HUMAN COMPUTER INTERACT BASED EYE CONTROLLED MOUSE

An Mini Project Report Submitted
In partial fulfillment of the requirement for the award of the degree of

Bachelor of Technology
in
Computer Science and Engineering (Internet of Things)

by

L.SREENIVAS REDDY 20N31A6933

Under the Guidance of
Dr. I NAGARAJU
PROFESSOR
Department of Emerging Technologies
MRCET (Autonomous Institution, UGC Govt. of India)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING-(INTERNET OF THINGS)
(EMERGING TECHNOLOGIES)
MALLA REDDY COLLEGE OF ENGINEERING AND TECHNOLOGY
(Autonomous Institution – UGC, Govt. of India)
(Affiliated to JNTU, Hyderabad, Approved by AICTE, Accredited by NBA & NAAC – 'A' Grade, ISO 9001:2015 Certified)
Maisammaguda (v), Near Dullapally, Via: Kompally, Hyderabad – 500 100, Telangana State, India

2023-2024

DECLARATION

I hereby declare that the project entitled **"HUMAN COMPUTER INTERACT BASED EYE CONTROLLED MOUSE"** submitted to **Malla Reddy College of Engineering and Technology,** affiliated t**o** Jawaharlal Nehru Technological University Hyderabad (JNTUH) as part of IV Year B.Tech – I Semester and for the partial fulfillment of the requirement for the award of **Bachelor of Technology** in **Computer Science and Engineering (Internet Of Things)** is a result of original research work done by us.

It is further declared that the project report or any part thereof has not been previously submitted to any University or Institute for the award of degree or diploma.

L.SREENIVAS REDDY - 20N31A6933

CERTIFICATE

This is to certify that this is the bonafide record of the project titled **" HUMAN COMPUTER INTERACT BASED EYE CONTROLLED MOUSE"** submitted by **L.SREENIVAS REDDY(20N31A6933),** of **B.Tech IV Year – I Semester** in the partial fulfillment of the requirements for the degree of **Bachelor of Technology** in **Computer Science and Engineering (Internet of Things)**, Dept. of CSE (Emerging Technologies) during the year 2023-2024. The results embodied in this project report have not been submitted to any other university or institute for the award of any degree or diploma.

Dr. I NAGARAJU
Project Guide
Department of CSE (ET)

Mrs.V.BALA
Project Coordinator
Department of CSE (ET)

Dr.DILEEP
OVERALL COORDINATOR

EXTERNAL EXAMINER

Dr.M V Kamal
HEAD
OF THE DEPARTMENT

Date of Viva-Voce Examination held on:_____

ACKNOWLEDGEMENTS

I feel my self honored and privileged to place my warm salutation to our college "Malla Reddy College of Engineering and Technology (Autonomous Institution – UGC Govt. of India) and our Principal **Dr. S Srinivasa Rao,** Professor who gave us the opportunity to do the Mini project (Human computer interact based eye controlled mouse) during my IV Year B.Tech I Semester and profound the technical skills.

I express my heartiest thanks to my Director **Dr. V S K Reddy,** Professor for encouraging us in every aspect of our project and helping us realize our full potential.

I also thankful to my Head of the Department **Dr. M V Kamal,** Professor for providing training and guidance, excellent infrastructure and a nice atmosphere for completing this project successfully.

I would like to express my sincere gratitude and indebtedness to our project supervisor
Dr. I NAGARAJU, Professor for his valuable suggestions and interest throughout the course of this project.

I convey my heartfelt thanks to our Project Coordinator **Dr. P Dileep,** Professor for allowing for their regular guidance and constant encouragement during my dissertation work.

I would like to thank all our supporting **staff** of the Department of CSE (Emerging Technologies) and even all other department who have been helpful directly and in-directly in making our project a success.

Finally, I would like to take this opportunity to thank my **family** for their support and blessings for completion of my project that gave me the strength to do my project.

L.SREENIVAS REDDY - 20N31A6933

ABSTRACT

Eye Movement can be regarded as a pivot real-time input medium for human computer communication, which is important for people with physical disability. The proposed system focuses on providing a simple and convenient interactive mode by only using user'seye. The system built is an eye based interface that acts as a computer mouse to translate eye movements such as blinking, gazing, and squinting towards the mouse cursor actions. In this project two interactive tasks with different difficulty were done to compare the proposed eye control tool with an existing system. This paper presents a HCI system thatis great important to amputees and those who have issues with using their hands. The system in discussion makes use of simple webcam and its software requirements are Python, OpenCV, NumPY and a few other packages. An algorithm to carry out the functionsof a mouse by providing a hand free interaction between humans and computers by using different expression of a face using computer vision and matching it with already stored expression. The "Camera Mouse" system tracks the computer user's movements with a video camera and translates them into the mouse movements of the mouse pointer on the screen.

TABLE OF CONTENTS

S. No. Topic Page No.

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION :

Eye tracking technology, which is based on an eye tracker that measures the movement and positions of the eye has played an increasingly important role in psychology, marketing and user interfaces. Eye trackers have existed for a number of years, but, early in the development of the field of eye tracking, the use of eye trackers was largely confined to laboratory experiments to observe the nature of human eye movements, rather than to use these movements as an actual control medium within a human-computer interaction (HCI).

As eye tracking gear gets cheaper, new applications with the concept of using eye tracking in HCI are clearly beginning to blossom.Traditional user interfaces provide much more bandwidth from computer to user, such as images, animations, videos, and other media which can output large amounts of information rapidly. This is one of our mainly input mediums, and about 80 to 90 percent of the outside world information is obtained from the human eye. For multimedia communication from user tocomputer, the eye movements can be regarded as a pivotal real-time input medium, which is especially important for people with motor disability (such as persons with Amyotrophic Lateral Sclerosis).

By installing an eye tracker and using its, coordinate output stream as a virtual mouse, the moving of user's gaze would directly cause the mouse cursor to move. But the natural hand moving a mouse is very different from the eye movement to control virtual mouse. There are significant differences between the mouse and eye position to be considered in designing eye tracking based control system for user-computer dialogue.

The system combines both the mouse functions and keyboard functions, so that users can use our system to achieve almost all of the inputs to the computer without traditional input equipment. It was oriented towards improving the reliability, mobility, and usability for user to interact with computer by only using their eyes.

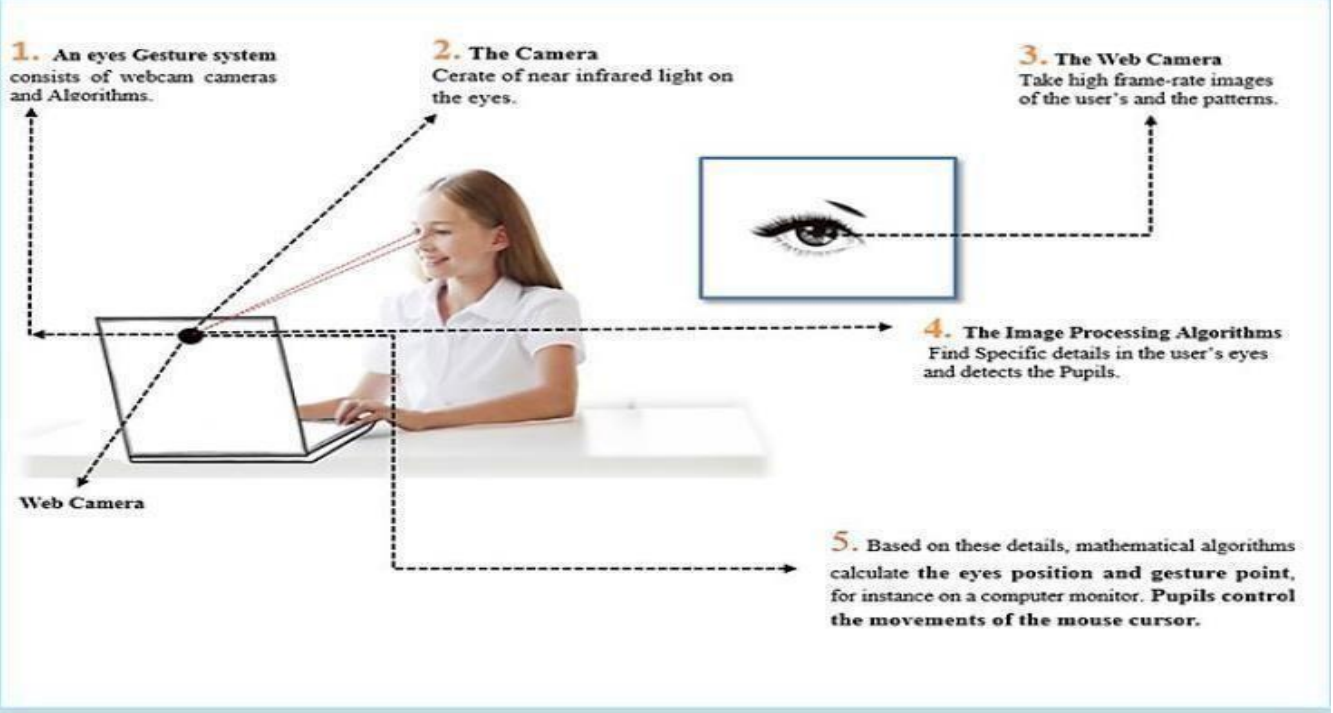

fig 1: capturing of eyes

1.2 MOTIVATION:

In today's digital age, computer technology has become an integral part of our daily lives. We use computers for various tasks, from communication and entertainment to work and education. However, for individuals with physical disabilities, the conventional means of interacting with computers, such as using a keyboard and mouse, can be a significant challenge. This is where the motivation for an eye-based human-computer interaction (HCI) system becomes evident. The proposed system aims to provide a simple and convenient means of computer interaction solely through the use of a user's eyes. Such a system has the potential to revolutionize the way individuals with physical disabilities access and control computers.

The primary motivation behind developing an eye-based HCI system is to enhance the quality of life

for individuals with physical disabilities, particularly those who may be amputees or have limitations in hand function. These individuals often face substantial barriers in accessing the digital world, which can have profound implications on their personal, educational, and professional lives. By creating an eye-based interface that effectively acts as a computer mouse, we are empowering these individuals with a newfound sense of independence and inclusivity in the digital realm.

1.3 LITERATURE REVIEW :

The system consists of a standard electric wheelchair with an on-board computer, sensors and a graphic user interface run by the computer. On the other hand, this eye-control method can be applied to handle graphical interfaces, where the eye is used as a mouse computer. Results obtained show that this control technique could be useful in multiple applications, such as mobility and communication aid for handicapped persons.

The aim and scope of the journal is to emphasize research, development and application within the fields of Scientific Research Engineering &Technology that supporthigh-level of learning, teaching, development and research. It is an international journal that aims to contribute to the constant research and training to promote research in the relevant field.

If the user sees the monitor, the center of a pupil is always in a polygon that is madeby the glints. Consequently, the direction of the user's eye gaze can be computed without computing the geometrical relation between the eye, the camera and the monitor in 3D space. Our method is comparatively simple and fast. We introduce the method and show some experimental results.

1.4 PROBLEM DEFINITION:

The proposed system aims to address a critical issue in human-computer interaction: providing a real-time input medium for individuals with physical disabilities, particularly those who have limited or no use of their hands. The central problem at hand is the need for a simple and convenient interactive mode that relies solely on a user's eye movements. This eye-based interface is designed to function as a computer mouse, enabling users to control their computers through actions such as blinking, gazing, and squinting. The system serves as an innovative solution that can significantly enhance the accessibility and usability of computers for people with physical impairments.

1.5 OBJECTIVE OF THE PROJECT:

The objective of the Human-Computer Interaction (HCI) project based on an eye-controlled mouse system is to enhance accessibility and usability for individuals with physical disabilities. This project aims to create a user-friendly and responsive interface that empowers users to interact with computers solely through their eye movements. The primary goals include enabling independent computer usage for tasks like web browsing, communication, and gaming, while minimizing user fatigue and reducing errors. Customization, integration with existing software, and cross-platform compatibility are essential considerations. The project seeks to contribute to the broader field of assistive technology, making computer interaction more inclusive and user-centric, and ultimately improving the quality of life for those with physical limitations.

CHAPTER 2

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM:

- Currently eye tracking mouse technology is not available at a large scale.The computer mouse or moving the finger has been a very common approach to move the cursor along the screen in the current technology.
- The system detects the movement in the mouse or the finger to map it to the movement of cursor.
- The people with disability will not be able to make use of the current technology.
- 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.

PROPOSED SYSTEM:

The system proposed in this works based on the following action:

- Squinting your eyes
- Winking
- Blinking
- Gazing

The methodology is as follows:

- Since the project is based on detecting the features of the face and mapping them to thecursor, the webcam needs to be accessed first, which means that the webcam will be opened.
- The frame-rate of the video is generally around 30 frames per second, so a frame at every 1/30th of a second will be used to be processed.
- This frame undergoes a set of processes before the features of the frame are detected and mapped to the cursor. And this process continuously takes place for every frame asa part of a loop.
- Once the frame is extracted, the regions of the face need to be detected. Hence, the frames will undergo a set of image-processing functions to process the frame in a suitable way, so that it is easy for the program to detect the features such as eyes, mouth,nose, etc.
- Real time eye tracking and eye gaze estimation is achieved through eye based humancomputer interaction provide.

2.2 FUNCTIONAL REQUIREMENTS(HARDWARE AND SOFTWARE):

- Functional requirement should include function performed by a specific screen outline work-flows performed by the system and other business or compliance requirement the systemmust meet.
- Functional requirements specify which output file should be produced from the given file they describe the relationship between the input and output of the system, for each functional requirement a detailed description of all data inputs and their source and the range of valid inputs must be specified.
- The functional specification describes what the system must do, how the system does itis described in the design specification.
- If a user requirement specification was written, all requirements outlined in the user requirements specifications should be addressed in the functional requirements.

- Eye detection
- Eye Moment Detection
- Mouth Moment Detection
- Head Moment Detection
- Mouse Cursor Controlling

Hardware Requirements:

- Processor : Intel i3 and above
- RAM : 4GB and Higher
- Hard Disk : 500GB Minimum

Software Requirements:

- Programming Language : Python
- IDE : pycharm/jupyter/VScode

CHAPTER 3
SOFTWARE ENVIRONMENT

3.1 PYTHON:

Python is an interpreter, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse.

The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception.

When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a programis to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

Advantages of Python :

Let’s see how Python dominates over other languages.

1. Simple and Easy:

When working with Java, you may have to create a class to print **‘Hello World’**. But in Python, just a print statement will do. It is also quite **easy to learn, understand,** and **code.** This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

2. Readable:

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and **indentation is mandatory.** This further aids the readability of the code.

3. Object-Oriented:

This language supports both the **procedural and object-oriented** programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

Disadvantages of Python:

So far, we've seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let's now see the downsides of choosing Python over another language.

1. Speed Limitations:

We have seen that Python code is executed line by line. But since Python is interpreted, it often results in **slow execution**. This, however, isn't a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

2. Weak in Mobile Computing and Browsers:

While it serves as an excellent server-side language, Python is much rarely seen on the **clientside**. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called **Carbonnelle**.

3.2 MODULES:

Resizing:

The image is first flipped over the y-axis. Next, the image needs to be resized. The resize function refers to setting the new resolution of the image to any value as per the requirement. In this project, the new resolution is 640 X 480.

BGR to Gray:

The data that we are using to detect the different parts of the face requires image ofa grayscale format to give more accurate results. Hence, the image, i.e. the frame of the video from the webcam needs to undergo the process of converting its format from RGB tograyscale. Once the image is converted to a grayscale format, it can be used to locate the face and identify the features of the face.

Detection and Prediction of facial features:

- To detect the face and the features, a prebuilt model is used in the project, which has the available values that can be interpreted by python to make sure that the face is located in the image. There is a function called ‘detector()’, made available by the models,which helps us to detect the face. After the face is detected, the features of the face can now be detected,using the function ‘predictor’.
- The function helps us to locate 68 points on any 2D image. These points correspond to different points on the face near the required parts such as eyes, mouth, etc. The values of the function that are obtained are in the form of 2D coordinates. Every one of the 68 points are essentially values of the x and y coordinates that, when connected, will roughly form an identifiable face.
- They are then stored as an array of values so that they can be arranged and used in the next step to connect any of the coordinates and draw a boundary to represent the requiredregions of the face. Four sets of arrays are taken as 4 different parts of these values which are stored in the array, to separately be stored as the coordinates to be connected to representthe required regions, those are the: Left eye, Right eye, nose and the mouth. Once the 4 arrays are prepared, boundaries, or ‘contours’ are drawn around the points using 3 of these arrays by connecting these points, using the ‘drawcontour’ function and the shape formed is around the two eyes and the mouth.

Mouth and Eye Aspect ratio:

Once the contours are drawn, it is necessary to have a reference for the shapes, which, when compared with, gives the program any information about any action made bythese regions such as blinking, yawing, etc. These references are understood as ratios, between the 2D coordinates, and a change in the coordinates, essentially tell us that, the parts of the region of the face have moved from the regular position and an action has beenperformed.

The system is built on predicting facial landmarks of the face. The Dlib prebuiltmodel helps in fast and accurate face detection along with 68 2D facial landmarks as explained already. Here, Eye-AspectRatio (EAR) and mouth-aspect-ratio (MAR) are used to detect blinking/winking and yawing respectively. These actions are translated into mouseactions.

Similarly, the MAR goes up when the mouth opens. This is used as an action to startand switch off the mouse. For example, if the ratio has increased, it can mean that the distances between the points representing the region of the face have changed and an actionhas been performed by the person. This action is supposed to be understood as the person trying to perform an operation using the mouse. Hence, for these functionalities to be madeoperational, there need to be some defined ‘aspect_ratios’, which when cross a defined limit, interprets an action being performed.

Detection of Actions Performed by the face:

After the ratios are defined, the frame can now compare the ratios of the parts of theface with the ratios defined for different actions, of the current frame being processed.

It is done using the ‘if’ statement. The actions which the program identifies are:

i. **For activating the Mouse:** The user needs to ‘yaw’ which is opening his mouth, vertically, in turn increasing the distance between the corresponding 2D points of the mouth. The algorithm detects the change in the distance by computing the ratio, and when this ratio crosses a specified threshold, the system is activated and the cursor can be moved.

The user needs to place his nose towards, either the top, bottom, left or right of a rectangle that appears, to move the cursor in the corresponding direction. The more he is away from the rectangle, the faster is the movement of the cursor.

ii. **Left/Right Clicking:** For clicking, he needs to close any one of his eye, and make sure to keep the other open. The program first checks whether the magnitude of the difference is greater than the prescribed threshold by using the difference between the ratiosof the two eyes, to make sure that the user wants to perform either the left or right click, and does not want to scroll(For which both the eyes need to squint).

iii. **Scrolling:** The user can scroll the mouse, either upwards or downwards. He needs to squint his eyes in such a way that the aspect ratio of both the eyes is less than the prescribed value. In this case, when the user places his nose outside the rectangle, the mouse performs scroll function, rather than moving the cursor. He can move his nose either above the rectangle to scroll upwards, or move it below the rectangle to scroll downwards.

CHAPTER 4
SYSTEM DESIGN AND UML DIAGRAMS

4.1 DATA FLOW DIAGRAM:

1. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
2. Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation.
3. Data flow diagrams can be divided into logical and physical. The logical data flow diagram describes flow of data through a system to perform certain functionality of a business. The physical data flow diagram describes the implementation of the logical data flow.

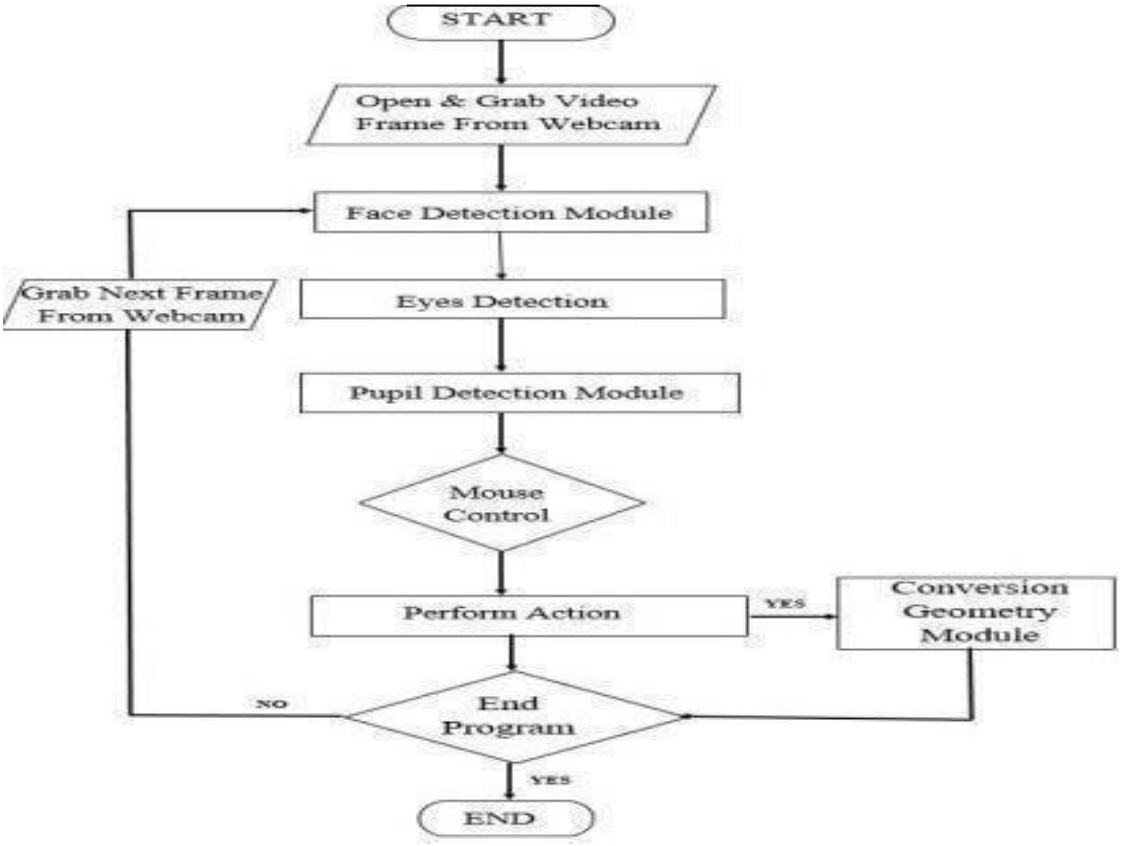

fig 2: data flow diagram

4.2 SYSTEM ARCHITECTURE:

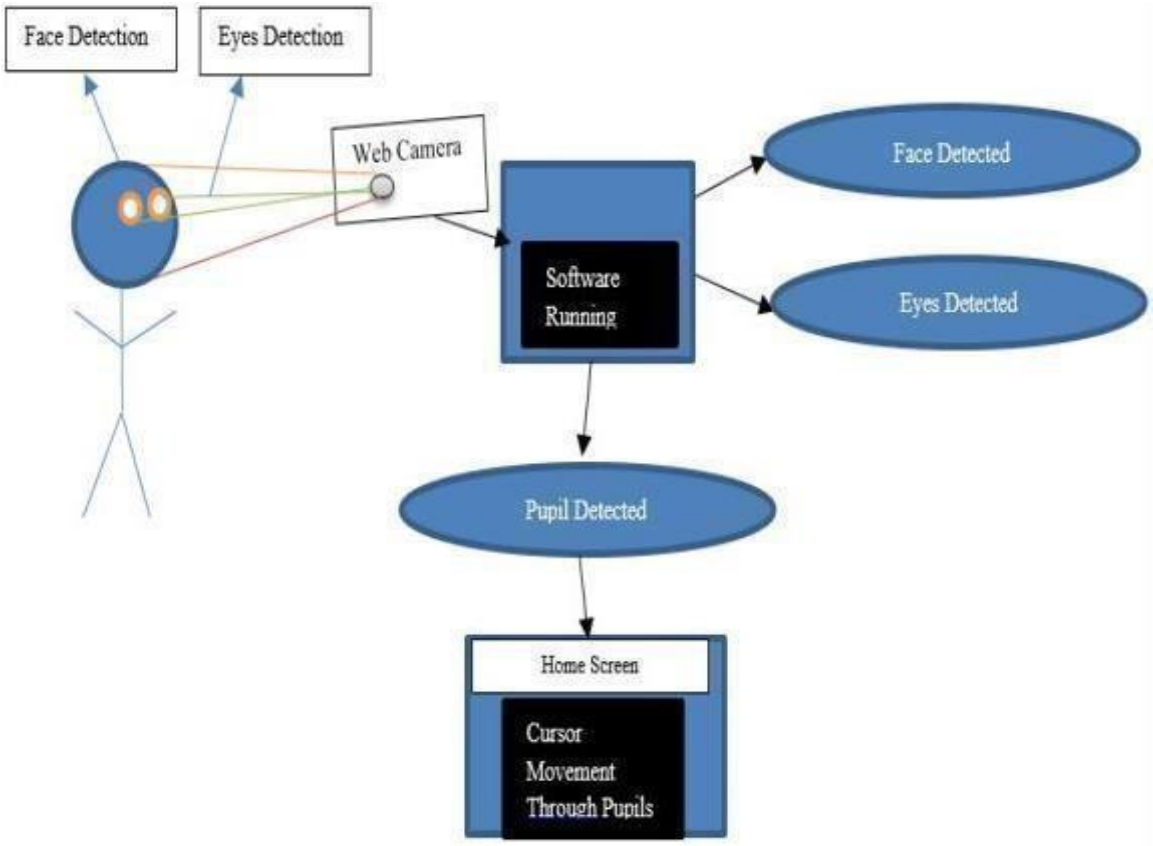

fig 3: system architecture

4.3 UML DIAGRAMS:

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling

and other non-software systems. The UML represents a collection of bestengineering practices that have proven successful in the modeling of large and complex systems.

Goals:

The Primary goals in the design of the UML are as follows:

- Provide users a ready-to-use, expressive visual modeling Language so that they candevelop and exchange meaningful models.
- Provide extendibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development process.
- Provide a formal basis for understanding the modeling language.
- Support higher level development concepts such as collaborations, frameworks,patterns and components.

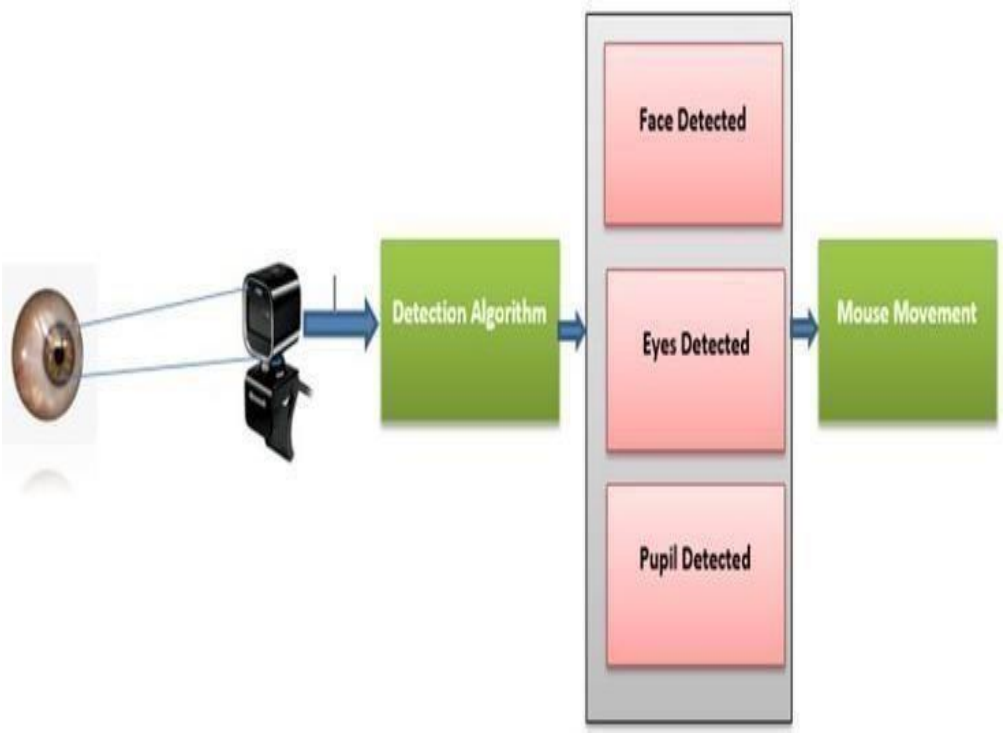

fig 4: detecting the facial expressions

4.3.1 USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals(represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

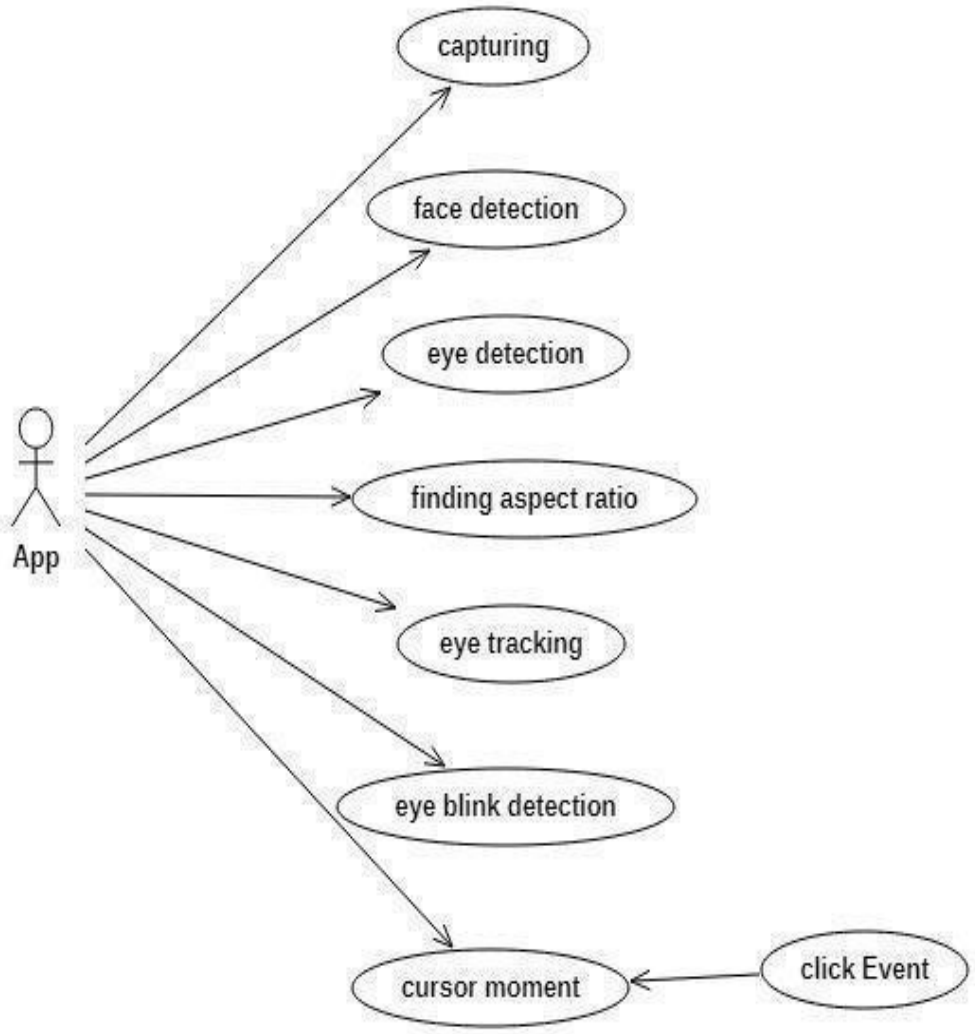

fig 5: use case diagram for cursor

4.3.2 CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

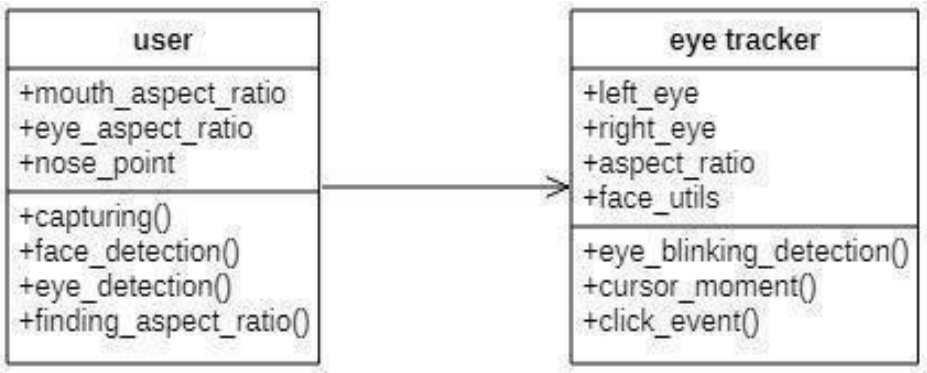

fig 6: class diagram for cursor movements

4.3.3 SEQUENCE DIAGRAM:

Sequence Diagrams Represent the objects participating the interaction horizontally and time vertically. A Use Case is a kind of behavioral classifier that represents a declaration of an offered behavior. Each use case specifies some behavior, possibly including variants thatthe subject can perform in collaboration with one or more actors. Use cases define the offered behavior of the subject without reference to its internal structure. These behaviors, involving interactions between the actor and the subject, may result in changes to the state of the subject and communications with its environment. A use case can include possible variations of its basic behavior, including exceptional behavior and error handling.

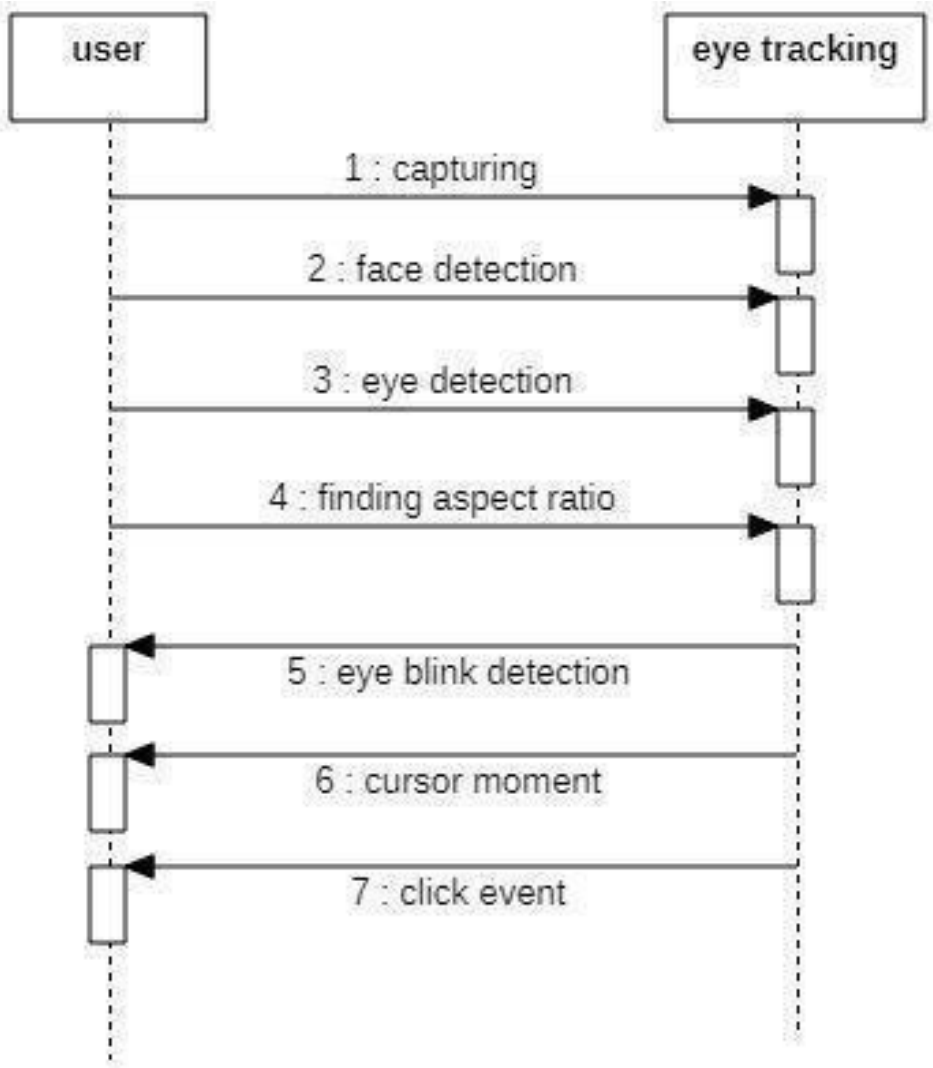

fig 7: sequence diagram for cursor movements

4.3.4 ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of Workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step by step workflows of components in a system. An activity diagram shows the overall flow of control.

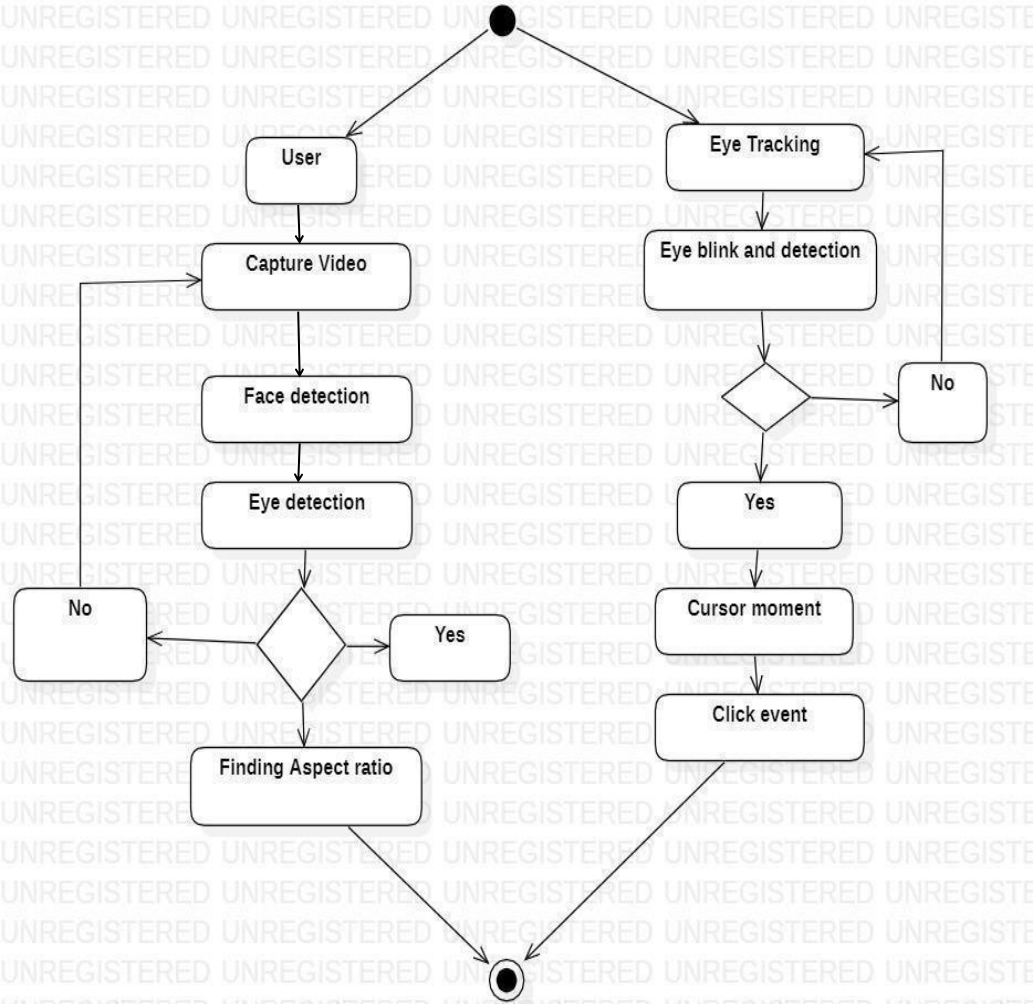

fig 8: activity diagram for cursor movements

4.3.5 STATE CHART DIAGRAM:

State chart diagram is one of the five UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states arechanged by events. State chart diagrams are useful to model the reactive systems. Reactivesystems can be defined as a system that responds to external or internal events.

State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of State chart diagram is to model lifetime of Following are the main purposes of using Statechart diagrams −

- To model the dynamic aspect of a system.

- To model the life time of a reactive system.
- To describe different states of an object during its life time.
- Define a state machine to model the states of an object.

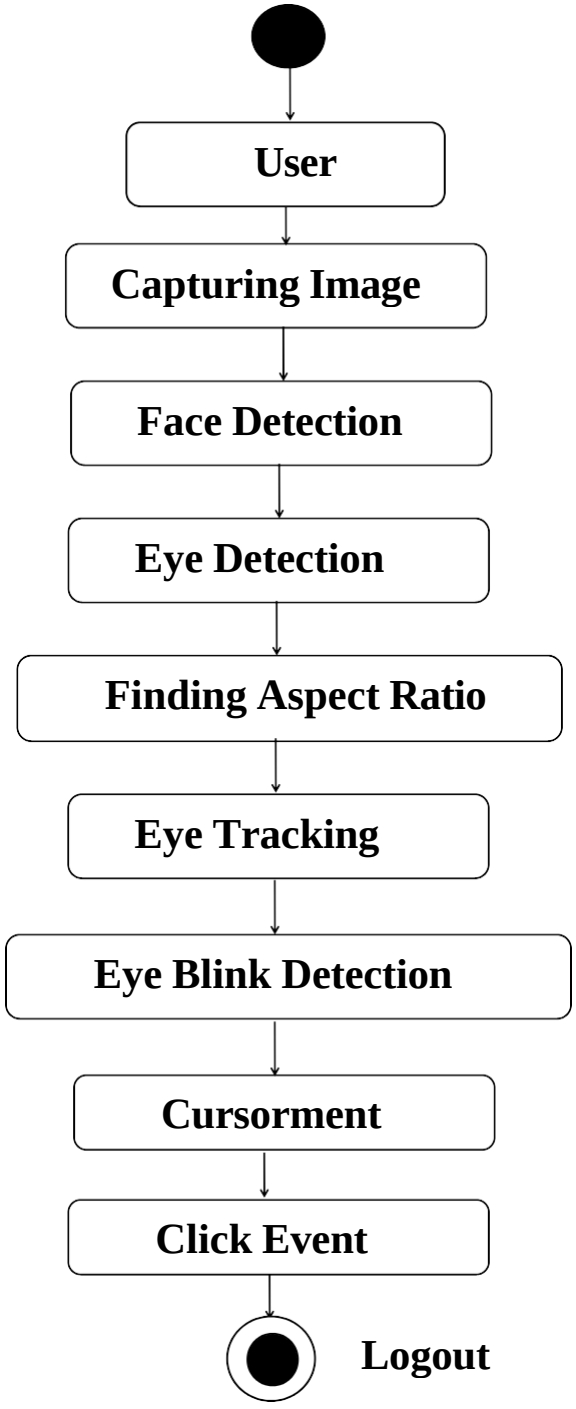

fig 9: state chart diagram for cursor movements

CHAPTER 5

SOFTWARE DEVELOPMENT LIFE CYCLE

The Systems Development Life Cycle (SDLC), or Software Development Life Cycle in systems engineering, information systems and software engineering, is the process of creating or altering systems, and the models and methodologies use to develop these systems.

fig 10: sdlc model

5.1 PHASE OF SDLC

TESTING:

In this phase the system is tested. Normally programs are written as a series of individual modules, this subject to separate and detailed test. The system is then tested as a whole. Theseparate modules are brought together and tested as a complete system. The system is tested to ensure that interfaces between modules work (integration testing), the system works on the intended platform and with the expected volume of

data (volume testing) and that the system does what the user requires (acceptance/beta testing).

5.2 MAINTENANCE:

Inevitably the system will need maintenance. Software will definitely undergo change once it is delivered to the customer. There are many reasons for the change. Change could happenbecause of some unexpected input values into the system. In addition, the changes in the system could directly affect the software operations. The software should be developed to accommodate changes that could happen during the post implementation period.

5.3 SDLC METHODOLOGIES:

This document play a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basic during testing phase. Any changes made to the requirements in the future will haveto go through formal change approval process.

5.4 SPIRAL MODEL:

It was defined by Barry Boehm in his 1988 article, “A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models.
As originally envisioned, the iterations were typically 6 months to 2 years long. Each phase starts with a design goal and ends with a client reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye towardthe end goal of the project.
The following diagram shows how a spiral model acts like:

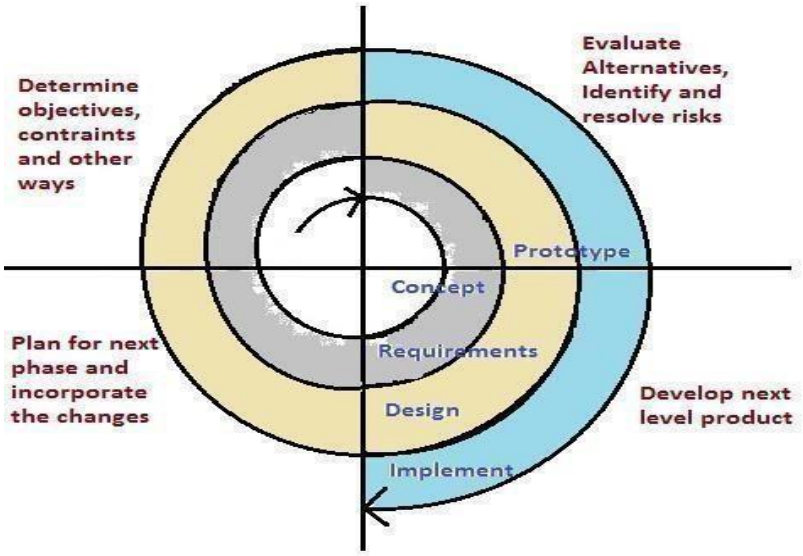

fig 11: spiral model

The steps for Spiral Model can be generalized as follows:

- The new system requirements are defined in as much details as possible.
- This usually involves interviewing a number of users representing all the external orinternal users and other aspects of the existing system.
- A preliminary design is created for the new system.
- A first prototype of the new system is constructed from the preliminary design. This isusually a scaled-down system, and represents an approximation of the characteristics ofthe final product.
- A second prototype is evolved by a fourfold procedure:
- Evaluating the first prototype in terms of its strengths, weakness, and risks.
- Defining the requirements of the second prototype.
- Planning a designing the second prototype.
- Constructing and testing the second prototype.
- At the customer option, the entire project can be aborted if the risk is deemed too great.Risk factors might involve development cost overruns, operating-cost miscalculation, or any other factor that could, in the customer's judgment, result in a lessthan satisfactory final product.
- The existing prototype is evaluated in the same manner as was the previous prototype, and if necessary, another prototype is developed from it according to the fourfold procedure outlined above.

CHAPTER 6
IMPLEMENTATION

Currently, for face detection, perhaps deep learning models perform the best. But face detection was there before the emergence of deep learning as well. Earlier, classical featuredescriptors and linear classifiers were a really good solution for face detection. And the Dlib library provides one such classical solution for face detection. That is, HOG and Linear SVM. This is based on the **HOG** (Histogram of Oriented Gradients) feature descriptor with a **linear SVM** machine learning algorithm to perform face detection.

HOG is a simple and powerful feature descriptor. It is not only used for face detection but also it is widely used for object detection like cars, pets, and fruits. HOG is robust for object detection because object shape is characterized using the local intensity gradient distribution and edge direction.

Step 1: The basic idea of HOG is dividing the image into small connected cells.

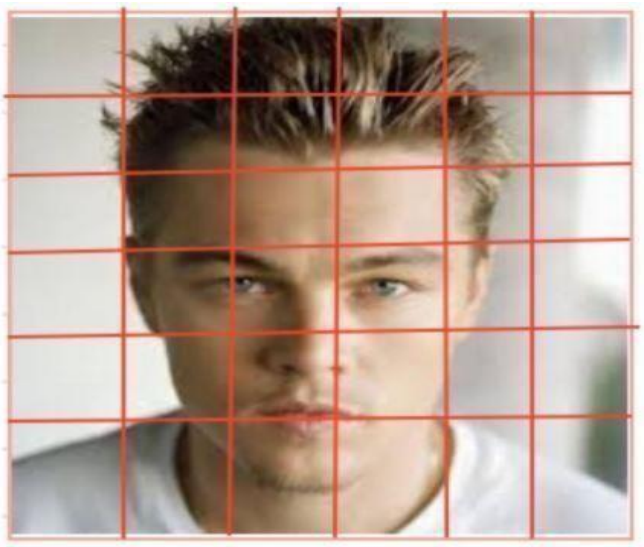

fig 12: image into small connected cells

Step 2: Computes histogram for each cell**.**

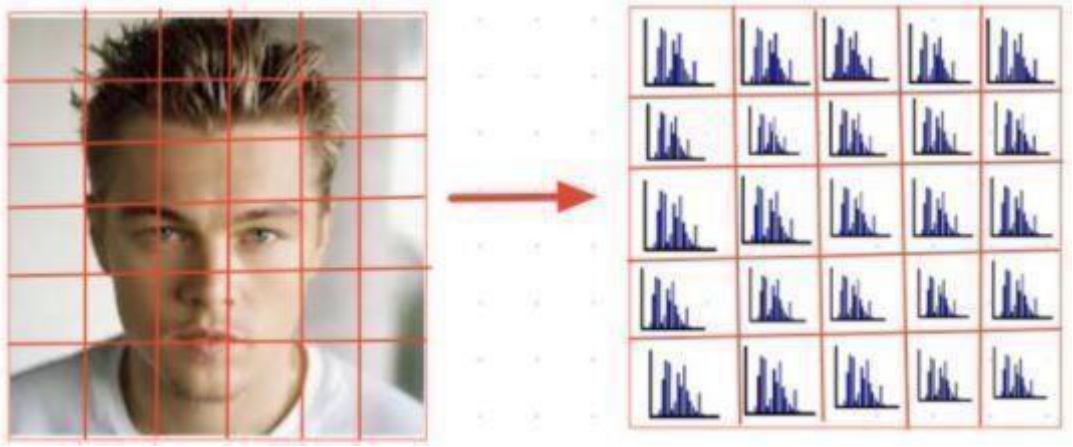

fig 13: histogram for each cell

Step 3: Bring all histograms together to form feature vector i.e., it forms one histogramfrom all small histograms which is unique for each face.

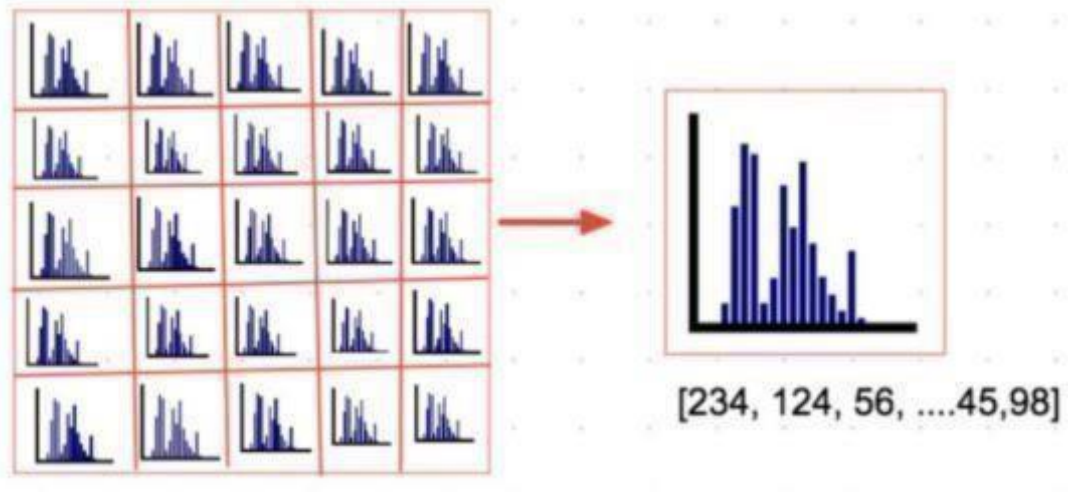

fig 14: forming unique histogram

6.1 SAMPLE CODE:

```
from imutils import face_utils
from utils import *
import numpy as np
import pyautogui as pyag
import imutils
import dlib
import cv2
pyag.FAILSAFE=False
# Thresholds and consecutive frame length for triggering the mouse action.
MOUTH_AR_THRESH = 0.6
MOUTH_AR_CONSECUTIVE_FRAMES = 15
EYE_AR_THRESH = 0.19
EYE_AR_CONSECUTIVE_FRAMES = 15
WINK_AR_DIFF_THRESH = 0.04
WINK_AR_CLOSE_THRESH = 0.19
WINK_CONSECUTIVE_FRAMES = 10

# 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 = "C:/Users/Windows/Desktop/Mini
Project/Mouse_Cursor_Control_Handsfree-master (1)/Mouse_Cursor_Control_Handsfree-
master/model/shape_predictor_68_face_landmarks.dat/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]

# Because I flipped the frame, left is right, right is left.
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, [mouthHull], -1, YELLOW_COLOR, 1)
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:
                pyag.click(button='left')

                WINK_COUNTER = 0

    elif leftEAR > rightEAR:
        if rightEAR < EYE_AR_THRESH:
```

```
            WINK_COUNTER += 1

            if WINK_COUNTER > WINK_CONSECUTIVE_FRAMES:
                pyag.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, "READING INPUT!", (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
        if dir == 'right':
            pyag.moveRel(drag, 0)
        elif dir == 'left':
            pyag.moveRel(-drag, 0)
        elif dir == 'up':
            if SCROLL_MODE:
                pyag.scroll(40)
            else:
                pyag.moveRel(0, -drag)
        elif dir == 'down':
            if SCROLL_MODE:
                pyag.scroll(-40)
            else:
                pyag.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, "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, 80),
    #            cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)

    # Show the frame
    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF

    # If the `Esc` key was pressed, break from the loop
    if key == 27:
        break

# Do a bit of cleanup
cv2.destroyAllWindows()
vid.release()
```

CHAPTER 7

TESTING

7.1 INTRODUCTION:

Field testing will be performed manually and functional tests will be written in detail.

TEST OBJECTIVES:

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.
- To ensure that during operation the system will perform as per specification.
- To make sure that system meets the user requirements during operation
- To make sure that during the operation, incorrect input, processing and output will bedetected
- To see that when correct inputs are fed to the system the outputs are correct
- To verify that the controls incorporated in the same system as intended
- Testing is a process of executing a program with the intent of finding an error
- A good test case is one that has a high probability of finding an as yet undiscoverederror

FEATURES TO BE TESTED:

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct pag

The actual purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner.

TYPES OF TESTING:

- Unit testing.
- Black box testing.
- White box testing.
- Integration testing.
- System testing.

There are many types of testing methods are available in that mainly used testing methodsare as follows:

UNIT TESTING:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration.

INTEGRATION TESTING:

Integration tests are designed to test integrated software components to determine if theyactually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

FUNCTIONAL TESTING:

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions,or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identifiedand the effective value of current tests is determined.

SYSTEM TEST:

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

WHITE BOX TESTING:

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Tested	Test name	Inputs	Expected output	Actual Output	status
1	Load Dataset	Facial Land Mark Dataset	Read dataset	Load dataset	success
2	Capture Frame from Camera	Video Stream	Need to read frame from Camera	Reading Frame is done	success
3	Detect Face	Gray Scale Image	Need to Extract Face Region from Given Gray Scale Image	Face Detected Successfully	success
4	Detect Facial Features	Face Bounding Box Region	Need to Extract Eye Mouse Nose Contours	Contours Generated	success
5	Calculate Aspect Ratio	Starting and ending point	Need to calculate the aspect ratio of given contour	Aspect ratio calculated	success

Table 1: Test cases For cursor movements

CHAPTER 8
OUTPUT SCREENS

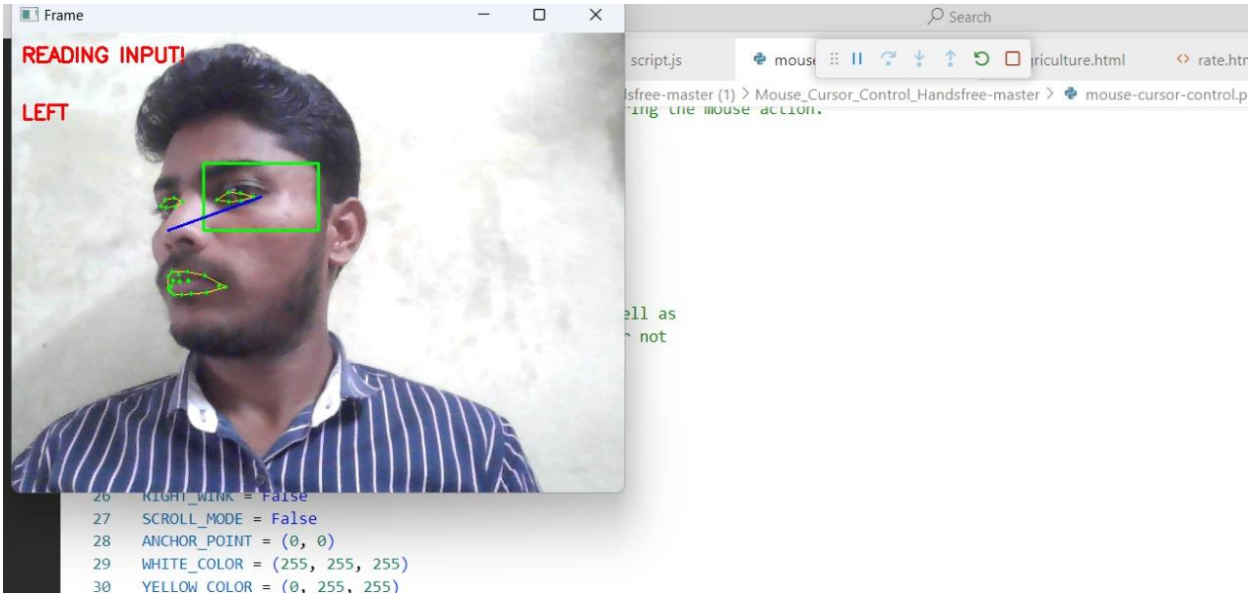

fig 15: cursor movement for left side

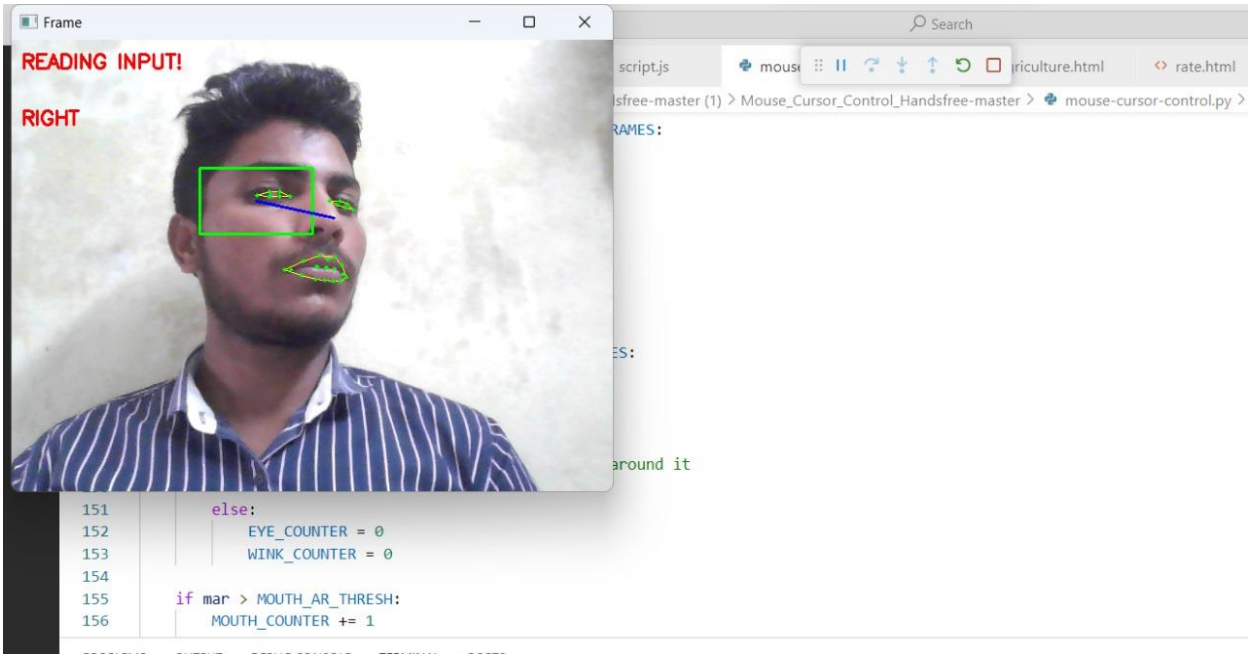

fig 16: cursor movement for right side

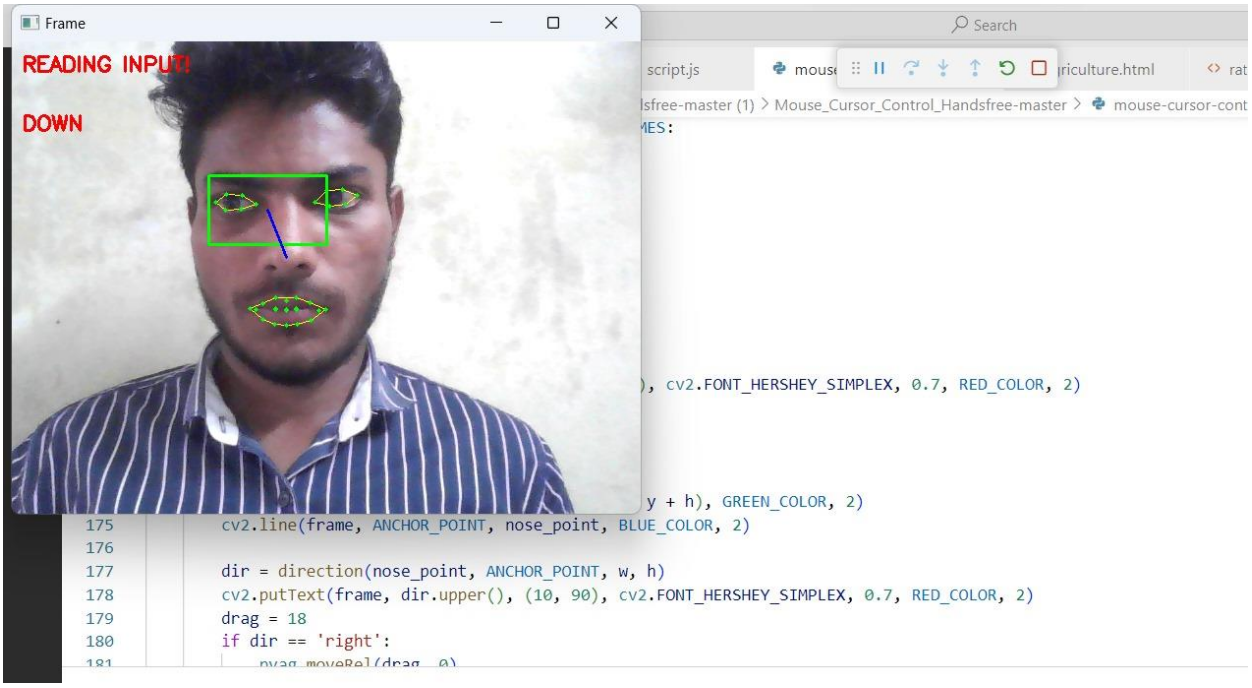

fig 17: cursor movement for down side

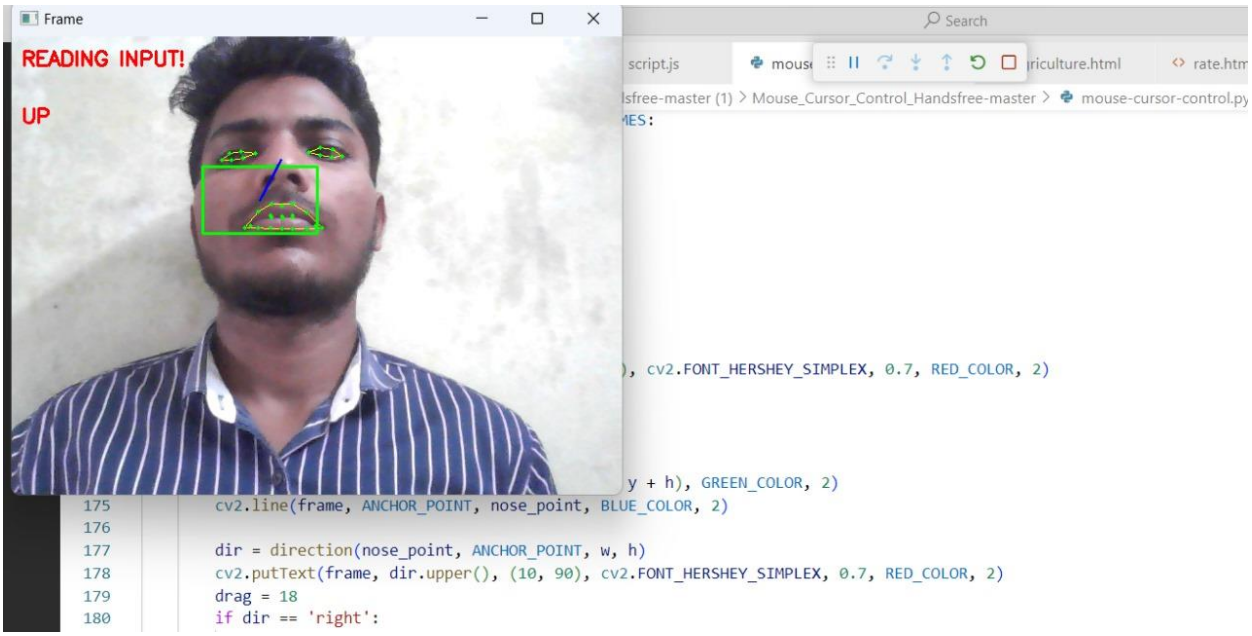

fig 18: cursor movement for up side

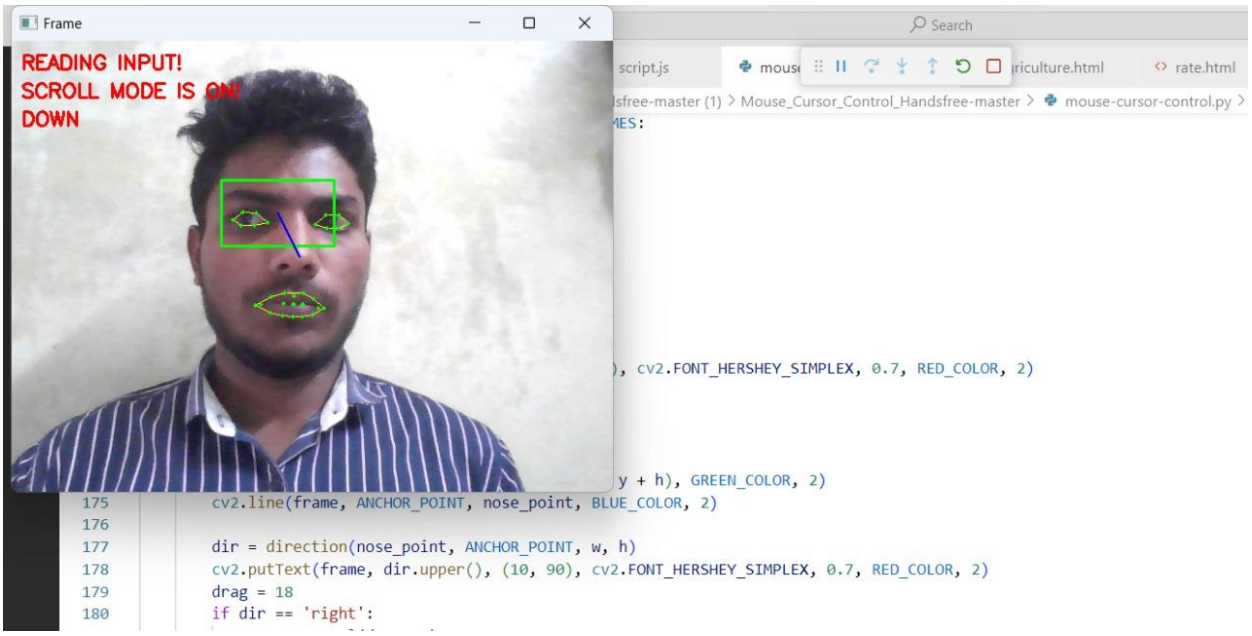

fig 19: cursor movement for scrolling down

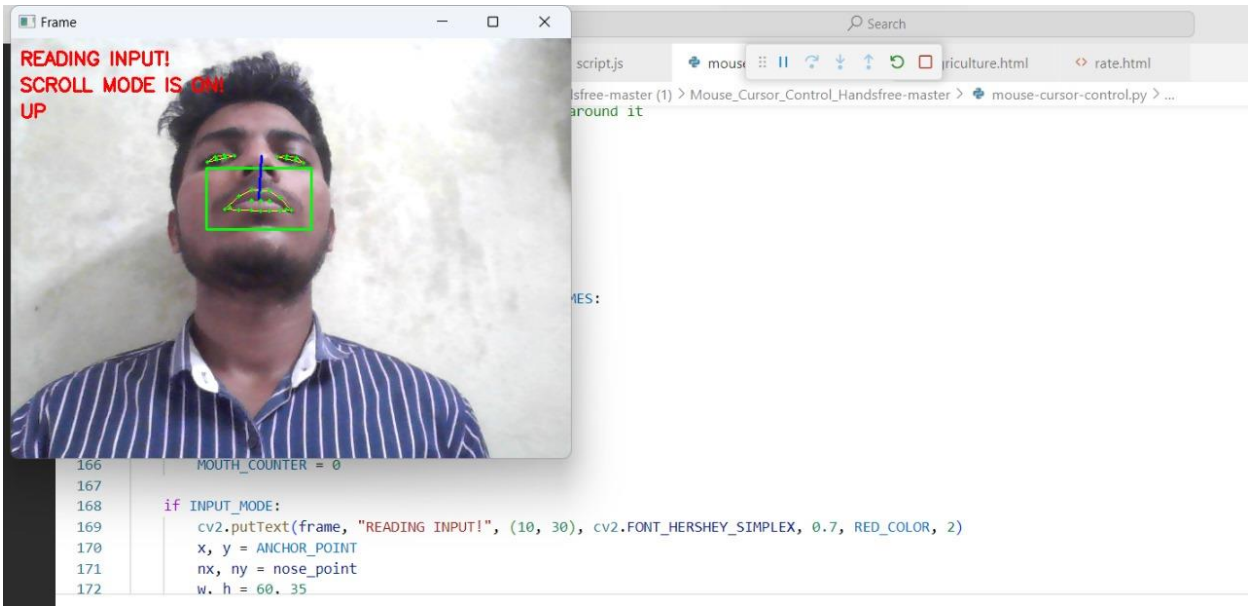

fig 20: cursor movement for scrolling up

CHAPTER 9
CONCLUSION AND FUTURE SCOPE

9.1 CONCLUSION:

This work can be extended to improve the speed of the system by using better trained models. Also, the system can be made more dynamic by making the change in the position of the cursor, proportional to the amount of rotation of the user's head, i.e., the user can decide, at what rate he wants the position of the cursor to change.

Also, future research work can be done on making the ratio more accurate, since the range of the values are the result of the aspect ratios, which is usually small. Hence, to make the algorithm detect the actions more accurately, there can be some modification in the formulae for the aspect ratiosused. Also, to make the process of detection of the face more easy, some image processing techniques can be used before the model detects the face and features of the face.

In order to make user interact with computer naturally and conveniently by only using their eye, we provide an eye tracking based control system. The system combines both themouse functions and keyboard functions, so that users can use our system to achieve almost all of the inputs to the computer without traditional input equipment. The system not only enables the disabled users to operate the computer the same as the normal users do but also provides normal users with a novel choice to operate computer.

9.2 FUTURE SCOPE:

In future, many people who are unable to operate a standard computer mouse or keyboard because of disabilities of their hands or arms, can get possible alternative in multi modal system, which allows controlling a computer without using standard mouse and keyboard.Using head movements to control the cursor across the computer screen and by using the speech for giving the control commands. Automatic speech recognition and head tracking in joint multi modal action are combined to operate the system.

Eye movements provide objective data on how subjects perceive the world and how they react when subjected to different kinds of stimuli, which can be put to use for Researches in Psychology. Eye tracking devices combined with physiological data such asbrain imaging can help identify how the information is processed in the brain.

Eye trackingcan be used to analyze visual development and link it to developmental aspects of neurological functions, neurological diseases and brain damage. Also, Reading patterns can be cross referenced with different demographics of people and therefore provide insight into how they gather information.

Human computer interaction allows users to input information in a more natural way into their computers. Eye tracking can be used as a control medium, like moving the cursorand clicking on icons on the screen, as well as creating adaptive user interfaces, where the computer reacts to the eye gaze of the user and create an interactive environment.

Eye tracking is great for coaches who want to train their players on effectively gathering information from the field through simulations. Bats men's eye movements monitor the moment when the ball is released, make a predictive cascade to the place where they expectit to hit the ground, wait for it to bounce, and follow its trajectory for 100−200ms after thebounce.

Learning to analyze an environment quickly can be a valuable skill in air traffic control, radar control, medical X-ray examinations, video surveillance, industrial process control, driving, army or police field work, surgical training and others. Teaching this skill through simulations using an eye tracker can eliminate the time consuming process of onlylearning it with experience.

CHAPTER 10
REFERENCES

10.1 WEBSITES:

ACM Digital Library:
Website: https://dl.acm.org/
ACM (Association for Computing Machinery) is a leading publisher of research papers and articles related to HCI and assistive technology. You can search for specific papers and publications on eye-controlled mouse systems.

IEEE Xplore:
Website: https://ieeexplore.ieee.org/
IEEE Xplore is another valuable resource for research papers and articles in the field of computer science, electrical engineering, and HCI.

Google Scholar:
Website: https://scholar.google.com/
Google Scholar is a free search engine that indexes scholarly articles, including those related to HCI and eye-controlled mouse technology.

10.2 BOOKS:

1. "The Handbook of Multimodal-Multisensor Interfaces" by Sharon Oviatt:
- This book covers various aspects of multimodal interfaces, including eye tracking, which is relevant to HCI projects involving eye-controlled systems.

2. "Eye Tracking Methodology: Theory and Practice" by Andrew T. Duchowski:
- This book provides a comprehensive overview of eye tracking methodology, which is fundamental to understanding and developing eye-controlled systems.

3. "The Design of Everyday Things" by Don Norman:

- While not specific to eye-controlled mouse projects, this classic book on design and usability provides principles that are crucial for creating user-friendly interfaces.

4. "Interaction Design: Beyond Human-Computer Interaction" by Helen Sharp, Yvonne Rogers, and Jenny Preece:
- This book covers principles of interaction design, which are important for creating effective HCI systems, including those involving eye-tracking technology.

5. "Assistive Technologies: Principles and Practice" by Albert M. Cook and Janice Miller Polgar:
- This book covers a wide range of assistive technologies, which can include eye-controlled systems. It provides insights into the principles and practices of assistive technology.

6. "Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Guidelines" by Jeff Johnson:
- This book explores user interface design principles, which are critical for creating intuitive and effective interfaces for eye-controlled mouse projects.

10.3 RESEARCH PAPERS:

[1] Alex Poole and Linden J. Ball, “Eye Tracking in Human Computer Interaction and Usability Research: Current Status and FutureProspects,” in Encyclopaedia of Human Computer Interaction (30 December 2005) Key: citeulike:3431568,
2006, pp. 211-219.

[2] D. H. Yoo, J. H. Kim, B. R. Lee, and M. J. Chung, “Non contact Eye Gaze Tracking System by Mapping of Corneal Reflections,” in Fifth IEEE International Conference on Automatic Face and Gesture Recognition (FGR02), 2002, pp. 94-99.

[3] Rafael Barea, Luciano Boquete, Manuel Mazo, and Elena Lpez, “System for assisted mobility using eye movements based on electrooculography,” IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING, vol. 10, no. 4, pp. 209-217, DECEMBER 2002.

[4] H. Singh and J. Singh, “A Review on Electrooculography,” International Journal of Advanced Engineering Technology, vol. III, no. IV, 2012.

[5] K. Irie, B. A. Wilson, and R. D. Jones, “A laser-based eye tracking system,” Behavior Research Methods, Instruments, & Computers, vol. 34, no. 4, pp. 561-572, 2002 .

[6] P Ballard and George C. Stockman, “Computer operation via face orientation,” in Pattern Recognition, 1992. Vol.I. Conference A: Computer Vision and Applications, Proceedings., 11th IAPR International Conference on, 1992, pp. 407-410.

[7] T. Horprasert, Y. Yacoob, and L.S. Davis, “Computing 3-D head orientation from a monocular image sequence,” in Second International Conference on Automatic Face and Gesture Recognition, 1996, pp. 242- 247.

[8] K. Arai and M. Yamaura, “Computer Input with Human Eyes Only Using Two Purkinje Images Which Works in a Real Time Basis without Calibration,” CSC Journals, vol. 1, no. 3, pp. 71-82, 2010.

[9] D. Back, “Neural Network Gaze Tracking using Web Camera.,” Linkping University, MS Thesis 2005.

[10] R. Gonzalez and R. Woods, Digital Image Processing, 3rd ed.: Pearson Education, 2009.