analyzes the eye's video image on an ongoing basis and decides where the user views the device. There is nothing connected to the head or body of the customer. The user stares at the key for a given time span in order to "pick" any key, and to "press" any key, the user only blinks his eye. Calibration procedure isn't needed in this framework. Data for this method is eye-only. There is neither external hardware attached nor needed.

## **3.2. Non-Functional Requirements**

### **• FACE DETECTION**

Face detection has always been a vast research field in the computer vision world. Considering that it is the back bone of any application that deals with the human face. The face detection method can be organized in two categories:

#### **➤ FEATURE-BASED METHOD:**

The first involves finding facial features (e.g. noses, eye brows, lips, eye pupils) and in order to verify their authenticity performs by geometrical analysis of their locations, areas and distances from each other. This analysis will eventually lead to localization of the face and the features that it contains. The feature based analysis is known for its pixel-accuracy, features localization and speed, on the other hand its lack of robustness.

#### **➤ IMAGE-BASED METHOD:**

The second method is based on scanning the image of interest with a window that looks for faces at all scales and locations. This category of face detection implies pattern recognition, and achieves it with simple methods such as template matching or with more advanced techniques such as neural networks and support vector machines. Before over viewing the face detection algorithm we applied in this work here is an explanation of some of the idioms that are related to it.

### **• EYE DETECTION APPROACHES**

Following are the various eye detection approaches:

#### **➤ REGRESSION APPROACH**

Tries to minimize the distance between the predicted and actual eye positions. Simply by understanding the functional mapping from the input image to eye positions.

### ➤ **BAYESIAN APPROACH**

Learns model of eye appearance and non-eye appearance. Use Baye's principle to build a "probability of eye". Produces output for patches around each pixel of the input image, from which a prediction will be extracted.

### ➤ **DISCRIMINATIVE APPROACH**

Treats the problem as one of classification. A classifier is trained to produce positive output for patches around the eye and negative elsewhere.

From the above approaches, Bayesian approach is been taken into consideration.

## **3.3. SYSTEM REQUIREMENTS**

- **Hardware Requirements**

Processor : Intel i3 2.0 GHz or more.  
RAM : 4 GB or More  
Hard disk : 512GB or more

Monitor : 15" CRT, or LCD monitor  
Keyboard : Normal or Multimedia  
Mouse : Logitech Mouse

- **Software Requirements:**

Operating System : Windows 7 or above  
Technology : Python, Machine Learning  
Libraries : Numpy — 1.13.3, OpenCV — 3.2.0, PyAutoGUI — 0.9.36  
Dlib — 19.4.0, Imutils — 0.4.6

## CHAPTER 4

# SYSTEM ANALYSIS AND DESIGN

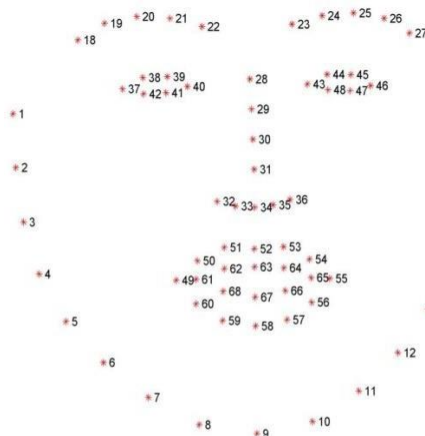
### 4.1. System Analysis

The user has to sit in front of the screen of personal computer or laptop, a specialized video camera mounted above the screen to observe the user's eyes. The computer continually analyzes the video image of the eye and determines where the user is looking on the screen. Nothing is attached to the user's head or body. To "select" any key, the user looks at the key for a specified period of time and to "press" any key, the user just blink the eye. In this system, calibration procedure is not required. For this system input is only eye. No external hardware is attached or required.

### 4.2. High Level Design

This project is deeply centered around predicting the facial landmarks of a given face. We can accomplish a lot of things using these landmarks. From detecting eye-blinks [3] in a video to predicting emotions of the subject. The applications, outcomes, and possibilities of facial landmarks are immense and intriguing.

Dlib's prebuilt model, which is essentially an implementation of [4], not only does a fast face-detection but also allows us to accurately predict 68 2D facial landmarks. Very handy.

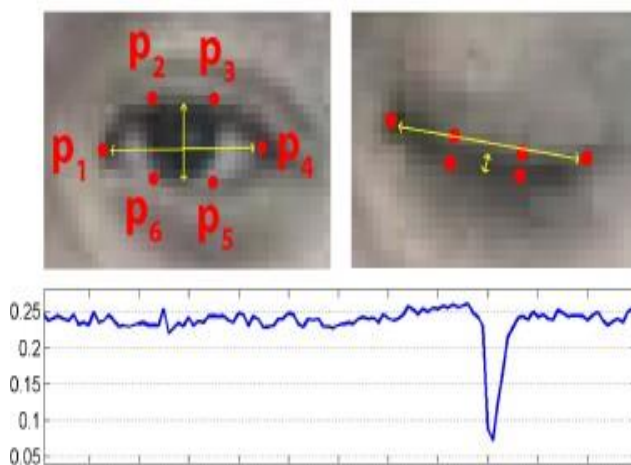


Source: C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, M. Pantic. 300 Faces in-the-Wild Challenge: The first facial landmark localization Challenge. Proceedings of IEEE Int. Conf. on Computer Vision (ICCV-W, 300 Faces in-the-Wild Challenge (300-W)). Sydney, Australia, December 2013.

Using these predicted landmarks of the face, we can build appropriate features that will further allow us to detect certain actions, like using the eye-aspect-ratio (more on this below) to detect a blink or a wink, using the mouth-aspect-ratio to detect a yawn etc or maybe even a pout. In this project, these actions are programmed as triggers to control the mouse cursor. PyAutoGUI library was used to move the cursor around.

### • EYE-ASPECT-RATIO (EAR)

You will see that Eye-Aspect-Ratio [1] is the simplest and the most elegant feature that takes good advantage of the facial landmarks. EAR helps us in detecting blinks [3] and winks etc.



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

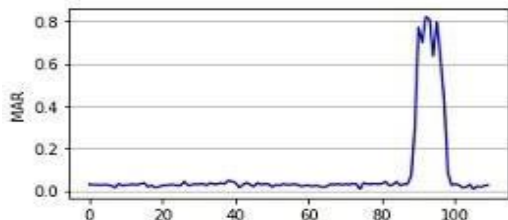
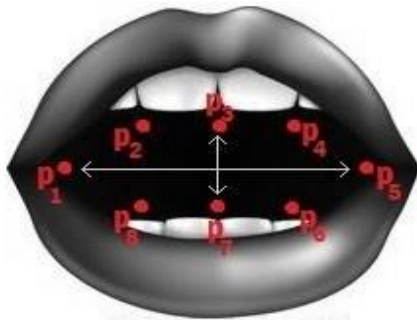
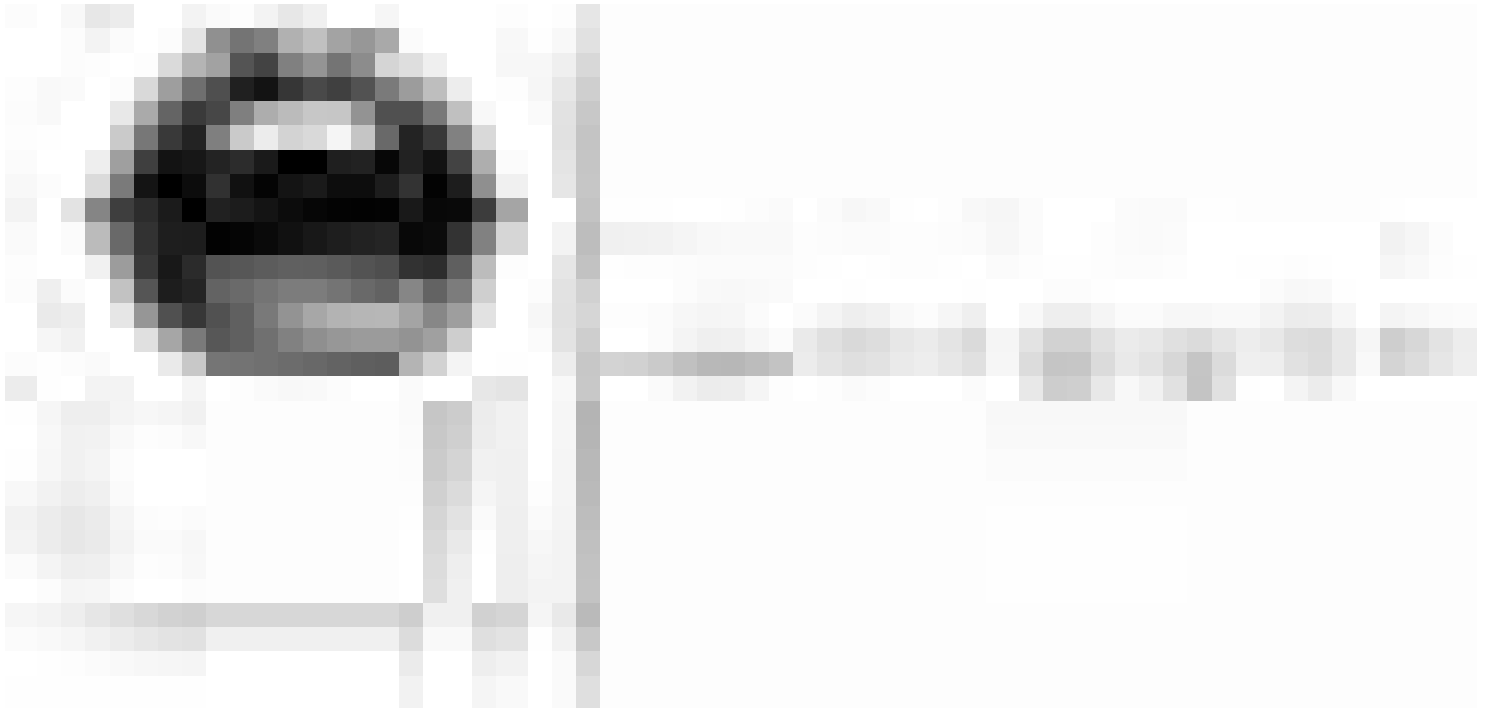
Source: Tereza Soukupova<sup>1</sup> and Jan C<sup>2</sup>ech. *Real-Time Eye Blink Detection using Facial Landmarks* at <https://vision.fe.uni-lj.si/cvww2016/proceedings/papers/05.pdf>

You can see that the EAR value drops whenever the eye closes. We can train a simple classifier to detect the drop. However, a normal *if* condition works just fine. Something like this:

```
if EAR <= SOME_THRESHOLD:
    EYE_STATUS = 'CLOSE'
```

### • MOUTH-ASPECT-RATIO (MAR)

Highly inspired by the EAR feature, I tweaked the formula a little bit to get a metric that can detect opened/closed mouth. Unoriginal but it works.

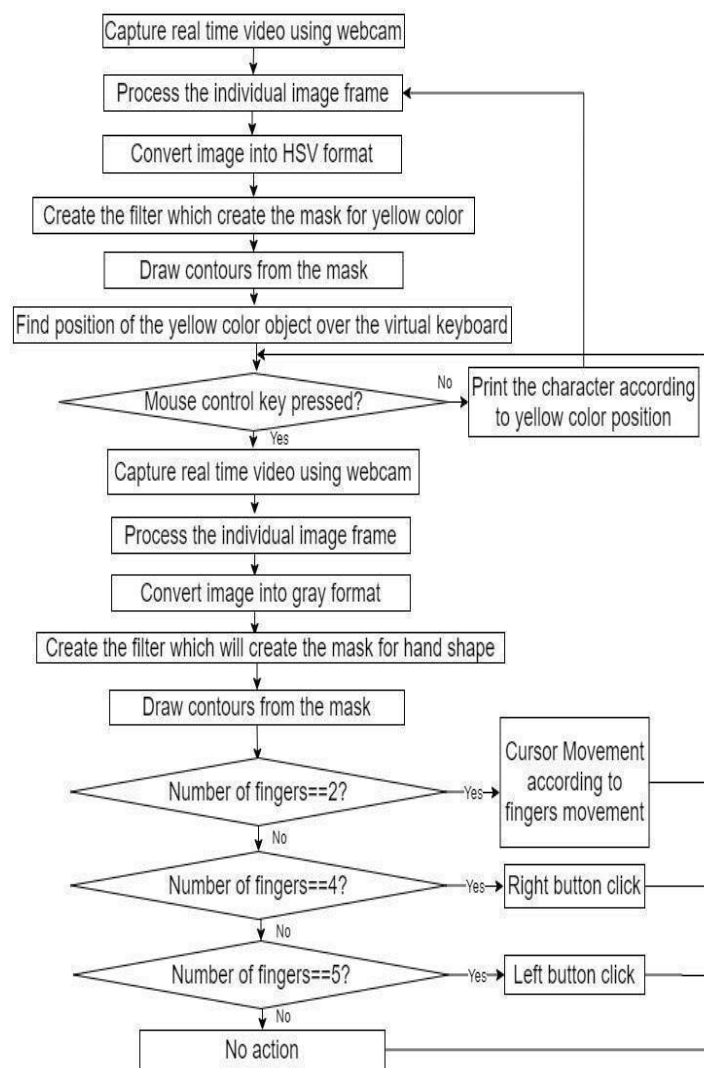


$$\text{MAR} = \frac{\|p_2 - p_8\| + \|p_3 - p_7\| + \|p_4 - p_6\|}{2 \|p_1 - p_5\|}$$

Similar to EAR, MAR value goes up when the mouth opens. Similar intuitions hold true for this metric as well.

### 4.2.1. System Architecture

The camera mouse was proposed by Margrit Betke for people who are quadriplegic and nonverbal. The movements of the user are tracked using a camera and these can be mapped to the movements of the mouse pointer which is visible on the screen. Yet another method was proposed by Robert Gabriel Lupu, for human computer interaction that made use of head mounted device to track eye movement and to translate it on screen. Fig 3.1 Procedure of gesture-based mouse and keyboard. Another technique by Prof. Prashant Salunke represents a techniques of eye tracking using Hough transform.



**FIG 4.1 PROCEDURE OF GESTURE-BASED MOUSE AND KEYBOARD**



### 4.2.3. Data Base Design

In the system architecture, we have included the face detection, face tracking, eye detection and interpretation of a sequence of eye blinks in real time for controlling a nonintrusive human computer interface. Conventional method of interaction with the computer with the mouse is replaced with the human eye movements. This technique will help the paralyzed person, physically challenged people especially person without hands to compute efficiently and with the ease of use. Firstly, camera captures the image and focuses on the eye in the image using OpenCV code for face and eye detection. This results the center position of the human face. Then the center position of the face is taken as a reference and based on that the human or the user will control the cursor by moving left and right.

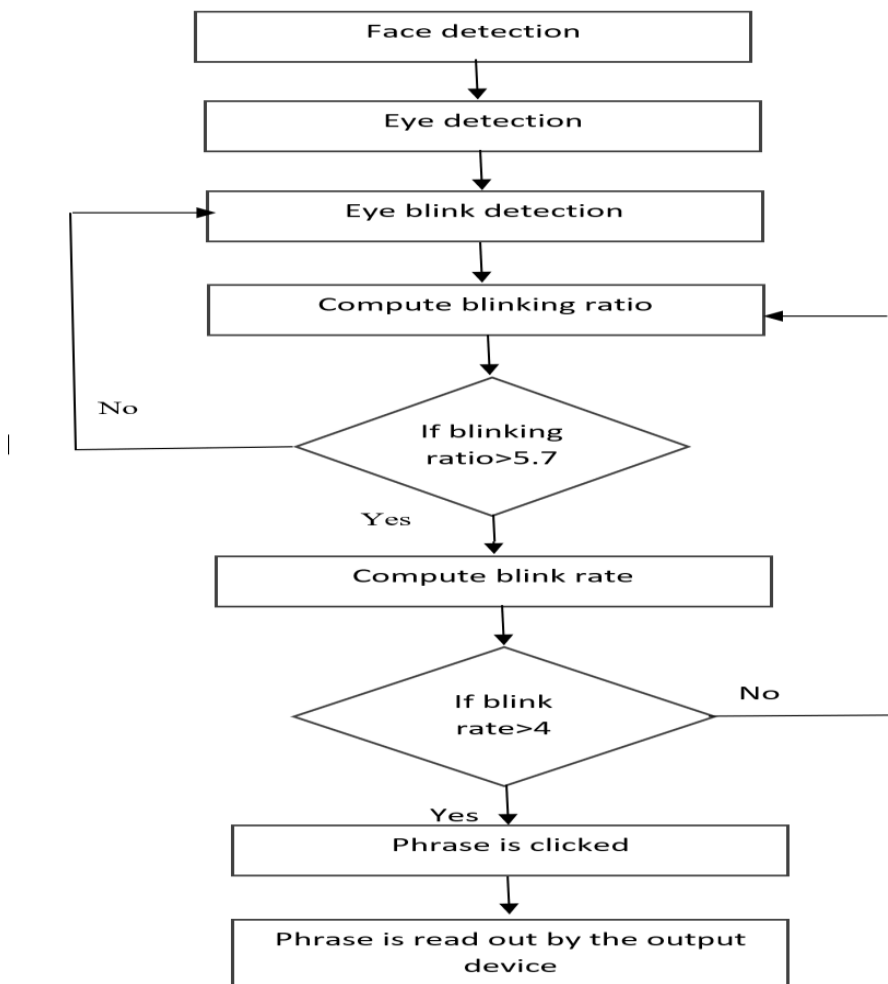


FIG 4.2 THE WORK FLOW OF THE SYSTEM ARCHITECTURE.



Firstly the working of this Fig 4.2 The work flow of the proposed system is recognizing the face followed by the eye and then the eye blink that enables the user to select the needed phrase in the keyboard that appears in the screen. So in general, the system is set to propose a protocol when the blink ratio is said to be greater than 5.7 then the blink rate is computed, else if the ratio is lesser than 5.7 it is said to be a non blink and it is not computed further. This is totally dependent on the ratio in which the blink is made and checking whether it's a blink or not. Secondly, on to the rate of the blink, if the blink rate is greater than 4 then the pointer clicks the desired phrase set and hence finally the audio assistance is also delivered, but whereas when the rate is lesser than 4, it has to be again computed and checked to get the output. The working is so simple that any physically disabled people can follow this, and the main feature is the blink detection that happens. To detect the face, facial landmarks feature is implemented with the help of dlib.

The complete process of proposed system mainly under goes in following process which is as follows.

- **Viola-Jones Face Detection**

The Cascade Classification is done as follows.

Stage 1: is there a face in the current rectangle according to classifier 1? If yes then go to stage 2, if no rectangle does not contain a face.

Stage 2: is there a face in the current rectangle according to classifier 2? If yes then go to stage 3, if no rectangle does not contain a face.

Stage n: is there a face in the current rectangle according to classifier n? If yes there is a face in current rectangle.

- **Template Matching**

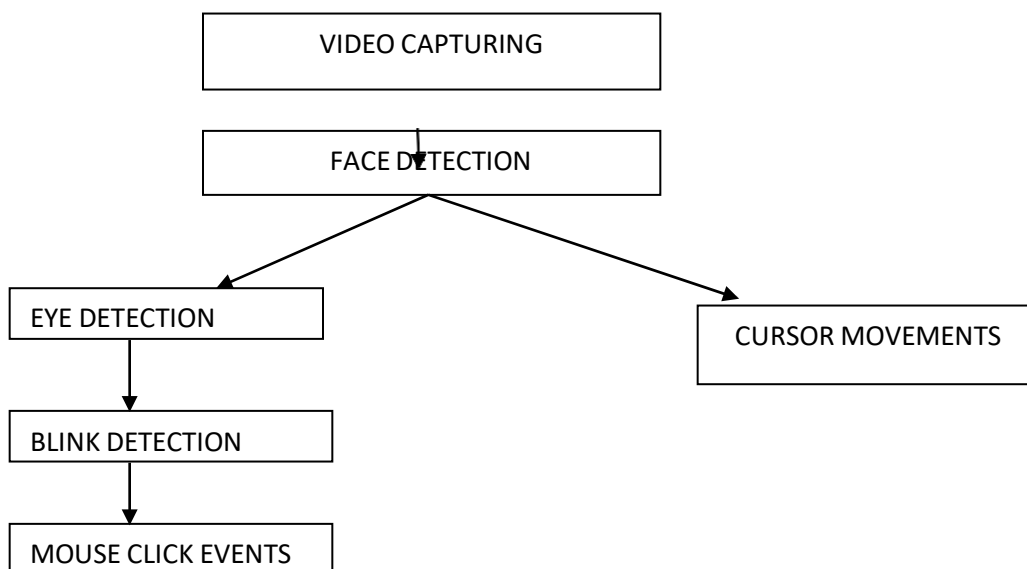
We use Template Matching for eye blink detection in proposed system. Template matching is a technique in image processing for finding small parts of an image which match a template image in real time system. It can be used in many real time systems for image recognition.

Template matching can be done two ways as follows

- i. Feature-based Method
- ii. Template-based Method

### 4.3. Low Level Design

Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms.



**FIG 4.3 SYSTEM FLOW OF THE PROJECT**

- FLOW CHART**

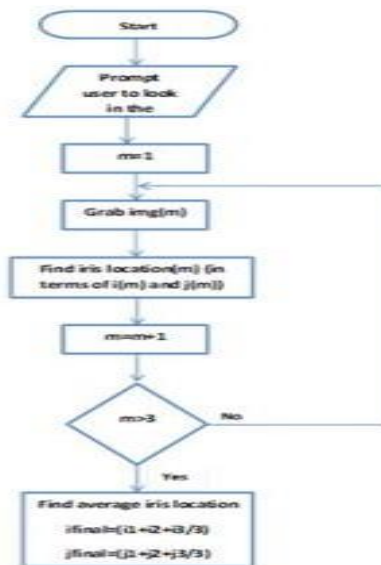


FIG 4.4 FLOW CHART OF IRIS DETECTION

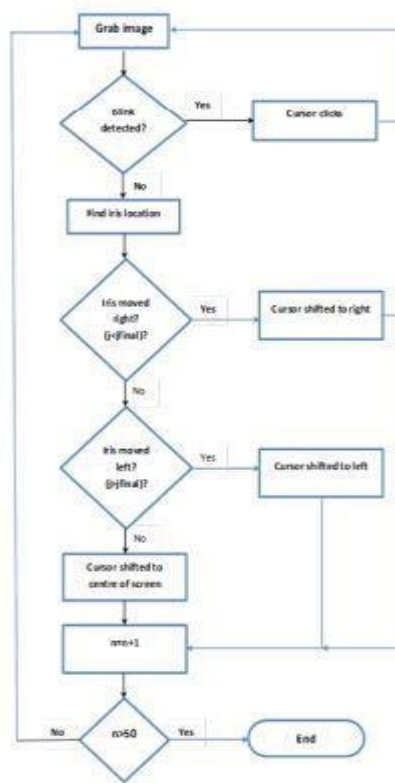
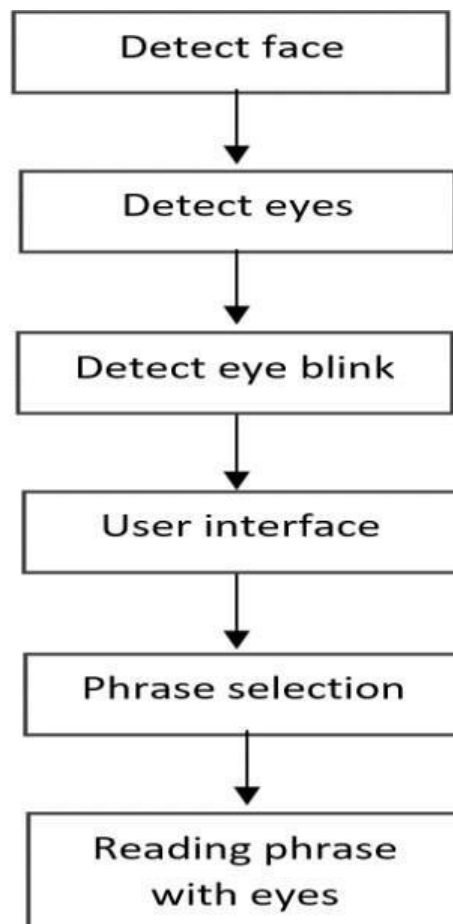


FIG 4.5 FLOW CHART OF BLINK DETECTION

## 4.4. User Interface Design

### • ARCHITECTURE

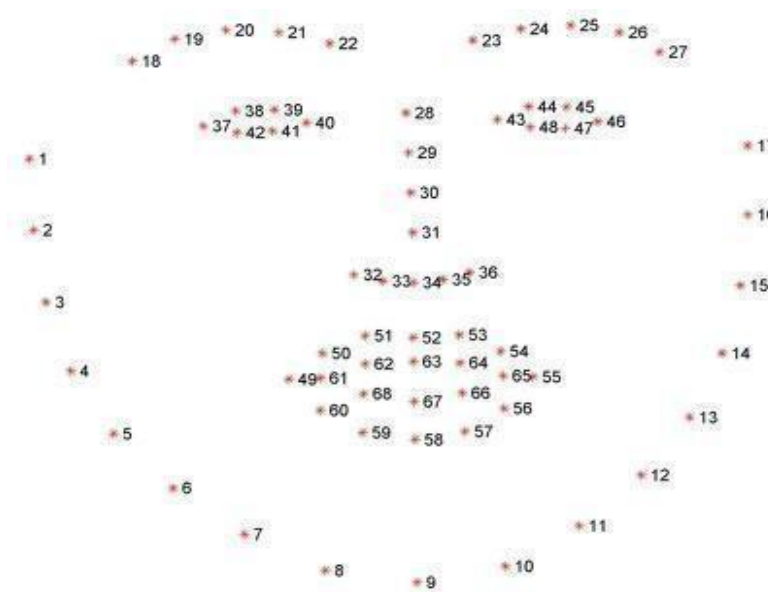
In this paper, we have designed a system fig 4.6 which can be easily controlled by the paralysed people. This system provides the speaking power without using mouth. Users can able to speak what they want through their eye blink. The user interface is very easy to use for all age groups from children to elderly person. The constructed system takes live video taken using webcam as input. From the input, the system will detect face and eye using facial landmark structure. The system will be built on several parts as detecting face, eyes, eye blinks, virtual interface on screen, select the phrase button and finally read the phrase using eye blinking with the help of a speaker. The system architecture is as follows:



**FIG 4.6 THE ARCHITECTURE FOR PARALYSED PEOPLE.**

- **FACE DETECTION:**

It is a technology of recognizing human faces from any image or video . Mostly, OpenCv and dlib are used to detect face by using various methods .The detector used here is made up of classic Histogram of Oriented Gradients (HOG) feature along with a linear classifier. Facial landmarks detector is implemented inside dlib to detect facial features like eyes, ear, nose, etc.



## • EYE DETECTION:

After detecting the face, eye region is detected with the help of facial landmark features. Using the face landmarks datasets, we can point out 68 landmarks on the face, each landmark is assigned with an index. Using these indices, the desired region of the face is detected, . Point index for two eyes:

- left eye :- ( 37, 38, 39, 40, 41, 42)
- right eye :- (43, 44, 45, 46, 47, 48)

After extracting eye region, it is processed for detecting eye blinks. The eye region detection is done at the initial stage of the system.

## • EYE BLINK DETECTION:

With the exact eye region, we can detect the blinks with the help of two lines. One drawn horizontally and other drawn vertically splitting the eyes. Temporary closure of eyes along with the movement of eyelids is known as blink. It is a rapid natural process.

We have to find out what happens when eye is blinked. We can conclude that the eye is closed/blinked when:

- Eyeball is not visible Eye lid is closed
- Upper and lower eyelids are connected together

If these actions are occurred for a period of (approximately 0.3 seconds to 0.4 seconds) time, we can assume it as a blink if it is longer than that then it can be taken as closed eyes. For a opened eye, both vertical and horizontal lines are almost identical while for a closed eye, vertical line becomes very smaller or almost vanished. Keeping the horizontal line as point of reference, a ratio is computed with respect to vertical line. We have to set a threshold value here and if the ratio is greater than the value then we can assume the eye is closed otherwise open.



Opened Eye



Closed Eye

## • USER INTERFACE:

In our system, user interface consists of list of phrases and the mouse pointer scrolling across the phrases top to bottom and vice versa.



### i) Buttons:

Each phrase is written on a button. buttons are created using python's Tkinter module. Tkinter is a interface to the GUI toolkit. This module is mostly used to create GUI applications easily.

### ii) Controlling the mouse pointer:

Pyautogui is a GUI automation module. This module is used to control the mouse and keyboard events programmatically. In our system, scrolling of the mouse pointer across the phrases is achieved through `moveTo()` and `moveRel()` methods by giving the relevant x, y location of the screen as arguments.

## • PHRASE SELECTION AND READING:

While the mouse pointer is scrolling across the phrases, user has make voluntary blink (little longer than normal blink) when the mouse pointer reaches the desired button. And if the blink is captured in more than four frames, that button will be clicked and speaker will read the phrase which is already converted from text to speech and loaded into the system.

## CHAPTER 5

### IMPLEMENTATION

#### 5.1. Methodology and Techniques :

- **METHODOLOGY**

The first step was to use a face detection algorithm locate the face on an image frame captured by an ordinary webcam. The next step was to detect only the eyes from this frame. We consider tracking only one eye movement for faster processing time. Then the iris movement was tracked. Since the color of the iris is black, its image has a significantly lower intensity compared to the rest of the eye. This helps us in easy detection of the iris region. Taking the left and right corners of the eye as reference points, the shift of the iris as the person changed his eyes focus was determined. The shift was then used to map cursor location on the test graphical user interface (GUI).

- **EYE TRACKING TECHNIQUES**

There is no universal technique to track the movement of the eyes. In any study, the selection of the technique rests with the actual demands of the application. During the analysis phase of this research, three techniques were analyzed; the Limbus tracking, Pupil tracking, and Electrooculography. Every technique has its own robust points and disadvantages.

- **LIMBUS TRACKING**

Limbus Tracking explains a way of tracking the eye using the limbus. The limbus is the boundary between the white sclera of the eye and the darker iris. As the sclera is white and the iris is darker, this boundary can easily be visually detected as well as tracked. This technique is based on the position and shape of the limbus relative to the head, therefore the head must be kept quite still or the apparatus must be fixed to the user's head. This technique is negatively affected by the eyelid often concealing all or part of the limbus. This makes its uses limited to horizontal tracking. Usually this technique does not involve the use of infra red light.

- **PUPIL TRACKING**

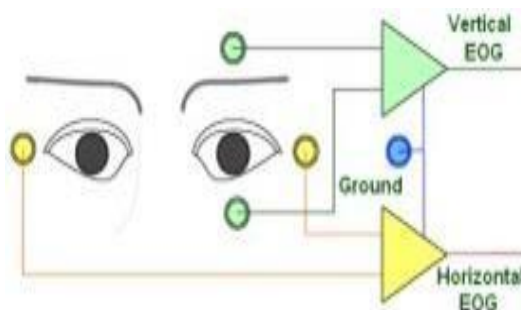
Pupil tracking is a technique of gaze detection that is commonly used often in conjunction with different forms of tracking.



There are several reasons for this; however the main advantage is the notion of the “bright spot”. Like the situation associated with red eye when taking flash photographs at night, infrared can be used in pupil detection to form a high intensity bright spot that is easy to find with image processing. This bright spot occurs when infrared is reflected off the back of the pupil and magnified by the lens. The main advantage of pupil tracking is that as the border of the pupil is sharper than the limbus, a higher resolution is achievable. Also, as the pupil is never really covered by the eyelid, x-y tracking is more feasible as compared to Limbus tracking. The disadvantage is that the difference in contrast is lower between the pupil and iris than between the iris and sclera-thus making the border detection more difficult.

## • ELECTROOCULOGRAPHY

Electrooculography is based on electrodes attached to the human skin. Due to the higher metabolic rate at the retina compared to the cornea, the eye maintains a constant voltage with respect to the retina. This can be approximately aligned with the optical axis. Voltage rotates with the direction of gaze and can be measured by surface electrodes placed on the skin around the eyes. This technique is easily mounted elsewhere other than directly in front of the person as compared to other techniques. Electrical skin potential tracking is often used in medicine and practice to diagnose certain conditions. For example, EOG is employed to diagnose sixth nerve palsy. From their analysis it can be seen that while a clinical orthotic examination is still the best technique of diagnosis. Electrooculography provides a suitable replacement within the follow-up stage of treatment programs. While these uses are beneficial, the utilization of electrodes makes this technique of gaze tracking unsuitable for use in everyday applications.



**FIGURE 5.1 CORRECT POSITIONS OF FIVE ELECTRODES .**

## ➤ SACCADE

A saccade is a fast/rapid movement of an eye. Especially as it jumps from one fixation point to another (as in reading). When something attracts our attention, we position our gaze on it, thus performing a fixation. A fixation usually has duration of at least 100 to 150 milliseconds (ms). The fast eye movements that occur between fixations are known as SACCADES.

## 5.2. CONTROL FLOW

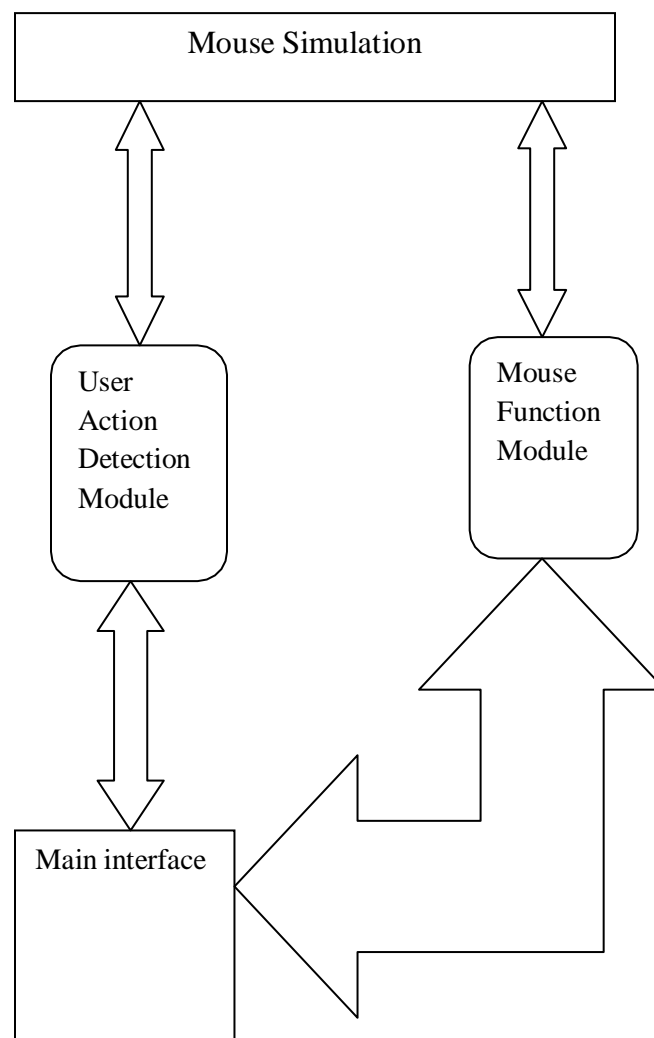


FIG 5.2 CONRTOL FLOW OF MOUSE FUNCTION MODULE

Fig 5.2 Illustrates the system components and interactions of the proposed virtual mouse system. The software application is organized into two parts—the eye tracking engine part and mouse simulation part. The eye tracking engine part matches our software application with eye tracking device (such as Eye Tribe Tracker, which enables the eye to control the mouse cursor) to eye control mouse function. The mouse simulation part defines how the information provided by the mouse cursor is processed. This part consists of components including main interface

Mouse function module: it receives the view point of user from simulation engine and transfers the coordinate to the fixation function within user action detection module, which allows the system to perform directly at the view point after the function is selected. If a second view point is required to finish the action, the view coordinate is transferred to the simulation engine to execute the event. Besides, the module provides various virtual mouse functions for users to operate computers with eye movement.

### 5.3. PSEUDO CODE OR ALGORITHM

```
from imutils import face_utils
from utils import *
import numpy as np
import pyautogui as pag
import imutils
import dlib
import cv2

# Thresholds and consecutive frame length for triggering the mouse action.
MOUTH_AR_THRESH = 0.3
MOUTH_AR_CONSECUTIVE_FRAMES = 10

EYE_AR_THRESH = 0.21

EYE_AR_CONSECUTIVE_FRAMES = 10

WINK_AR_DIFF_THRESH = 0.01
```

WINK\_AR\_CLOSE\_THRESH = 0.21

WINK\_CONSECUTIVE\_FRAMES = 5

# Initialize the frame counters for each action as well as

# booleans used to indicate if action is performed or not

MOUTH\_COUNTER = 0

EYE\_COUNTER = 0

WINK\_COUNTER = 0

INPUT\_MODE = False

EYE\_CLICK = False

LEFT\_WINK = False

RIGHT\_WINK = False

SCROLL\_MODE = False

ANCHOR\_POINT = (0, 0)

WHITE\_COLOR = (255, 255, 255)

YELLOW\_COLOR = (0, 255, 255)

RED\_COLOR = (0, 0, 255)

GREEN\_COLOR = (0, 255, 0)

BLUE\_COLOR = (255, 0, 0)

BLACK\_COLOR = (0, 0, 0)

# Initialize Dlib's face detector (HOG-based) and then create

# the facial landmark predictor

shape\_predictor = "model/shape\_predictor\_68\_face\_landmarks.dat"

detector = dlib.get\_frontal\_face\_detector()

predictor = dlib.shape\_predictor(shape\_predictor)

# Grab the indexes of the facial landmarks for the left and

# right eye, nose and mouth respectively

```
(lStart, lEnd) = face_utils.FACIAL_LANDMARKS_IDXS["left_eye"]  
(rStart, rEnd) = face_utils.FACIAL_LANDMARKS_IDXS["right_eye"]  
(nStart, nEnd) = face_utils.FACIAL_LANDMARKS_IDXS["nose"]  
(mStart, mEnd) = face_utils.FACIAL_LANDMARKS_IDXS["mouth"]
```

# Video capture

```
vid = cv2.VideoCapture(0)  
resolution_w = 1366  
resolution_h = 768
```

cam\_w = 640

cam\_h = 480

unit\_w = resolution\_w / cam\_w

unit\_h = resolution\_h / cam\_h

while True:

```
    # Grab the frame from the threaded video file stream, resize  
    # it, and convert it to grayscale  
    # channels)
```

```
    _, frame = vid.read()
```

```
    frame = cv2.flip(frame, 1)
```

```
    frame = imutils.resize(frame, width=cam_w, height=cam_h)
```

```
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

```
    # Detect faces in the grayscale frame
```

```
    rects = detector(gray, 0)
```

```
    # Loop over the face detections
```

```
    if len(rects) > 0:
```

```
rect = rects[0]
else:
    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF
    continue
    # Determine the facial landmarks for the face region, then

    # convert the facial landmark (x, y)-coordinates to a NumPy
    # array
    shape = predictor(gray, rect)

    shape = face_utils.shape_to_np(shape)

    # Extract the left and right eye coordinates, then use the

    # coordinates to compute the eye aspect ratio for both eyes
    mouth = shape[mStart:mEnd]
    leftEye = shape[lStart:lEnd]
    rightEye = shape[rStart:rEnd]
    nose = shape[nStart:nEnd]
    temp = leftEye
    leftEye = rightEye
    rightEye = temp
    # Average the mouth aspect ratio together for both eyes
    mar = mouth_aspect_ratio(mouth)
    leftEAR = eye_aspect_ratio(leftEye)
    rightEAR = eye_aspect_ratio(rightEye)
    ear = (leftEAR + rightEAR) / 2.0
    diff_ear = np.abs(leftEAR - rightEAR)
```

```
nose_point = (nose[3, 0], nose[3, 1])

# Compute the convex hull for the left and right eye, then
# visualize each of the eyes
mouthHull = cv2.convexHull(mouth)
leftEyeHull = cv2.convexHull(leftEye)
rightEyeHull = cv2.convexHull(rightEye)
cv2.drawContours(frame, [leftEyeHull], -1, YELLOW_COLOR, 1)

cv2.drawContours(frame, [rightEyeHull], -1, YELLOW_COLOR, 1)
for (x, y) in np.concatenate((mouth, leftEye, rightEye), axis=0):
    cv2.circle(frame, (x, y), 2, GREEN_COLOR, -1)

# Check to see if the eye aspect ratio is below the blink
# threshold, and if so, increment the blink frame counter
if diff_ear > WINK_AR_DIFF_THRESH:
    if leftEAR < rightEAR:
        if leftEAR < EYE_AR_THRESH:

            WINK_COUNTER += 1

if WINK_COUNTER > WINK_CONSECUTIVE_FRAMES:

    pag.click(button='left')

    WINK_COUNTER = 0

elif leftEAR > rightEAR:

    if rightEAR < EYE_AR_THRESH:

        WINK_COUNTER += 1

if WINK_COUNTER > WINK_CONSECUTIVE_FRAMES:

    pag.click(button='right')

    WINK_COUNTER = 0
```

else:

WINK\_COUNTER = 0

else:

if ear <= EYE\_AR\_THRESH:

EYE\_COUNTER += 1

if EYE\_COUNTER > EYE\_AR\_CONSECUTIVE\_FRAMES:

    SCROLL\_MODE = not SCROLL\_MODE

    #INPUT\_MODE = not INPUT\_MODE

    EYE\_COUNTER = 0

    # nose point to draw a bounding box around it

else:

EYE\_COUNTER = 0

WINK\_COUNTER = 0



```
if mar > MOUTH_AR_THRESH:
```

```
    MOUTH_COUNTER += 1
```

```
if MOUTH_COUNTER >= MOUTH_AR_CONSECUTIVE_FRAMES:
```

```
    # if the alarm is not on, turn it on
```

```
    INPUT_MODE = not INPUT_MODE
```

```
    #SCROLL_MODE = not SCROLL_MODE
```

```
    MOUTH_COUNTER = 0
```

```
    ANCHOR_POINT = nose_point
```

```
else:
```

```
    MOUTH_COUNTER = 0
```

```
if INPUT_MODE:
```

```
    cv2.putText(frame, "CURSOR STARTED!", (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 0.7,  
    RED_COLOR, 2)
```

```
    x, y = ANCHOR_POINT
```

```
    nx, ny = nose_point
```

```
    w, h = 60, 35
```

```
    multiple = 1
```

```
    cv2.rectangle(frame, (x - w, y - h), (x + w, y + h), GREEN_COLOR, 2)
```

```
    cv2.line(frame, ANCHOR_POINT, nose_point, BLUE_COLOR, 2)
```

```
    dir = direction(nose_point, ANCHOR_POINT, w, h)
```

```
    cv2.putText(frame, dir.upper(), (10, 90), cv2.FONT_HERSHEY_SIMPLEX, 0.7, RED_COLOR, 2)
```

```
    drag = 18
```

```
    ifdir == 'right':
```

```
        pag.moveRel(drag, 0)
```

```
    elifdir == 'left':
```

```
        pag.moveRel(-drag, 0)
```

```
elifdir == 'up':
if SCROLL_MODE:
pag.scroll(40)
else:
pag.moveRel(0, -drag)
elifdir == 'down':
if SCROLL_MODE:

pag.scroll(-40)
else:
pag.moveRel(0, drag)
if SCROLL_MODE:
cv2.putText(frame, 'SCROLL MODE IS ON!', (10, 60), cv2.FONT_HERSHEY_SIMPLEX, 0.7,
RED_COLOR, 2)

cv2.putText(frame, "PRESS ESC TO QUIT", (30, 430),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0,255,0), 2)
cv2.putText(frame, "MAR: {:.2f}".format(mar), (500, 30),

cv2.FONT_HERSHEY_SIMPLEX, 0.7, YELLOW_COLOR, 2)

cv2.putText(frame, "Right EAR: {:.2f}".format(rightEAR), (460, 80),

cv2.FONT_HERSHEY_SIMPLEX, 0.7, YELLOW_COLOR, 2)

cv2.putText(frame, "Left EAR: {:.2f}".format(leftEAR), (460, 130),

cv2.FONT_HERSHEY_SIMPLEX, 0.7, YELLOW_COLOR, 2)

cv2.putText(frame, "Diff EAR: {:.2f}".format(np.abs(leftEAR - rightEAR)), (460, 180),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, YELLOW_COLOR, 2)
# Show the frame
cv2.imshow("Frame", frame)
key = cv2.waitKey(1) & 0xFF
if cv2.waitKey(1) & 0xFF == ord('m'):
INPUT_MODE = not INPUT_MODE
MOUTH_COUNTER = 0
ANCHOR_POINT = nose_point
if cv2.waitKey(1) & 0xFF == ord('s'):
SCROLL_MODE = not SCROLL_MODE
```

## CHAPTER 6

### TESTING

Testing is a dynamic technique of verification and validation. It involves executing an implementation of the software with test data and examining the outputs of the software and its operational behaviour to check that it performing as required.

The following statement serve as the objectives for testing:

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

#### • SOFTWARE DEVELOPMENT LIFE CYCLE

The software development life cycle(SDLC) is a conceptual model used in project management that describes the stages involved in an information system development project from an initial feasibility study through maintenance of the completed application. Software development life cycle is a process used by the software industry to design, develop and test the software.

Phases in Software Development

- Requirement Analysis
- Software Design
- Development or coding
- Testing
- Maintenance

#### 1. Requirement Analysis

The requirements of a desired software product are excited. Based on the business scenario the SRS(Software Requirement Specification) document is prepared in this phase. It is the process of determining user expectations for a system under consideration. These should be quantifiable and detailed. Such requirements are also called functional specifications. It serves as a foundation for test plans and also an agreement between developer and customer. It is the process to make stated and unstated requirements clear and to validate requirement for completeness, unambiguity and feasibility.

## 2. Design

Plans are laid out concerning the physical construction, hardware, operating systems, programming, communications and security issues for the software design. Design phase is concerned with making sure the software system will meet the requirements of the product.

There are 2 stages in design:

- **High Level Design(HLD):** It gives the architecture of the software product to be developed and is done by architects and senior developers.
- **Low Level design(LLD):** It is done by senior developers. It describes how each and every feature in the product should work and how every component should work. Here, only the design will be there and not the code.

## 3. Testing

Testing is evaluating the software to check for the user requirements. Here the software is evaluated with intent of finding effects. It involves executing an implementation of the software with test data and examining the outputs of the software and its operational behaviour to check that it performing as required. It is the process of evaluating a system or its components with the intent to find whether it satisfies the specified requirements or not.

## 4. Maintenance

Once the new system is up and running for a while, it should be exhaustively evaluated. Maintenance must be kept up rigorously at all times. Users of the system should be kept up-to-date concerning the latest modifications and procedures. It is the process of repairing tests so they stay up to date with code changes. It is also important to update your automation framework infrastructure if there are any changes to the tools or third-party libraries that you use. Code changes can happen often in agile companies, even several times a day.

### 6.1. STLC (Software testing Life Cycle)

Testing itself has many phases i.e is called as STLC. It includes

- **Test Plan:** It is a document which describes the testing environment, purpose, scope, objectives, test strategy, schedules, mile stones, testing tool, roles and responsibilities, risks,

training, staffing and who is going to test the application, what type of tests should be performed and how it will tracks the attacks.

- **Test Development:** Preparing test cases, test data, preparing test procedure, preparing test scenario, writing test script.
- **Test Execution:** In this stage, we execute the documents those are prepared I test development phase.
- **Analyze Result:** Once executed documents will get results either pass or fail. We need to analyze the result during the phase.
- **Defect Tracking:** Whenever we get defect on the application, we need to report the bug report file and forwards to test lead team and Dev team. Dev team will fix the bug.

## 6.2. LEVEL OF TESTING USED IN PROJECT

1. **Unit Testing :** Initialization testing is the first level of dynamic testing and is the first responsibility of developers and then that that of the test engineers. Unit testing is performed after the expected test results are met or differences are explainable/acceptable.
2. **Integration Testing:** All module which make application are tested. Integration testing is to make sure that the interaction of two or more components produces results that satisfy functional requirements.
3. **System Testing:** To test the complete system in terms of functionality and non-functionality. It is black box-testing performed by test team, and at the start of the system testing the complete system is configured in a controlled environment.
4. **Functional Testing:** The outgoing links from all the pages from specific domain under test. Test all internal links. Test links jumping on the same pages. Check for the default values of fields. Wrong inputs to the fields in the forms

## 6.3. TEST CASES

Testing is evaluating the software to check for the user requirements. Here the software is evaluated with intent of finding effects. It involves executing an implementation of the software with test data and examining the outputs of the software and its operational behaviour to check that it performing as required.

Test case Number	TC_01
Module Under Test	Face Detection
Description	The method is based on scanning the image of interest with a window that looks for faces at all scales and locations.
Output	Face detection implies pattern recognition, and achieves it with simple methods such as template matching.
Remarks	Test Successful

The above table represents the Test case 01. The module under test case is Face detection has always been a vast research field in the computer vision world. Considering that it is the back bone of any application that deals with the human face. This category of face detection implies pattern recognition, and achieves it with simple methods such as template matching or with more advanced techniques such as neural networks and support vector machines. Before over viewing the face detection algorithm we applied in this work here is an explanation of some of the idioms that are related to it. The expected output is, Face detection implies pattern recognition, and achieves it with simple methods such as template matching, and the actual output is Test is successful.

Test case Number	TC_02
Module Under Test	Eye Detection
Description	Regression approach, Bayesian approach, Discriminative approach techniques are used to detect the eye.
Output	Produces output for patches around each pixel of the input image, from which a prediction will be extracted
Remarks	Test Successful

The above table represents the Test case 02. The module under test case is Eye Detection Regression approach, Bayesian approach, Discriminative approach techniques are used to detect the eye. Tries to minimize the distance between the predicted and actual eye positions. Simply by understanding the functional mapping from the input image to eye positions. The expected output is, patches around each pixel of the input image, from which a prediction will be extracted, and the actual output is Test is successful.

Test case Number	TC_03
Module Under Test	Input From the Eye
Description	Mouse Pointer Control and Mouse Click Events Control are Use to detect the input from the eyes
Output	Left and right movement of the pupil, Eye blink motion detection will be detected
Remarks	Test Successful

The above table represents the Test case 02. The module under test case is Input From the Eye. Now input medium turns to eye movements as a real time. Eye movement input is distinctly faster than other current input. Before the user operates any mechanical pointing device, he or she usually looks at the destination to which he or she wishes to move. Thus the eye movement is available as an indication of the user's goal before he or she could actuate any other input device. The expected output is, Left and right movement of the pupil, Eye blink motion detection will be detected and the actual output is Test is successful.

### ➤ Libraries Used

- Numpy — 1.13.3
- OpenCV — 3.2.0
- PyAutoGUI — 0.9.36
- Dlib — 19.4.0
- Imutils — 0.4.6
- Numpy

- **What is a Python NumPy?**

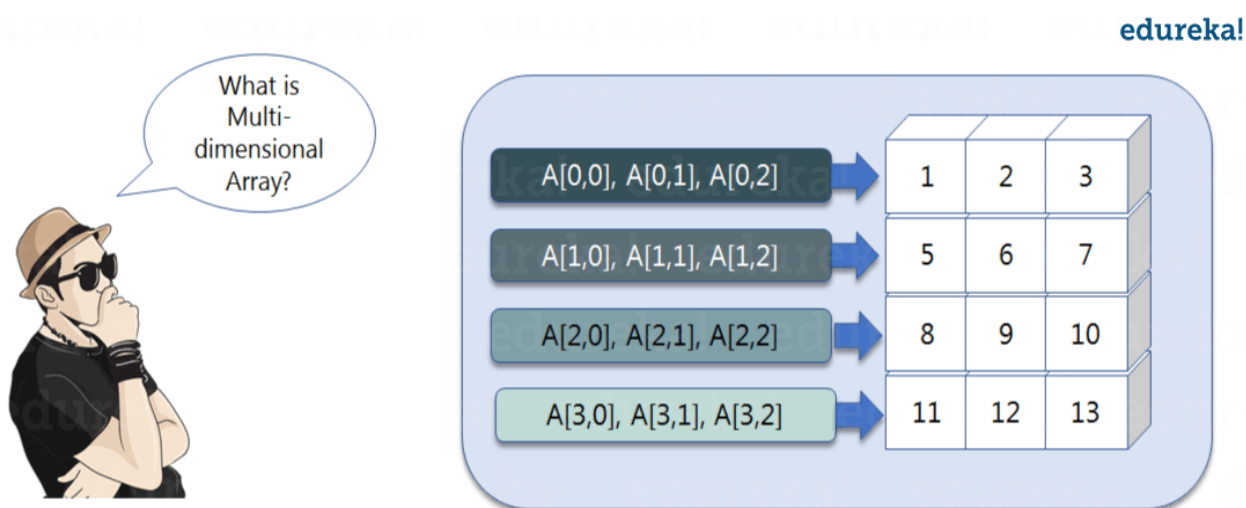
NumPy is a Python package which stands for 'Numerical Python'. It is the core library for scientific computing, which contains a powerful n-dimensional array object, provide tools for integrating C, C++ etc. It is also useful in linear algebra, random number capability etc. NumPy array can also be used as an efficient multi-dimensional container for generic data. Now, let me tell you what exactly is a python numpy array.

**NumPy Array:** Numpy array is a powerful N-dimensional array object which is in the form of rows and columns. We can initialize numpy arrays from nested Python lists and access it elements. In order to perform these numpy operations, the next question which will come in your mind is:

- **How do I install NumPy?**

To install Python NumPy, go to your command prompt and type "pip install numpy". Once the installation is completed, go to your IDE (For example: PyCharm) and simply import it by typing: "import numpy as np"

Moving ahead in python numpy tutorial, let us understand what exactly is a multi-dimensional numPy array.



Here, I have different elements that are stored in their respective memory locations. It is said to be two dimensional because it has rows as well as columns. In the above image, we have 3 columns and 4 rows available.

Let us see how it is implemented in PyCharm:



**Single-dimensional Numpy Array:**

```
1. import numpy as np  
2. a=np.array([1,2,3])
```

**➤ OPENCV**

OpenCV is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it integrated with various libraries, such as Numpy, python is capable of processing the OpenCV array structure for analysis. To Identify image pattern and its various features we use vector space and perform mathematical operations on these features.

The first OpenCV version was 1.0. OpenCV is released under a BSD license and hence it's free for both **academic** and **commercial** use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. When OpenCV was designed the main focus was real-time applications for computational efficiency. All things are written in optimized C/C++ to take advantage of multi-core processing.

**➤ Applications of OpenCV:**

There are lots of applications which are solved using OpenCV, some of them are listed below

- Face recognition
- Automated inspection and surveillance number of people – count (foot traffic in a mall, etc)
- Vehicle counting on highways along with their speeds
- Interactive art installations
- Anamoly (defect) detection in the manufacturing process (the odd defective products)
- Street view image stitching
- Video/image search and retrieval
- Robot and driver-less car navigation and control object recognition.

- Medical image analysis
- Movies – 3D structure from motion

### ➤ **OpenCV Functionality**

- Image/video I/O, processing, display (core, imgproc, highgui)
- Object/feature detection (objdetect, features2d, nonfree)
- Geometry-based monocular or stereo computer vision (calib3d, stitching, videostab)
- Computational photography (photo, video, superres)
- Machine learning & clustering (ml, flann)
- CUDA acceleration (gpu)

### ➤ **Image-Processing**

Image processing is a method to perform some operations on an image, in order to get an enhanced image. If we talk about the basic definition of image processing then “Image processing is the analysis and manipulation of a digitized image, especially in order to improve its quality”.

### ➤ **Digital-Image**

An image may be defined as a two-dimensional function  $f(x, y)$ , where  $x$  and  $y$  are spatial(plane) coordinates, and the amplitude of  $f$  at any pair of coordinates  $(x, y)$  is called the intensity. In another word An image is nothing more than a two-dimensional matrix (3-D in case of coloured images) which is defined by the mathematical function  $f(x, y)$  at any point is giving the pixel value at that point of an image, the pixel value describes how bright that pixel is, and what colour it should be. Image processing is basically signal processing in which input is an image and output is image or characteristics according to requirement associated with that image. Image processing basically includes the following three steps:

1. Importing the image
2. Analysing and manipulating the image
3. Output in which result can be altered image or report that is based on image analysis

**PYAUTO PyAutoGUI is a cross-platform GUI automation Python module for human beings. Used to programmatically control the mouse & keyboard.**

```
pip install pyautogui
```

### ➤ Keyboard and Mouse Control

The x, y coordinates used by PyAutoGUI has the 0, 0 origin coordinates in the top left corner of the screen. The x coordinates increase going to the right (just as in mathematics) but the y coordinates increase going down (the opposite of mathematics). On a screen that is 1920 x 1080 pixels in size, coordinates 0, 0 are for the top left while 1919, 1079 is for the bottom right.

Currently, PyAutoGUI only works on the primary monitor. PyAutoGUI isn't reliable for the screen of a second monitor (the mouse functions may or may not work on multi-monitor setups depending on your operating system and version).

All keyboard presses done by PyAutoGUI are sent to the window that currently has focus, as if you had pressed the physical keyboard key.

### **DLIB**

Dlib is a general purpose cross-platform software library written in the programming language C++. Its design is heavily influenced by ideas from design by contract and component-based software engineering. Thus it is, first and foremost, a set of independent software components. It is open-source software released under a Boost Software License.

Since development began in 2002, Dlib has grown to include a wide variety of tools. As of 2016, it contains software components for dealing with networking, threads, graphical user interfaces, data structures, linear algebra, machine learning, image processing, data mining, XML and textparsing, numerical optimization, Bayesian networks, and many other tasks. In recent years, much of the development has been focused on creating a broad set of statistical machine learning tools and in 2009 Dlib was published in the Journal of Machine Learning Research.[2] Since then it has been used in a wide range of domains.

## CHAPTER 7

# RESULTS AND SNAPSHOTS

## 7.1. PREBUILT MODEL DETAILS

The model offers two important functions. A detector to detect the face and a predictor to predict the landmarks. The face detector used is made using the classic Histogram of Oriented Gradients (HOG) feature combined with a linear classifier, an image pyramid, and sliding window detection scheme. The facial landmarks estimator was created by using Dlib's implementation of the paper: *One Millisecond Face Alignment with an Ensemble of Regression Trees* by Vahid Kazemi and Josephine Sullivan, CVPR 2014. And was trained on the iBUG 300-W face landmark dataset: C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, M. Pantic. 300 faces In-the-wild challenge: Database and results. *Image and Vision Computing (IMAVIS), Special Issue on Facial Landmark Localisation "In-The-Wild"*. 2016.



Source: Vahid Kazemi, Josephine Sullivan. *One millisecond face alignment with an ensemble of regression trees* at <https://ieeexplore.ieee.org/document/6909637>. In CVPR, 2014.

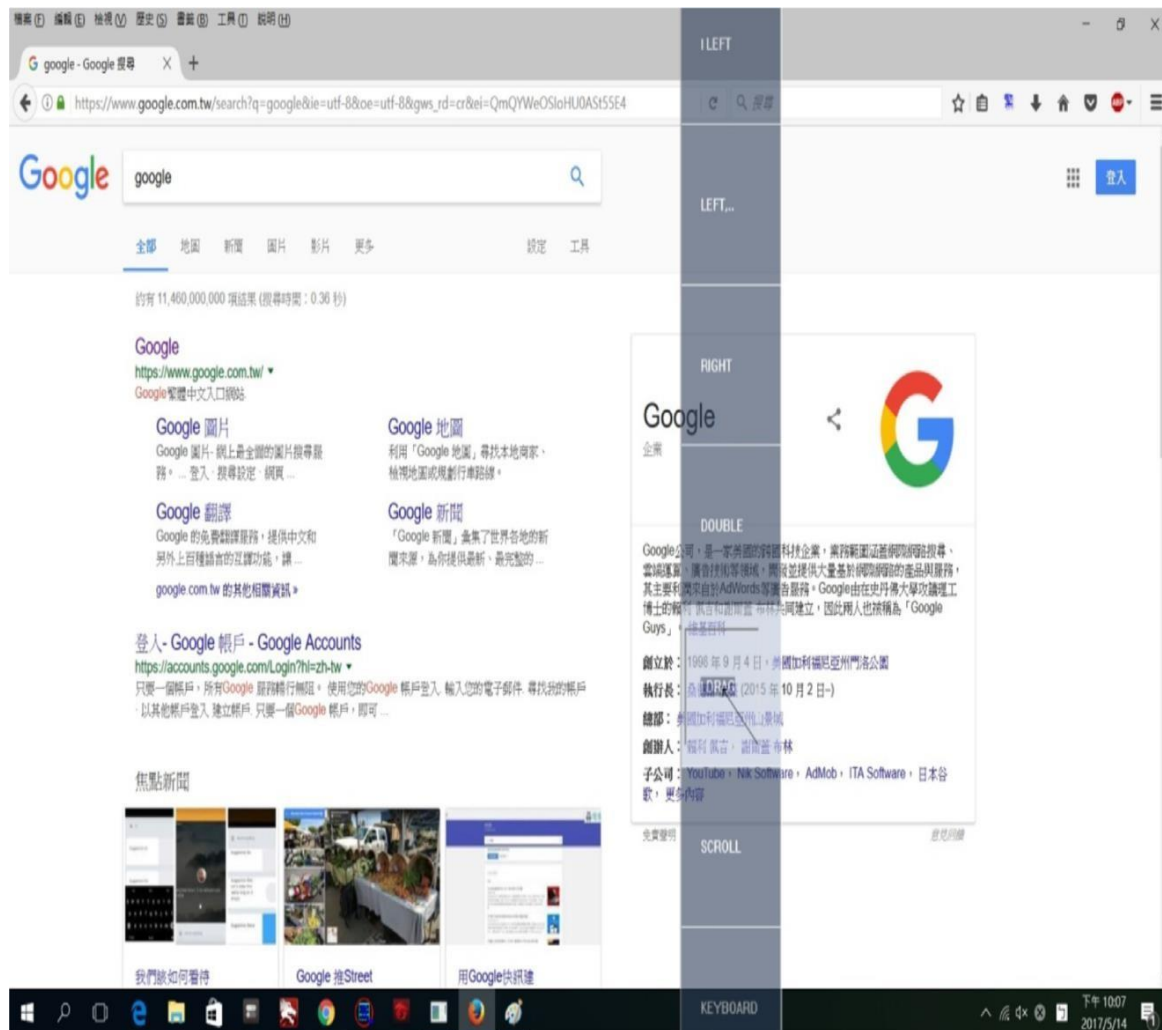


FIG 7.1 THE SCREENSHOT OF THE TOOLBAR FOR THE PROPOSED VIRTUAL MOUSE.

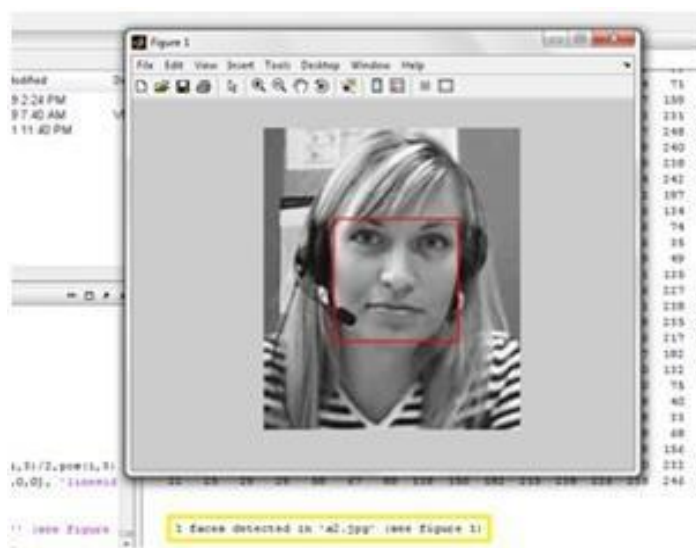
Figure 7.1 illustrates the system components and interactions of the proposed virtual mouse system. The software application is organized into two parts—the eye tracking engine part and mouse simulation part. The eye tracking engine part matches our software application with eye tracking device (such as Eye Tribe Tracker, which enables the eye to control the mouse cursor) to eye control mouse function. The mouse simulation part defines how the information provided by the mouse cursor is processed. This part consists of components including main interface, mouse/keyboard simulation engine, user action detection module, halt (sleep) module, mouse function module, and keyboard function module, as they are presented in Figure 1.1.





**FIG 7.2 SYSTEM SETUP FOR EYE MOUSE.**

In order to capture the face image accurately, the user sat upright with the eye level parallel to the webcam as shown in the Fig. 7.2.



**FIG 7.3 IMAGE DETECTION AND IMAGE CAPTURE**

The image of the user's face is captured using a MATLAB tool called vfm [9], using which images can be captured with the help of a webcam or any other imaging tools attached to the PC. Fig 7.3 shows the image captured by this tool.



**FIG 7.4 EYE REGION EXTRACTION**

In order to extract the eyes region, the image of the face is divided into three equal horizontal areas. The upper third where the eyes are located is extracted as shown Fig 7.4.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
21	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
22	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
23	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
24	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
25	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
26	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
27	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
28	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

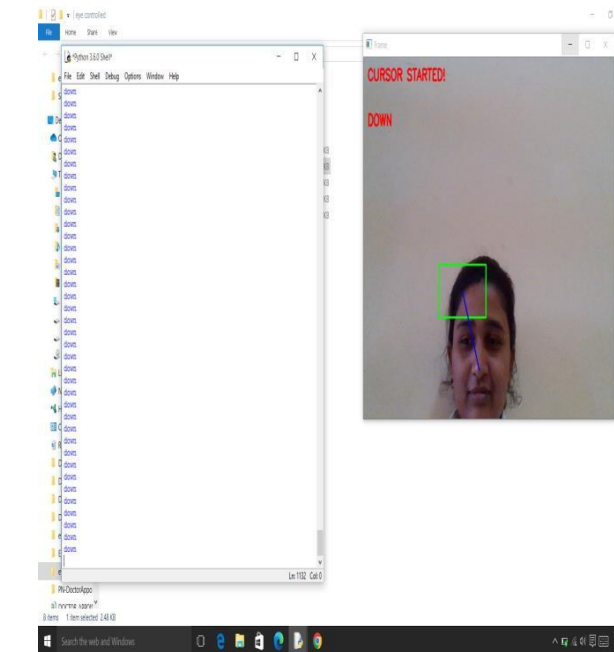
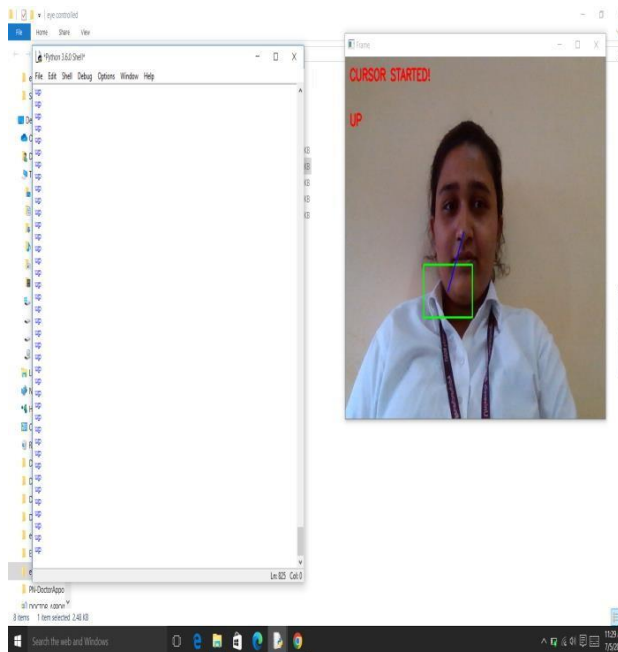
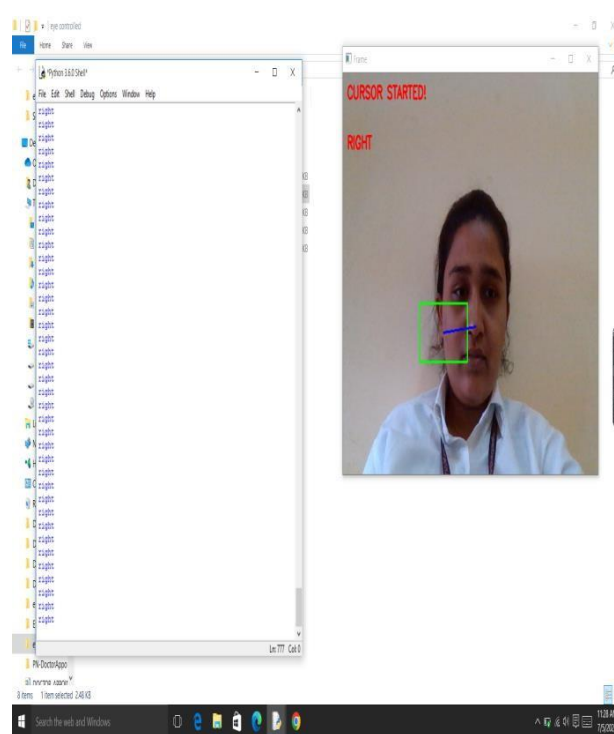
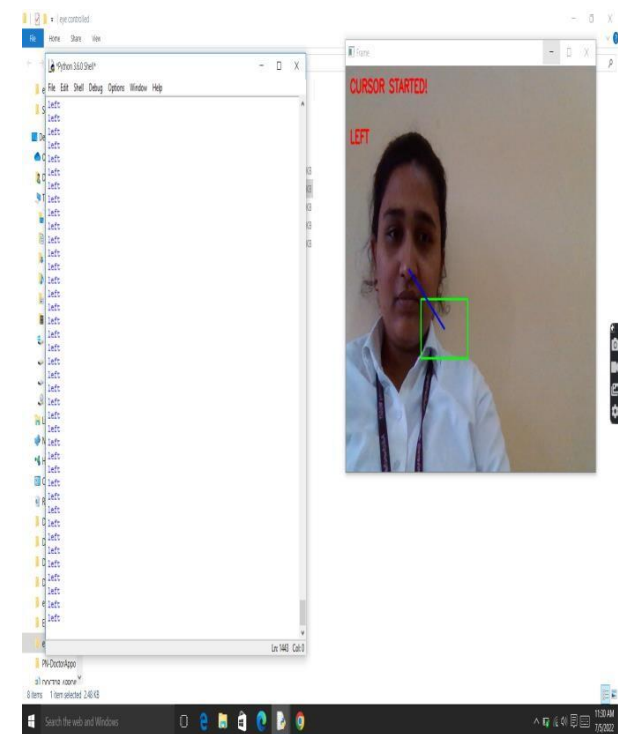
**FIG 7.5 NORMALIZED PIXELS VALUES OF LEFT EYE IMAGE**

The image of one eye was then extracted and normalized in order to remove the background noises and then it was converted to binary image to enhance contrast. The normalized pixel (0 – 1 scale) values of the left eye image is shown in Fig 7.5.

iris =				
0	0	0	0.1000	0.4000
0	0	0	0	0
0	0	0	0	0
0	0	0	0	1.0000
0.9000	0	0	1.0000	1.0000

**FIG 7.6 PIXEL'S INTENSITY VALUES IN THE IRIS AREA**

Since the iris region is black, corresponding pixels' values were very low and when the image was normalized, these values were approximated to zero. Hence the boundaries of iris region were determined as shown in the Fig7.6.



**FIG 7.7 THE SCREENSHOT OF THE USER LEFT, RIGHT, UP, DOWN VIEW POINT.**



Fig 7.7 describes the Mouse function module: it receives the view point of user from simulation engine and transfers the coordinate to the fixation function within user action detection module, which allows the system to perform directly at the view point after the function is selected. If a second view point is required to finish the action, the view coordinate is transferred to the simulation engine to execute the event. Besides, the module provides various virtual mouse functions for users to operate computers with eye movement. There are totally four mouse functions in this module, namely, left-click (LEFT), right-click (RIGHT), up-click (UP), down-click (DOWN). By implementing this process it is cleared that the cursor can be controlled by the eyeball movement i.e., without using hands on the computers. This will be helpful for the people having disability in using the physical parts of the computers to control the cursor points.

## **FUTURE ENHANCEMENT**

Keyboard and mouse actually form an integral part of the computer system. Our system architecture can facilitate the use of computer for the paralyzed people. We have developed a virtual system where people can communicate with the computer without using any physical keyboard and mouse. This could lead to a new age of Human Computer Interaction in which physical contact with the computer would not be necessary at all. The use of object detection and image processing in OpenCV for the implementation of our work has proved to be practically successful and the task of keyboard and mouse is achieved with good precision. This system can be beneficial to certain people who have no control over their limbs. Most of the applications require additional hardware which are often very expensive. The motive of this work is to create this technology as cheaply as possible and to create it under a standardized operating system as well. Though, our system can be used as an alternative for physical keyboard and mouse, it still may perform less accurately in a low light condition. This is a concern for further research. Moreover, the work can be extended for a wide variety of environments and can be tested using the sophisticated existing models. We have tested our system in real time for several times and analysed the results. Our system is not 100% accurate though clicking the phrases with eye blinks are more accurate. Sometimes normal blink also gets detected instead of voluntary blink. However these limitations can be minimized by using high resolution camera.

## CONCLUSION

This paper focused on the analysis of the development of hands-free PC control - Controlling mouse cursor movements using human eyes. Thus, the comprehensive study of the gaze-based interaction processes is implemented. The mouse pointer is operated using eye. The most unique aspect of this system is that it does not require any wearable attachments. This makes the interaction more efficient and enjoyable. A user interface is the system by which human interact with a computer. The user interface includes hardware and software components. No external hardware is attached or required. This system helps the paralyzed people to communicate their ideas and thoughts and needs for them. To differentiate voluntary and normal eye blinks, the eye blink frame rate is used. This algorithm helps the paralyzed people to communicate efficiently. This system doesn't require a person to operate and maintenance is very less. This complete project consists of proposed system, pc or laptop and web cam. The system can be used in many places like hospitals, homes, nursing homes etc. This system provides a new possibility in the life of paralyzed people with eye movement. It helps the paralyzed people to communicate their thoughts through the given phrases in the system. The aim of this system is to reduce the efforts of paralyzed people to communicate their thoughts by using eye movement algorithm. Surely this system will provide a solution for people with severe paralysis.

## REFERENCES

- [1] Sidra Naveed, Bushra Sikander, and Malik Sikander Hayat Khiyal “Eye Tracking System with BlinkDetection”,IEEE,2012.
- [2] Craig Hennessey, Jacob Fiset, “Long Range Eye Tracking: Bringing Eye Tracking into the Living Room”, IEEE,2012
- [3] Bacivarov, Ionita M., Corcoran,P. , “Statistical models of appearance for eye tracking and eye- blink detectionand measurement”, IEEE Transactions, August2010
- [4] Ioana Bacivarov, Mircea Ionita, Peter Corcoran, “Statistical models of appearance for eye tracking and eye blink detection and measurement”, IEEE transactions on consumer electronics, Vol.54 , No.3, pp. 1312-1320 August2009.
- [5] Mohan M. Trivedi, “Simultaneous eye tracking and blink detection with interactive particlefilters” , ACM , January 2010
- [6] Grauman, K.; Betke, M.; Gips, J.; Bradski, G.R. , “Communication via eye blinks- Detection and duration analysis in real-time”, IEEE CONFERENCE PUBLICATIONS, 2009
- [7] Toshiyuki Masui, Koji Tsukada, and Itiro Siio, “MouseField: A Simple and Versatile Input Device for Ubiquitous Computing”, Vol. 3205, Springer (2011), p. 319-328.
- [8] Akhil Gupta, Akash Rathi, Dr. Y.Radhika,“HANDS- FREE PC CONTROL” CONTROLLINGOF MOUSE CURSOR USING EYE MOVEMENT, International Journal of Scientific and Research Publications, Volume 2, Issue4, April 2012
- [9] Jixu Chen, Yan Tong ,Wayne Grayy ,Qiang Jiz “A Robust 3D Eye Gaze Tracking System”,IEEE,2011
- [10] Yingxi Chen, “Design and evaluation of a human-computer interface based on electro-oculography”, 2003,unpublished.URL: vorlon.case.edu/~wsn/theses/ yingxichen\_thesis.pdf
- [11] Arantxa Villanueva ,Rafael Cabeza,Sonia Porta “Eye Tracking System Model With Easy Calibration”,IEEE ,2011
- [12] Arie E. Kaufman, Amit Bandopadhyay, and Bernard D. Shaviv “An Eye Tracking Computer User Interface”, IEEE ,2011
- [13] Takehiko Ohno ,Naoki Mukawa ,Shinjiro Kawato “Just Blink Your Eyes: A Head-Free Gaze Tracking System”, IEEE, 2011

- [14] Shazia Azam, Aihab Khan, M.S.H. Khiyal, “design and implementation of human computer interface tracking system based on multiple eye features” JATIT-journal of theoretical and applied information technology, Vol.9, No.2 Nov, 2009.
- [15] Margrit Betke, James Gips, Peter Fleming, “The Camera Mouse: Visual Tracking of Body features to Provide Computer Access for People With Severe Disabilities”, IEEE Transactions On Neural Systems And Rehabilitation Engineering, Vol.10, No.1, March 2008.
- [16] H.T. Nguyen, “Occlusion robust adaptive template tracking.”, Computer Vision, 2001. ICCV 2001 Proceedings International Journal of Scientific and Research Publications, Volume 2, Issue 4, April 2012
- [17] M. Betke, “the camera mouse: Visual Tracking of Body Features to provide Computer Access For People With Severe Disabilities.” IEEE Transactions on Neural Systems and Rehabilitation Engineering. VOL. 10. NO 1. March 2002.
- [18] Abdul Wahid Mohamed, “Control of Mouse Movement Using Human Facial Expressions” 57, Ramakrishna Road, Colombo 06, Sri Lanka.
- [19] Arslan Qamar Malik, “Retina Based Mouse Control (RBMC)”, World Academy of Science, Engineering and Technology 31, 2007
- [20] G. Bradski, “Computer Vision Face Tracking for Use in a Perceptual User Interface”. Intel Technology 2nd Quarter, 98 Journal.
- [21] A. Giachetti, Matching techniques to compute image motion, Image and Vision Computing 18(2000). Pp. 247-260.
- [22] Sunita Barve, Dhaval Dholakiya, Shashank Gupta, Dhananjay Dhatrak, “Facial Feature Based Method For Real Time Face Detection and Tracking I-CURSOR”, International Journal of Engg Research and App., Vol. 2, pp. 1406-1410, Apr (2012).
- [23] Yu-Tzu Lin, Ruei-Yan Lin, Yu-Chih Lin, Greg C Lee “Real-time eye-gaze estimation using a low-resolution webcam”, Springer, pp. 543-568, Aug (2012).
- [24] Samuel Epstein-Eric Missimer, Margrit Betke “Using Kernels for a video based mouse-replacement interface”, Springer link, Nov (2012)
- [25] Hossain, Zakir, Md Maruf Hossain Shuvo, and Prionjit Sarker. "Hardware and software implementation of real time electrooculogram (EOG) acquisition system to control computer cursor with eyeball movement." In 2017 4th International Conference on Advances in Electrical Engineering (ICAEE), pp. 132-137. IEEE, 2017.

- [26] Mohamed Nasor, K KMujeeb Rahman, Maryam Mohamed Zubair, Haya Ansari, Farida Mohamed. "Eye- controlled mouse cursor for physically disabled individual" , 2018 Advances in Science and Engineering Technology International Conferences (ASET), 2018.
- [27] K. H. Shibly, S. Kumar Dey, M. A. Islam and S. IftekharShowrav, "Design and Development of Hand Gesture Based Virtual Mouse," 2019 1st International Conference on Advances inScience, Engineering and Robotics Technology (ICASERT), Dhaka, Bangladesh, 2019, pp. 1-5, doi: 10.1109/ICASERT.2019.8934612.
- [28] “Hand Gesture Recognition System Using Image Processing” Sagar P. More and Abdul Sattar,
- [29] International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) – 2016.
- [30] Neil Castellino and Michelle Alva, “An image-based eye controlled assistive system for paralytic patients”, IEEE Conference Publications, 2017.
- [31] Shu-Fan Lin, Xuebai Zhang, Shyan-Ming Yuan, "Eye Tracking Based Control System for Natural Human-Computer Interaction", Computational Intelligence and Neuroscience, 2017.
- [32] ‘Cech and Soukupova’, “Real-Time Eye Blink Detection using Facial Landmarks”, Center for Machine Perception, February 2016.