A Mini Project report on

AEROGESTURE: EYE GESTURE CONTROLLED VIRTUAL MOUSE

A documentation submitted in partial fulfillment of the academic requirement for the award of degree of

BACHELOR OF ENGINEERING

in

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

by

SYED ABDUL KAREEM AHMED

(1604-21-748-048)

MANZOOR MOHD

(1604-21-748-025)

AHMED ABDUL FAHEEM

(1604-21-748-015)

Under the guidance of

Dr. UMA N. DULHARE

Professor and Head, CS&AI Dept., MJCET

COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE DEPARTMENT

MUFFAKHAM JAH COLLEGE OF ENGINEERING AND TECHNOLOGY

(Affiliated to Osmania University) Hyderabad.

2023 – 2024

MUFFAKHAM JAH

COLLEGE OF ENGINEERING AND TECHNOLOGY

Mount Pleasant, 8-2-249 to 267, Road No. 3, Banjara Hills,Hyderabad – 500 034, Telangana State, India

CERTIFICATE

This is to certify that the mini project report on **"AeroGesture: Eye Gesture Controlled Virtual Mouse"** is a bonafide work carried out by **Syed Abdul Kareem Ahmed (1604-21-748-048), Manzoor Mohd (1604-21-748-025)** and **Ahmed Abdul Faheem (1604-21-748-015)** in the partial fulfillment of therequirements for the award of the B.E. CSE(AI&ML) in MUFFAKHAM JAH COLLEGE OF ENGINEERING AND TECHNOLOGY, Hyderabad for the academic year 2023-2024.

Project Guide

Dr. Uma N. Dulhare
Professor & Head,
CS &AI Dept, M.J.C.E.T.
Hyderabad.

Head of the Department

Dr. Uma N. Dulhare
Professor & Head,
CS&AI Dept, M.J.C.E.T.
Hyderabad.

External Examiner

DECLARATION

We hereby declare that the work entitled "**AeroGesture: Eye Gesture Controlled Virtual Mouse"** developed under the supervision of **Dr. UMA N. DULHARE, Professor & Head, CS&AI Department** and submitted to **MUFFAKHAM JAH COLLEGE OF ENGINEERING AND TECHNOLOGY** in original and has not been submitted in part or while for undergraduation degree to any other university.

SYED ABDUL KAREEM AHMED
(1604-21-748-048)

MANZOOR MOHD
(1604-21-748-025)

AHMED ABDUL FAHEEM
(1604-21-48-015)

ACKNOWLWDGEMENT

Our sincere thanks and gratitude to **Dr. UMA N. DULHARE, Professor & Head of the Department** for all the timely support and valuable suggestions during the period of my mini project.

We are extremely thankful to the Project Review Committee Members, Dept. of CS&AI for their encouragement and support throughout the project.

We are extremely thankful and indebted to my internal guide**, Dr. Uma N. Dulhare, Professor & Head, Department of CS&AI** for her exemplary guidance and encouragement. Her trust and support inspired me in the most important moments of making the right decisions and I am glad to work under her supervision.

We would like to thank all the faculty and staff of the department who helped us directly or indirectly in completing the project work. Last but not least, I would like to thank my family members and friends for their co-operation in completing the project.

SYED ABDUL KAREEM AHMED
(1604-21-748-048)

MANZOOR MOHD
(1604-21-748-025)

AHMED ABDUL FAHEEM
(1604-21-748-015)

TABLE OF CONTENTS

ABSTRACT

AeroGesture is the application, where one can handle the mouse action by just using their two eyes and face movement, this application will figure out where the face is moving by predicting face movement in real-time by getting video frame feed from your camera no infrared camera is needed for these actions it will detect your eyelid and when it closes the blinking event triggers.

The Aim of making this software is to simplify the use of the hand in just one place i.e. keyboard, keyboard-centric people don't like to flip their hands to use a mouse when handling the keyboard. It can also be used by people with severe disabilities who can use their eyes to communicate with computers.

One can see the results of what eye-tracking technology can work when we use it with an interface of a handy computer.

Keywords: Eye-Tracking, Mouse Control, Accessibility, Assistive Technology, User Interface (UI), Eye Movement, Gaze Tracking

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

Who says that we need one mouse/eye-tracking device when you have two of your eyes, that'sthe idea behind the eyemousecontroller and when it works it works amazingly, just look/gazeat the screen and move your face to drag the mouse cursor.

You don't have to worry about natural blinks, it will only consider the intentional blinks, mouse movement will gradually increase if the distance of the face middle point increased from the camera midpoint.

1.1 Purpose

This is an application made for keyboard-centric users and lazy users who do not want to use lift their hands for mouse movements or clicks. This application can also be used for the person with physical disabilities who wants to use a computer to communicate with others orjust to use computers.

So, this application is a medium to manage mouse events easily from the use of eyes and face movement. One can alter settings as per their needs, to save time to input camera number andstart camera one of the very useful features is added which starts the working of a program onthe start-up of application, the user will not have to start an application and perform setting toturn eye-tracking on.

1.2 Scope

This is one of the kind applications where users easily move their face and cursor teleport to the place user desires. Since it is an application that can be used in daily normal life, there is just no end to it! It is in the hands of the users to keep using and it is an easy alternative for an eye-tracking device like Tobii's eye tracker.[1]

This application is to be and will be used for the benefit of, especially disabled users. Users can change the aspect ratio based on their camera ratio for accurate tracking, users can also change the illumination in the video feed if the camera is not in good lighting conditions, andillumination settings come with a slider that helps to choose the correct illumination for the real-time lighting conditions. Users can also invert the camera this feature is added for the cameras which flip the camera feed automatically.

1.3 Objective

The main objective of this application is to make it easy to manage the mouse without any interference. Eye-tracking is aim to keep track of the human eye and perform basic mouse functionalities. This is achieved by employing computer vision and image processing algorithms. Conventionally, we use a mouse to make the interaction between the system and the user as an input device. This application gives us a proposal to control the cursor of the computer with the help of the user's eye. Making software for mouse movements will eliminate users wearing an incontinent device to take specific actions.

An eye-tracking system is one of the best applications for a handicapped person. To develop this application various algorithm methods and techniques of image processing are used. Thesemethods and techniques of image processing give a well-designed model for an eye-tracking system.

1.4 Source Of Idea

The idea of creating this application came suddenly while writing an essay where the use of akeyboard is more than a mouse, and using a mouse when writing was a drag. At the same time,I was working with some image processing programs in **OpenCV**[2], then I searched across theinternet to find software that can perform mouse events with eyes and found out it is possiblewith external eye-tracking hardware's, then idea about making a piece of software struck to head which can make it easy to control mouse events. And after some thought, I came to thinkthat it can help some physically handicapped people too who may be wanted to use computersas a normal human.

CHAPTER 2

LITERATURE SURVEY

S. No.	Title	Description	Results	Advantages	Drawbacks	Year of Publications
1.	**Eye-controlled Mouse Cursor for Physically Disabled Individual**	This research proposes a hybrid system using facial movements and eye tracking for hands-free mouse control.	Achieved 87% overall accuracy and 94% left/right click classification accuracy.	Enables hands-free computer interaction for individuals with physical limitations.	Requires good lighting conditions and may not be suitable for users with specific facial movements.	2018
2.	**Human-Eye Controlled Virtual Mouse**	This project demonstrates an eye-based interface that translates eye movements like blinking, staring, and squinting into mouse	Successfully implemented basic mouse functionalities using eye movements.	Offers a low-cost and accessible solution for individuals with upper limb disabilities.	Requires calibration and might have limited functionalities compared to a traditional mouse.	2018

		actions.				
3.	**Eye Tracking Based Mouse Control Using OpenCV**	This study explores utilizing OpenCV library for eye tracking to control the mouse cursor.	Developed a basic prototype demonstrating eye-based cursor movement.	Provides open-source tools and resources for further development of eye-controlled interfaces.	Requires advanced programming skills and hardware setup for accurate eye tracking. Can't use regular camera	2022
4.	**Eye-controlled Assistive Technology: A Comprehensive Review**	This review paper provides a broader perspective on various eye-controlled assistive technologies beyond just mouse control.	Highlights the diverse applications and ongoing advancements in eye-tracking technology.	Offers valuable insights into the potential of eye-controlled interfaces for various assistive needs.	Does not focus on specific details of building an eye-controlled mouse project.	2021
5.	**ArtaEye: A Web-based Eye-tracking Platform for Artistic Creation**	This project explores the use of eye tracking for	Demonstrates the potential of eye-controlled interfaces	Expands the scope of eye-tracking technology beyond	Requires specific hardware and software setups, and may not be	2021

		creating artwork and digital paintings.	for creative applications.	just computer interactions.	readily accessible to everyone.	
6.	**Calibration-Free Eye Tracking for Human-Computer Interaction**	This research investigates methods for eye tracking without requiring user calibration.	Achieves promising results for reducing calibration efforts and improving user experience	Makes eye-controlled interfaces more user-friendly and adaptable.	May require advanced algorithms and computational resources, potentially increasing implementation complexity.	2023
7.	**Deep Learning-based Eye Tracking for Enhanced Accuracy**	This study explores utilizing deep learning techniques for improving the accuracy and robustness of eye tracking systems.	Achieves state-of-the-art performance in eye tracking compared to traditional methods.	Contributes to the development of more accurate and reliable eye-controlled interfaces.	Requires computational power and may have higher resource requirements compared to simpler methods.	2023

CHAPTER 3
SURVEY OF TECHNOLOGIES

3.1 Why OpenCV?

OpenCV is the huge open-source library for computer vision, machine learning, and image processing and now it plays a key role in real-time operation which is important in today's systems. By using it, one can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries, such as NumPy,python can process the OpenCV array structure for analysis. To identify image patterns and their 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 is 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.

3.1.1 Advantages Of OpenCV

3.1.1.1 Its Fast and Simple

In image processing, since you are dealing with a large number of operations per second, it is mandatory that your code not only providing the correct solution but that it is also providing it in the fastest manner. The OpenCV function is nearly 25x faster than the NumPy function.[3]

Convolution is the basis of many computer vision algorithms and straightforward algorithms to implement in C, but in the comparison of various implementations OpenCV comes out as the winner.

If you are a python programmer, using OpenCV (Python) would be very easy. Python is an easy language to learn (especially compared to C++). It is also an excellent first language to learn.

3.1.1.2 Its Well Established

One of the great benefits of an open-source library is your ability to modify them to suit your needs. If you want to modify OpenCV, you have to modify the C/C++ source.

3.1.2 Limitations of OpenCV

3.1.2.1 Less Convenient

Well, MATLAB is more convenient in developing and data presentation, however, OpenCV is much faster in execution. In the case of OpenCV, the speed ratio reaches more than 80 in some cases.[4]

However, OpenCV is comparatively harder to learn due to a lack of documentation and error handling codes. This disadvantage is what makes novice computer vision users lean towards MATLAB more often. But once gained expertise with OpenCV, some professionals suggest sticking with it as it is the most comprehensive open-source library for computer vision and has a large user community.

Some professionals also suggest MATLAB as it is useful for rapid prototyping and its code is quite easy to debug. Moreover, it has good documentation and support.

3.2 Why PyQt5?

There are so many options provided by Python to develop GUI applications and PyQt5 is one of them. PyQt5 is a cross-platform GUI toolkit, a set of python bindings for Qt v5. One can develop an interactive desktop application with so much ease because of the tools and simplicity provided by this library.

A GUI application consists of Front-end and Back-end. PyQt5 has provided a tool called 'QtDesigner' to design the front-end by drag and drop method so that development can become faster and one can give more time on back-end stuff.

3.2.1 Advantages Of Using PyQt

Coding versatility –GUI programming with Qt is built around the idea of signals and slots for creating contact between objects. This allows versatility in dealing with GUI incidents which results in a smoother code base.

More than a framework: Qt uses a broad variety of native platform APIs for networking,database development, and more. It provides primary access to them through a special API.Various UI components: Qt provides multiple widgets, such as buttons or menus, all designed with a basic interface for all compatible platforms.

Various learning resources: As PyQt is one of the most commonly used UI systems for Python, you can conveniently access a broad variety of documentation.

3.2.2 Limitation Of Using PyQt

Lack of Python-specific documentation for classes in PyQt5

It takes a lot of time to grasp al the specifics of PyQt, meaning it's a pretty steep learning curve.

If the application is not open-source, you must pay for a commercial license.

3.2.3 PyQt vs TKINTER

Coding flexibility – GUI programming with Qt is designed around the concept of signals and slots for establishing communication amongst objects. That permits flexibility when dealing with GUI events and results in a smoother codebase.

More than a framework – Qt uses a wide array of native platform APIs for networking, database creation, *and* many more. It offers primary access to them via a unique API. Various UI components – Qt offers several widgets, such as buttons or menus, all designedwith a basic appearance across all supported platforms.

Various learning resources – because PyQt is one of the most used UI frameworks for Python, you can get easy access to a wide array of documentation.

Easy to master – PyQt comes with a user-friendly, straightforward API functionality, along with specific classes linked to Qt C++. This allows the user to use previous knowledge from either Qt or C++, making PyQt easy to understand.

Tkinter does not include advanced widgets. It has no similar tool as Qt Designer for Tkinter. It doesn't have a native look and feel

CHAPTER 4
REQUIREMENT ANALYSIS

4.1 Requirement Specifications

4.1.1 Feasibility Study

Feasibility is the determination of whether or not a project is worth doing. The process followedin making this determination is called a feasibility study.

This type of study determines if a project can and should be taken. Once it has been determined that a project is feasible, the analyst can go ahead and prepare the project specification whichfinalizes project requirements.

4.1.2 Technical Feasibility

Technical issues involved are the necessary technology exists, technical guarantees of accuracy, reliability, ease of access, data security, and aspects of future expansion. Technology exists to develop a system. The proposed system can hold data to be used. The proposed system is capable of providing adequate response regardless of the number of users. The proposed system being modular to the administrator if he/she wants can add more features in the future and as well as be able to expand the system.

4.1.3 Operational Feasibility

If the system meets the requirements of the customers and the administrator, we can say that the system is operationally feasible. The proposed system will be beneficial only if it can be turned into a system that will meet the requirements of the store when it is developed and installed, and there is sufficient support from the users. The proposed system will improve thetotal performance. Customers here are the most important part of the system, and the proposedsystem will provide them with a convenient mode of operation for them.

4.1.4 Economic Feasibility

Economic Feasibility is the most frequently used method for evaluating the effectiveness of the proposed system if the benefit of the proposed system outweighs the cost then the decisionis made to design and implement the system. The cost of hardware and software is affordable.

4.2 Software and Hardware Requirements

4.2.1 Hardware Requirements

4.2.1.1	Processor	: Intel Core 2 Duo or above
4.2.1.2	Ram	: 2GB or above
4.2.1.3	Hard Drive	: 250GB or above
4.2.1.4	System Bus	: 64 bit
4.2.1.5	Clock Speed	: 1.8 GHz
4.2.1.6	Webcam	: Quality Camera for Image Recognition

4.2.2 Software Requirements

4.2.2.1	Operating System	: Windows XP/7/Vista/8/10
4.2.2.2	Programming Language	: Python
4.2.2.3	Major Libraries	: OpenCV, Dlib, Pynput, PyQt5

CHAPTER 5
SYSTEM DESIGN DETAILS

5.1 Software Design

5.1.1 Interface

The interface of the eye mouse controller is to take input and show the webcam's video feed, start tracking, and detecting blinks.

Fig 5.1.1 Camera Number

After starting the eye mouse controller application first user needs to input the camera number,if it a laptop camera number default value is 1 and the user had attached an external camera then the user will input number 2 as a value if multiple cameras are attached and then user need to input correct camera number to run the application. After keying the correct camera numberEnter button needs to be pressed.

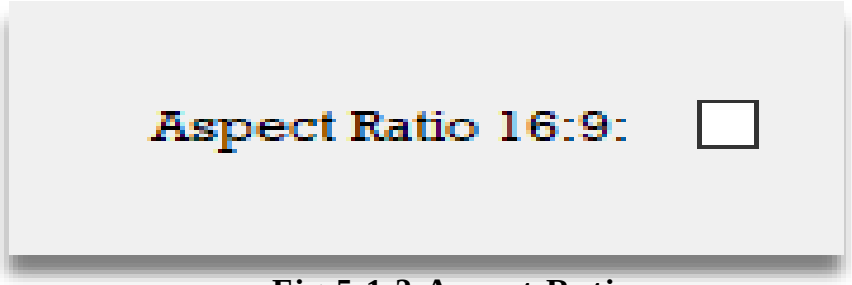

Fig 5.1.2 Aspect Ratio

After adding the camera number user need to specify the aspectratio of the camera which is used as a webcam. Most of the webcam comes with 4:3 as an aspect ratio so our application comes compatible with 4:3. If the camera-input aspect ratio is 16:9 then the user needs to inputselection in the aspect ratio checkbox.

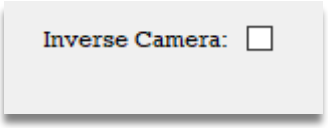

Fig 5.1.3 Inverse Camera

After choosing the correct aspect ratio user check whether a camera is flipped by default. If a webcam is flipped then check theinverse camera button. It will reverse the flipping and tracking will happen in exact order.

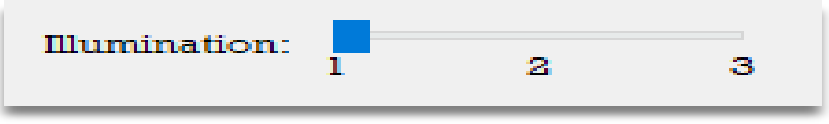

Fig 5.1.4 Illumination

After choosing the correct alternatives user need to balanceout the lighting with help of illumination slider which helps in low light condition. Illumination slider comes with 3 values 1/2/3. The default value is one of the value increases by 1 then the gamma valueincreases in a video frame.

And here is the full view of the interface of the AeroGesture

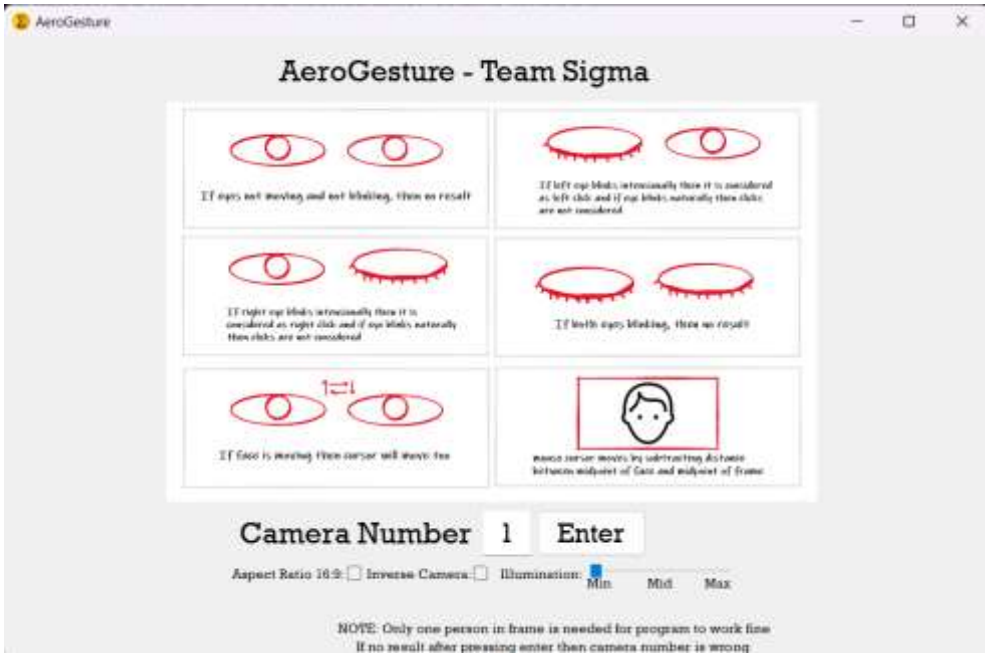

Fig 5.1.4 Interface

In this application for easy use and no repetitive assignment of camera number, the camera number is saved in a file where it stores the last used camera number which is used to run OpenCV frame with previously used frame.

5.1.1 OpenCV Frame

After configuring all the settings, a user selects enter to run a program and gets a new windowstarted of OPENCV video capture. Which illustrates how a face is being recognized and showsthe distance between the face midpoint and the camera midpoint.

It also shows if a user is far from a frame or not, user can different by looking at the video frame how gradually cursor speed is increasing.

Fig 5.1.5 (a) UI

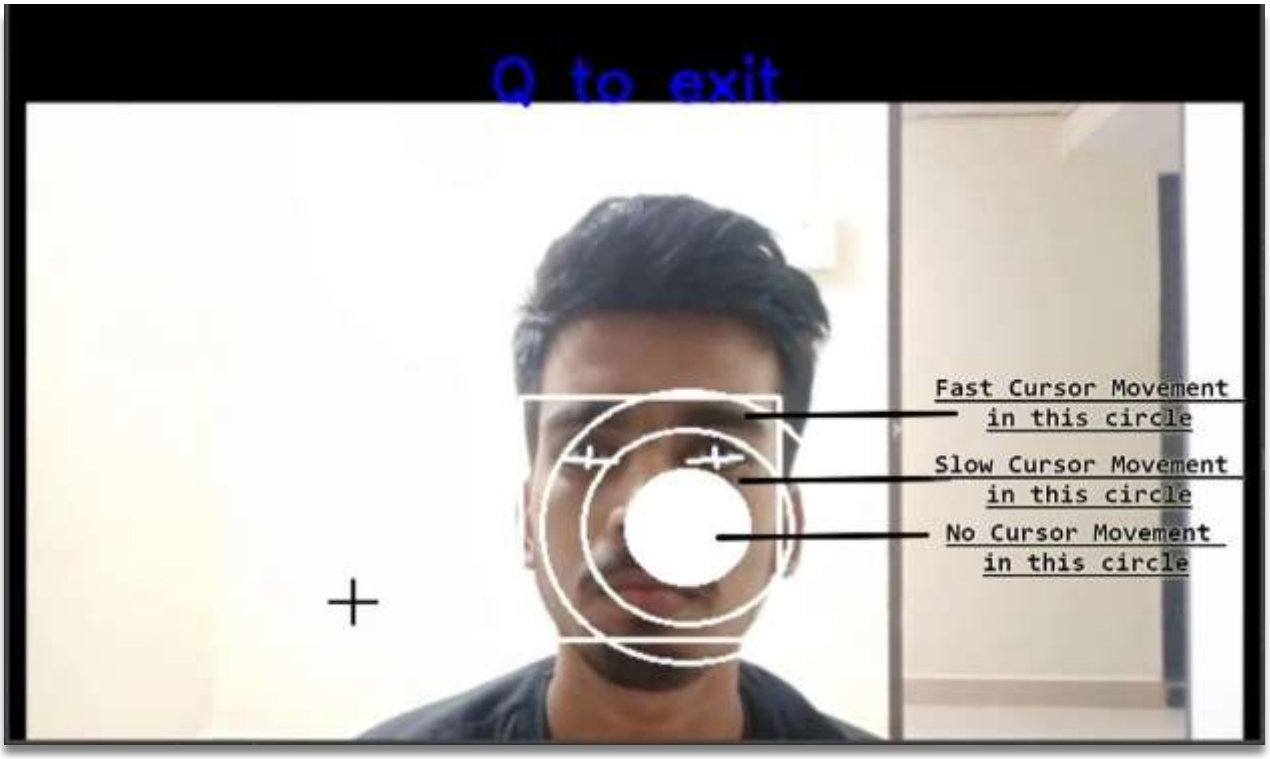

Fig 5.1.5(b) UI

Here if the user's face is in the middle circle then the cursor not moving, and if a face is in the second circle then the cursor moves slowly, and if the face crosses 2nd circle then the cursor moves at a faster speed.

Users can click the letter ‘Q’ to exit. By clicking ’Q’ the user just exits out of the OpenCV video frame.

To exit the full program user need to close the interface too.

5.2 Flow Chart

A **flowchart** is a type of diagram that represents a workflow or process. A flowchart can alsobe defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task.

The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows. This diagrammatic representation illustrates a solution model to a given problem. Flowcharts are used in analyzing, designing, documenting, or managing a process or program in various fields.

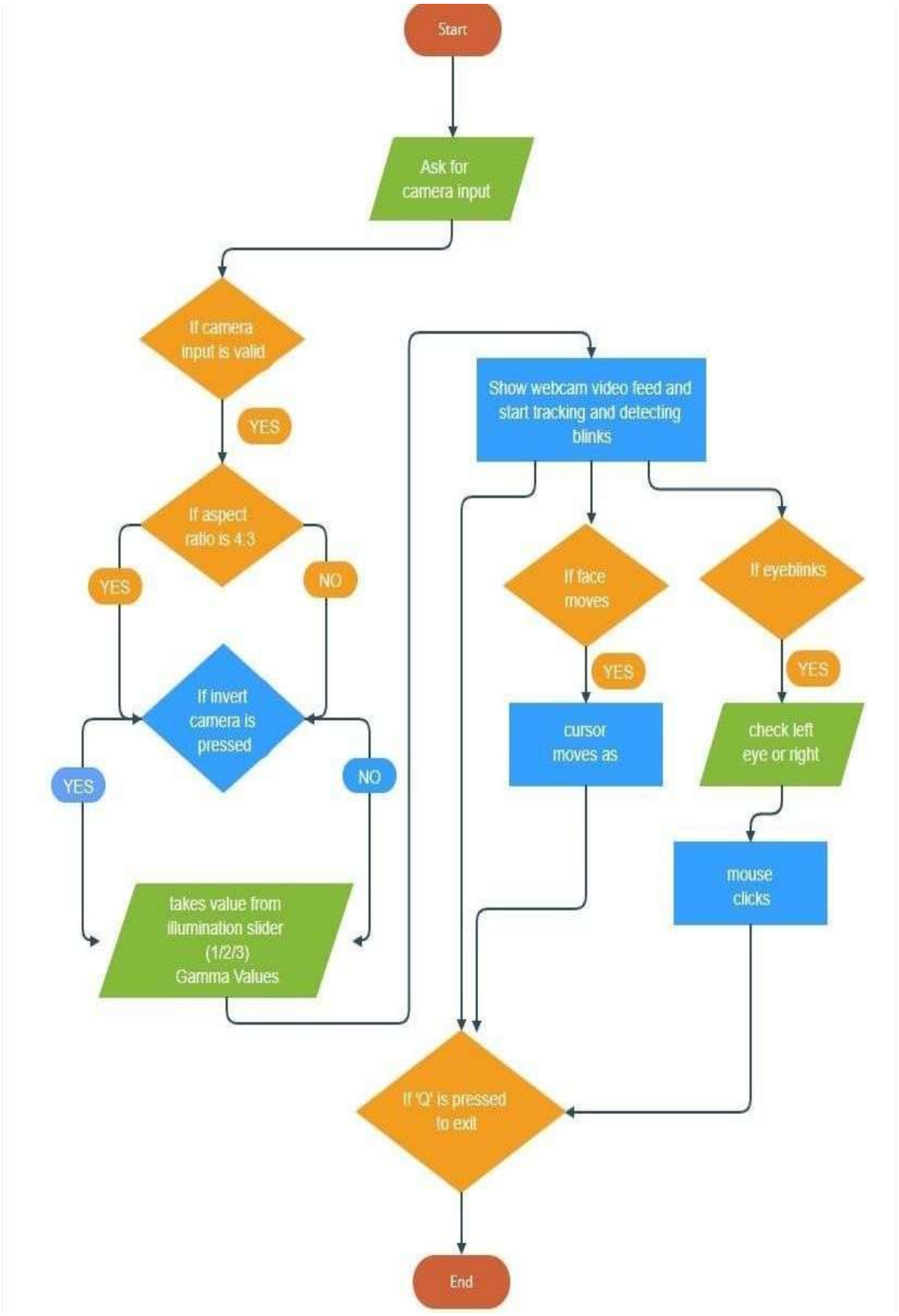

Fig 5.2 Flow Chart

5.3 Gantt Chart

A Gantt chart is a type of bar chart that illustrates a project schedule. This chart lists the tasks to be performed on the vertical axis, and time intervals on the horizontal axis. The width of the horizontal bars in the graph shows the duration of each activity. Gantt charts illustrate thestart and finish dates of the terminal elements and summary elements of a project. As we havealready discussed, Gantt charts are used for project management purposes. To use Gantt chartsin a project, there are a few initial requirements fulfilled by the project.

First of all, the project should have a sufficiently detailed Work Breakdown Structure (WBS). Secondly, the project should have identified its milestones and deliveries.

In some instances, project managers try to define the work breakdown structure while creating a Gantt chart. This is one of the frequently practiced errors in using Gantt charts. Gantt chartsare not designed to assist the WBS process; rather Gantt charts are for task progress tracking.

This method maximizes the float time available for all tasks. Gantt chart of this project is as follows:

TASKS	START DATE	DAYS TOOK
PROJECT PLANNING AND RESEARCH	6 Nov, 2023	10
SURVEY OF TECHNOLOGY	16 Nov, 2023	5
REQUIREMENT ANALYSIS	21 Nov, 2023	9
VIDEO OUTPUT ANALYSIS	30 Nov, 2023	1
CURSOR MOVEMENT	1 Dec, 2023	5
EYE BLINKING DETECTION OPTIMIZATION OF VIDEO FEED	6 Dec, 2023	5
RESCALING FRAME	11 Dec, 2023	2
IMPROVED CURSOR MOVEMENT	13 Dec, 2023	7
IMPROVED EYE BLINK	20 Dec, 2024	20
INTERFACE DESIGN	9 Jan, 2024	2
ASPECT RATIO WORK	11 Jan, 2024	2
INVERT CAMERA FEATURE	13 Jan, 2024	4
ADDED ILLUMINATION SLIDER	17 Jan, 2024	5
WORKED ON BUGS	23 Jan, 2024	5
COMPATIBILTY FOR ALL DEVICE SIZES	28 Jan, 2024	5
INTERFACE IMPROVEMENT	2 Feb, 2024	2
TESTING	4 Feb, 2024	2
DELIVERY	6 Feb, 2024	1

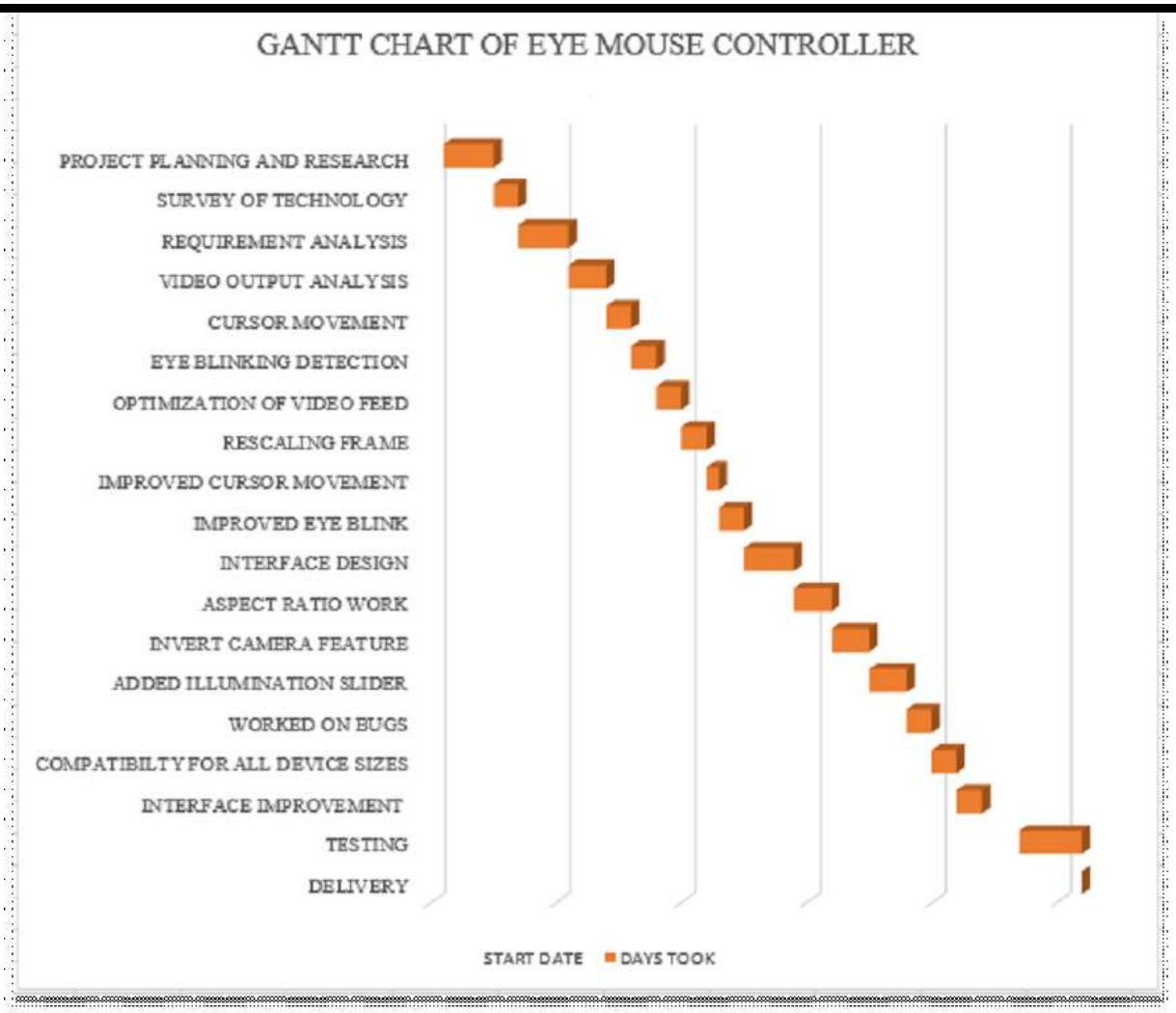

Fig 5.3 Ghantt Chart

5.4 ER (Entity-Relation Diagram)

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database.An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties.

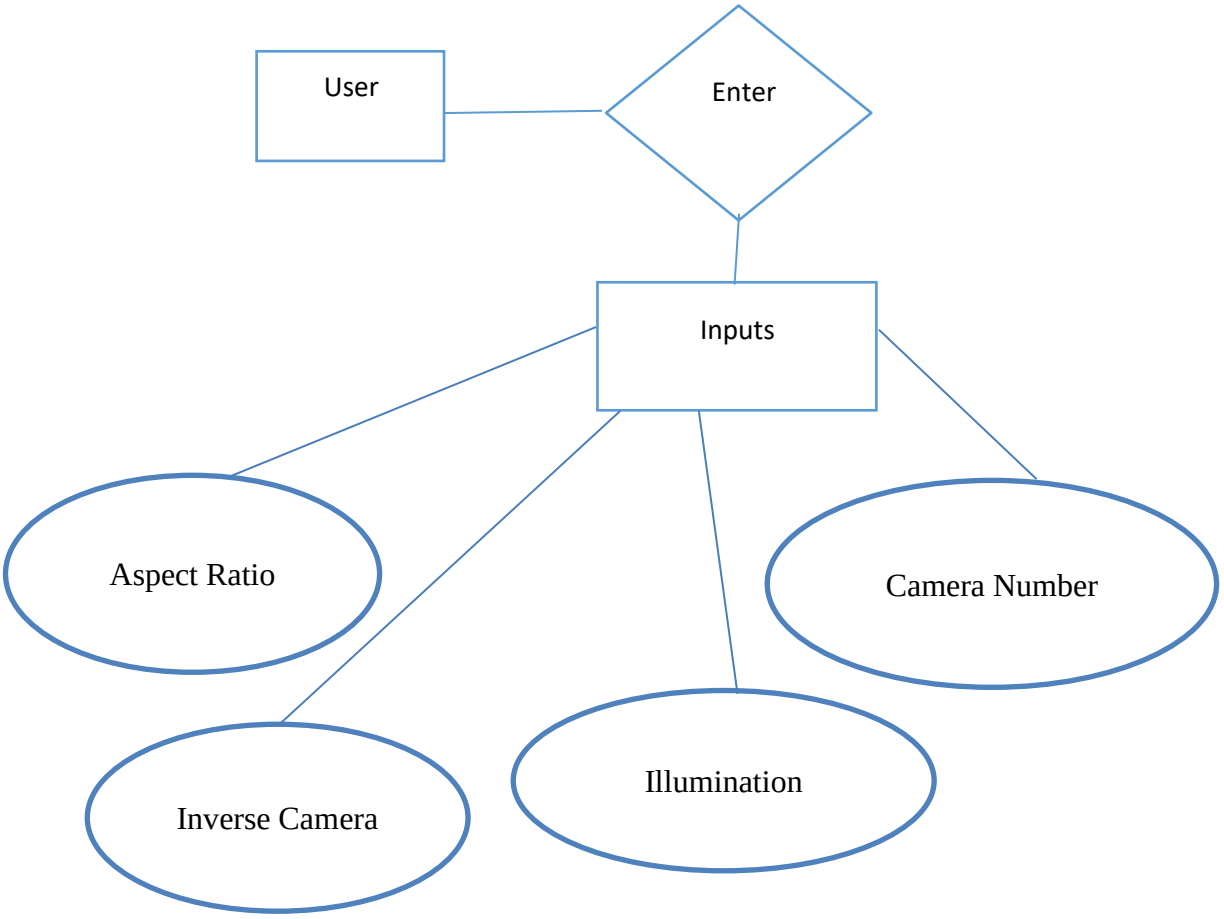

Fig 5.4 ER-Diagram

CHAPTER 6
SYSTEM IMPLEMENTATION

6.1 Code Snippets

Interface-imports: -

```
from PyQt5 import QtCore, QtGui, QtWidgets
import eyetracking_fastpc
from PyQt5 import sip
import cv2 as cv
cv2=cv
import dlib
import mousecontrol_eye
from win32.win32api import GetSystemMetrics
from pynput.mouse import Listener,Button,Controller
import sys
from PyQt5 import QtCore, QtGui, QtWidgets
from PyQt5.QtCore import QCoreApplication
import time
import webbrowser
from imutils.video import WebcamVideoStream
```

EyeMouse-Interface: -

```
class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.setEnabled(True)
        MainWindow.resize(1092, 684)
        sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Fixed, QtWidgets.QSizePolicy.Fixed)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)
        sizePolicy.setHeightForWidth(MainWindow.sizePolicy().hasHeightForWidth())
        MainWindow.setSizePolicy(sizePolicy)
        MainWindow.setContextMenuPolicy(QtCore.Qt.NoContextMenu)
        icon = QtGui.QIcon()
        icon.addPixmap(QtGui.QPixmap("icon.ico.png"), QtGui.QIcon.Normal, QtGui.QIcon.Off)
        MainWindow.setWindowIcon(icon)
        MainWindow.setToolTip("")
        MainWindow.setStatusTip("")
        MainWindow.setWhatsThis("")
        MainWindow.setAccessibleName("")
        MainWindow.setAccessibleDescription("")
        MainWindow.setAutoFillBackground(False)
        MainWindow.setWindowFilePath("")
        MainWindow.setDocumentMode(False)
        MainWindow.setTabShape(QtWidgets.QTabWidget.Rounded)
        MainWindow.setDockNestingEnabled(False)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
```

```
def retranslateUi(self, MainWindow):
    _translate = QtCore.QCoreApplication.translate
    MainWindow.setWindowTitle(_translate("MainWindow", "AeroGesture"))
    self.EnterButton.setText(_translate("MainWindow", "Enter"))
    self.EnterButton.setShortcut(_translate("MainWindow", "Return"))
    self.Eyemouselabel.setText(_translate("MainWindow", "AeroGesture - Team Sigma"))
    self.cameralabel.setText(_translate("MainWindow", "Camera Number"))
    self.lineEdit.setText(_translate("MainWindow", "1"))
    self.cameralabel_3.setText(_translate("MainWindow", "NOTE: Only one person in frame is needed for program to work fine"))
    self.cameralabel_4.setText(_translate("MainWindow", "If no result after pressing enter then camera number is wrong "))
    self.cameralabel_5.setText(_translate("MainWindow", "Inverse Camera:"))
    self.GithubButton.setShortcut(_translate("MainWindow", "Return"))
    self.cameralabel_6.setText(_translate("MainWindow", "Aspect Ratio 16:9:"))
    self.cameralabel_7.setText(_translate("MainWindow", "Illumination:"))
    self.cameralabel_8.setText(_translate("MainWindow", "Min         Mid        Max"))


if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    MainWindow = QtWidgets.QMainWindow()
    ui = Ui_MainWindow()
    ui.setupUi(MainWindow)
    MainWindow.show()
    ui.Enterbuttonclicked(True)#this is for running it on startup of program without any clicks
    # self.close()
    sys.exit(app.exec_())
```

Landmarks-Points: -

Fig 6.1 Face Ladmarks Point

Set-up: -

```
import cx_Freeze
import os
# Dependencies are automatically detected, but it might need
# fine tuning.
build_options = {'packages': [], 'excludes': []}
import cv2 as cv
cv2=cv
import dlib
import mousecontrol_eye
from win32.win32api import GetSystemMetrics
import time
from pynput.mouse import Listener,Button,Controller
import sys
from PyQt5 import QtCore, QtGui, QtWidgets
import webbrowser
import numpy
from imutils.video import WebcamVideoStream
base = 'Win32GUI' if sys.platform=='win32' else None
```

Face-Tracking: -

```
import cv2 as cv
cv2=cv
import dlib

cap=cv.VideoCapture("testvideo.mp4")
detector=dlib.get_frontal_face_detector()
predictor=dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")
def rescaleFrame(frame,scale=0.5):
    width=int(frame.shape[1]*scale)#frame.shape[1] is width of image
    height=int(frame.shape[0]*scale)#frame.shape[0] is height of image
    dimension=(width,height)
    return cv.resize(frame,dimension,interpolation=cv.INTER_AREA)
while True:
    _,frame=cap.read()
    gray=cv.cvtColor(frame,cv.COLOR_BGR2GRAY)
    gray=rescaleFrame(gray)
    frame=rescaleFrame(frame)
    faces=detector(gray)
    for face in faces:
        #print(face)
        x,y=face.left(),face.right()
        x1,y1=face.top(),face.bottom()

        cv2.rectangle(frame,(x,x1),(y,y1),(0,255,0),2)
        landmarks=predictor(gray,face)
        # x=landmarks.part(37).x
```

EyeTracking-imports: -

```
import cv2 as cv
cv2=cv
import dlib
import mousecontrol_eye
from win32.win32api import GetSystemMetrics
import time
from pynput.mouse import Listener,Button,Controller
import sys
from PyQt5 import QtCore, QtGui, QtWidgets
from imutils.video import WebcamVideoStream
import numpy as np
```

EyeTracking: -

```
class eye_mouse:
    def __init__(self,camerainput,cameracheck,aspectratio169,illumination):
        self.camerainput=int(camerainput)
        self.blinking_frames=0
        self.mousecontrol=mousecontrol_eye.mousecontrol()
        self.cameracheck=cameracheck
        self.aspectratio169=aspectratio169
        width = GetSystemMetrics(0)
        height = GetSystemMetrics(1)
        middlepoint1=width/2
        middlepoint2=height/2
        self.mousecontrol.firstpos(middlepoint1,middlepoint2)
        self.illumination=illumination
        if self.illumination==None:
            self.illumination=1
```

Choose Camera Option: -

```
import device
import cv2

def select_camera(last_index):
    number = 0
    hint = "Select a camera (0 to " + str(last_index) + "): "
    try:
        number = int(input(hint))
        # select = int(select)
    except Exception as e:
        print("It's not a number!")
        return select_camera(last_index)

    if number > last_index:
        print("Invalid number! Retry!")
        return select_camera(last_index)

    return number

def open_camera(index):
    cap = cv2.VideoCapture(index)
    return cap


def main():
    print("OpenCV version: " + cv2.__version__)

    # Get camera list
    device_list = device.getDeviceList()
    index = 0

    for name in device_list:
        print(str(index) + ': ' + name)
        index += 1

    last_index = index - 1

    if last_index < 0:
        print("No device is connected")
        return

    # Select a camera
    camera_number = select_camera(last_index)

    # Open camera
    cap = open_camera(camera_number)

    if cap.isOpened():
        width = cap.get(3) # Frame Width
        height = cap.get(4) # Frame Height
```

Face Tracking Using Shape Predictor: -

```
import cv2 as cv
cv2=cv
import dlib
cap=cv.VideoCapture("testvideo.mp4")
detector=dlib.get_frontal_face_detector()
predictor=dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")
```

6.2 TEST CASES

NO.	Test	Expected Results	Actual Result	Status
1	Running executable (application)	Interfac e opened	Interfac e opened	Pass
2	Checkbox test	Working andeditable checkbox	Working andeditable checkbox	Pass
3	Checkbox value test	Only takes an integer as camera number	Only takes an integer as camera number	Pass
4	Load Previous camera value	Successfully loads previously added value	Successfull yLoads previously added value	Pass
5	Opencv frame check	Opencv frame runs on startup	Opencv frame runs on startup	Pass
6	Aspect rationcheck	Aspect ratio checkbox works	Aspect ratio checkbox works	Pass
7	Changing aspect ratio	Aspect ratio changes to 16:9	Aspect ratio changes to 16:9	Pass

8	Inverse cameracheck	Inverse camera checkbox button works	Inverse camera checkbox button works	Pass
9	Flipping videoframe	Video gets inverted if thecheckbox is pressed	Video gets inverted if thecheckbox is pressed	Pass
10	Illumination slider	Illumination slider slides ifdragged	Illumination slider slides ifdragged	Pass
11	Illuminatio n slider works forwards	Illumination slider increases gamma value ofvideo	Illumination slider increases gamma value ofvideo	Pass
12	Illumination slider works backward	Illumination slider decreases gamma value ofvideo	Illumination slider decreases gamma value ofvideo	Pass
13	Enter button check	Enter button take press and capture values	Enter button take press and capture values	Pass
14	Value implement in OpenCV frame	After Enterbutton pressed OpenCV runs with custom settings	After Enterbutton pressed OpenCV runs with custom settings	Pass
15	Opencv frame close check	Opencv frame closes when the letter ‘Q’ is pressed	Opencv frame closes when the letter ‘Q’ is pressed	Pass

16	Face detection	Face detects And draws a rectangle around the face	Face detects anddraws a rectangle around the face	Pass
17	Cursor movemen t	The cursor moves on face movement	The cursor moves on face movement	Pass
18	Blink detection	Mouse clicks on blink detection	Mouse clicks on blink detection	Pass
19	Cursor speed check	Cursor speed gradually increases on fast face movement	Cursor speed gradually increases on fastface movement	Pass
20	Midpoint cursor movement	No movement around the midpoint of thecamera circle	No movement around the midpoint of thecamera circle	Pass

CHAPTER 7

CONCLUSION AND FUTUREWORK

7.1 Conclusion:

This application is to be and will be used for the benefit of an especially disabled individual who wants to achieve or use it in the same way as ordinary people. This application can be an essential part of the people who are lazy to use a mouse when writingwith a keyboard.

The Goal and soul purpose of making this software is to simplify the use of the hand in just one place i.e. keyboard, keyboard-centric people don't like to flip their hands to use a mouse when handling the keyboard. It can also be used by people with severe disabilities who can use their eyes to communicate to the computers.

One can see the results of what eye-tracking technology can work when we using it with an interface of a handy computer.

7.2 Advantages:

1. A one of a kind application where the cursor is controlled with a face.
2. Users can get creative and use it in-game for many game functions.
3. Any individual in distress can be helped with communication.
4. Camera input can be changed and use multiple webcams.
5. One can change illumination in video screens according to theirbackground illumination.
6. Easy to control setting and one can change aspect ratio settings.
7. Stores previously added a camera number which is used in a future boot.

7.3 Limitations:

7.3.1 Need fast computing for fast video compiling and processing.

7.3.2 Few things might need corrections in the initial stages.

7.3.3 Requires good quality camera.

7.3.4 Currently lacks extensive updating of the user profile.

7.3.5 Slow in responsein low light conditions.

7.4 Future Work:

7.4.1 In the upcoming updates, the users will be able to update their profiles to afar greater extent.

7.4.2 Will make this application for the Linux environment too.

7.4.3 The user interface will be made handier and more attractive over time.

CHAPTER 8
REFERENCES

[1] Manikandan B, Alwin S, Muhammed Shahinsha C, Sreeraj Unnikrishnan, and Abdul Raheem Bin Ummer. (2021). "Virtual Mouse Controlled by Tracking Eye Movement." International Journal of Computer Applications (IJCA).

[2] EyeMouse: Empowering Computer Interaction through Eye Gaze Tracking. (2018). IEEE Transactions on Human-Machine Systems.

[3] Virtual Mouse Using Hand and Eye Gestures. (2017). International Journal of Computer Applications (IJCA).

[4] S. Shriram, B. Nagaraj, J. Jaya, S. Shankar, and P. Ajay. (2020). "Deep Learning-Based Real-Time AI Virtual Mouse System Using Computer Vision to Avoid COVID-19 Spread." International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE).

[5] "Eye-Controlled Virtual Mouse for People with Disabilities". (2019). Journal of Assistive Technologies.