Faculty of Computer and Information
Information Technology Dept

Mansoura University

Computer Interaction System for Handicapped People Controlled by Iris Movements.

(A.I.EYE)

A Project submitted in partial fulfilment of the requirements for the degree of bachelor of Science in Information Technology.

Supervised by

Prof . Dr . Mohammed Elmogy

Eng . Yasmin Alsakar

2021/2022

SUPERVISOR CERTIFICATION

I certify that the preparation of this project entitled

…….. **,**

was made under my supervision in the Information Technology Branch / Mansoura University in partial fulfillment of the requirements for the Bachelor of Science degree in the College of Computing and Information - Department of Information Technology.

Signature:

Name:

Date:

DEDICATION

Praise be to God, who taught man what he does not know, praise be to God who guided us to this, and we would not have been guided had it not been for God guided us. I dedicate this project to everyone who wants to learn and be a reason for the renaissance of this nation
For people with special needs and anyone who can't use a computer To my parents and to our family who made this accomplishment possible
May God accept from us that you are the All-Hearing, the All-Knowing

Team work

1- Ahmed Salah Aboshendy
2- Ahmed Maher Elataby
3- Hazzem Ahmed Abdelatif
4- Ahmed Magdy Tawfiq
5- Ahmed Elsayed Khadr
6- Ahmed Atef Khamis
7- Shaimaa gamal mahmoud
8- Manar Maher Mahmoud
9- Ahmed Saber elhawary
10- Ahmed Saad Mohamed

Acknowledement

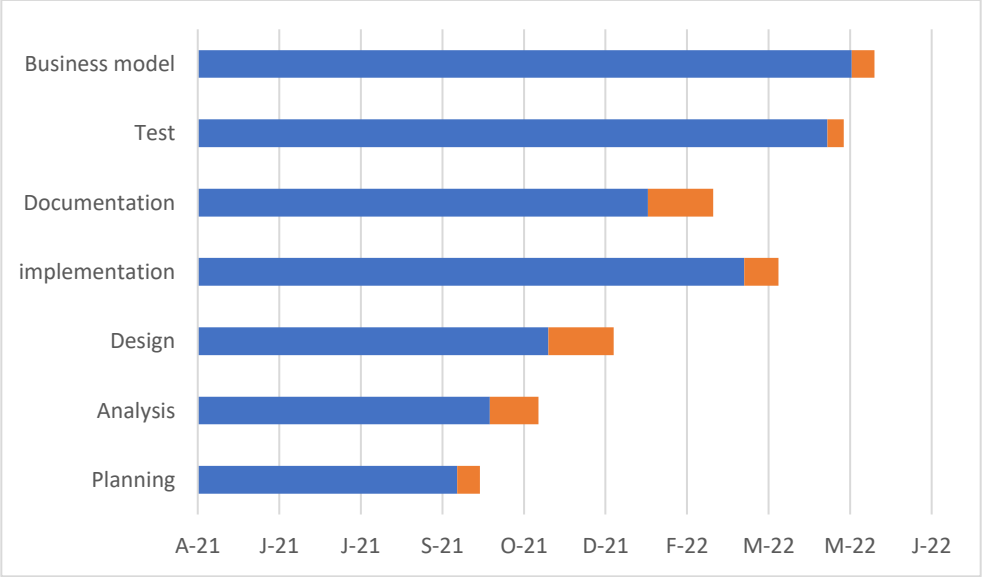
First and foremost, I would like to thank the chairman of our committee,

Prof . Dr . Mohammed Elmogy for his support, outstanding guidance and encouragement throughout my senior project. I would also like to express our gratitude and appreciation to Eng . Yasmin Alsakar for all the help and guidance he provided throughout my education, I would like to thank our family, especially my parents, for their encouragement, patience, and assistance over the years. We are forever indebted to our parents, who have always kept me in their prayer

Timeline

id	Task name	Project start	Duration	End
1	Planning	2021/09/20	14	2021/10/04
2	Analysis	2021/10/10	30	2021/11/09
3	Design	2021/11/15	40	2021/12/25
4	Implementation	2022/03/15	21	2022/04/05
5	Documentation	2022/01/15	40	2022/02/24
6	Test	2022/05/05	10	2022/05/15
7	Business model	2022/05/20	14	2022/06/03

Table 1.1 projects' Timeline



Abstract

.

Our program makes you dispense with the computer mouse by replacing it with the movement of eye to move mouse cursor so that you can only with your eyes do all functions of the computer mouse for example Scrolling up and down, it is to help people with special needs and those who have disabilities in their hands or those who suffer from diseases such as Parkinson's or total paralysis and other diseases that cause disabilities from the use of the mouse. And therefore, their inability to use the computer and perform tasks on it here comes an attempt from us to provide a way to use the mouse and do all the tasks of the mouse without catching it and simulating all the actions of the mouse, which is by using the human eye to carry out these actions

And therefore the user will not need any movement effort of the body everything on it is only a simple movement of his eyes to move the cursor and carry out all the tasks of the mouse, Our program makes you dispense with the computer mouse by replacing it with the movement of eye to move mouse cursor so that you can only with your eyes do all functions of the computer mouse, it is to help people with special needs. normal people can also use that application to increase well use the mouse. children can use aieye for being without the trouble and need to interactive computer gaming. our project depends on how to implement the actions of the mouse and its functions and simulate them to the human eye.

TABLE OF CONTENTS

List of Tables

List of Figures

Glossary

Abbreviation	Definition
A. I	Artificial Intelligence
ALS	Amyotrophic Lateral Sclerosis
MND	Motor Neuron Disease
GUI	Graphical user interface
UML	Unified Modelling Language

Chapter 1

Introduction

1.1 Introduction

The idea of the project is a program for human-computer interaction applications, which is how to move the mouse pointer through eye movement, especially the iris of the eye, and control its movement on a screen. And because the use of the mouse is of great importance and occupies almost all the tasks used in the computer, there are some people who cannot use the computer, because they are also unable to use the mouse.

The main purpose of this project was to help people with special needs and those who have disabilities in their hands or those who suffer from diseases such as Parkinson's or total paralysis and other diseases that cause disabilities from the use of the mouse. And therefore, their inability to use the computer and perform tasks on it here comes an attempt from us to provide a way to use the mouse and do all the tasks of the mouse without catching it and simulating all the actions of the mouse, which is by using the human eye to carry out these actions and therefore the user will not need any movement effort of the body everything on it is only a simple movement of his eyes to move the cursor and carry out all the tasks of the mouse.

Normal people can also use that application to increase well-being without the trouble and need to use the mouse. children can use ai-eye for interactive computer gaming. so, we will not need to use the mouse in our lives again and we will not need any other devices to implement the application and all our dependence is on the webcam that we have in the device is only for tracking and executing eye movements. our project depends on how to implement the actions of the mouse and its functions and simulate them to the human eye. Here is an illustration with examples:

First: We start with a slight movement of the eye (the iris) in all directions, which leads to moving the cursor in all directions and the places we have on the screen.

Second: Right and left clicking of the mouse and double clicking we will replace them with the movements of closing and opening the mouth and the left eye (winking) or closing one of the eyes twice in order.

Third: Simulation of scrolling up and down, by focusing and looking long at the top of the screen to scroll up, Focus and long look at the bottom of the screen to scroll down.

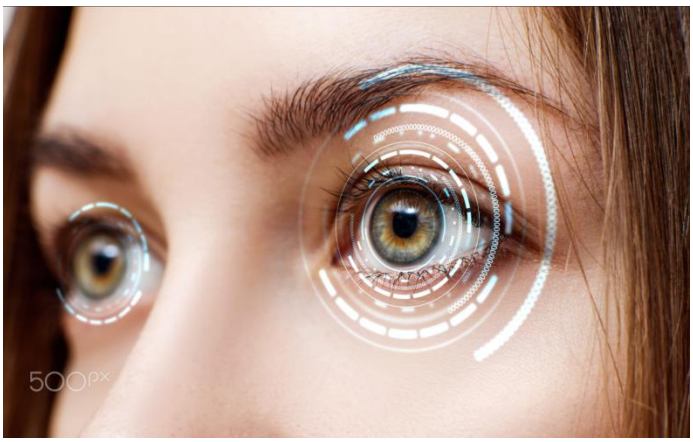


Fig 1.1 shape of human iris

1.2 Problem statement

I. How to move the mouse pointer through eye movement, especially the iris of the eye, and control its movement on a screen.

II. There are people with special needs and those who have disabilities in their hands or those who suffer from diseases such as Parkinson's or total paralysis and other diseases that cause disabilities from the use of the mouse.

III. People are unable to use the computer and perform tasks on it,

1.3 Motivation's statement

I. Help disabled and paralyzed people and people in need to use computer devices.
II. In previous programs ,it was only based on head movement.
III. As a result ,it leads to negative results that affect the user such as severe dizziness, headache and may lead to pain in the neck and spine.
IV. And those who have disabilities in their hands or those who suffer from diseases such as Parkinson's or total paralysis and other diseases that cause disabilities from the use of the mouse. and help them to perform tasks.

1.4 Objectives section

- An eye worldwide situating system is exceptional contrasted with other application for the obstructed person. To develop this application distinctive figuring procedures and strategies for the image taking care of are used.
- These methodologies and strategies for picture getting ready gives an overall arranged model for eye worldwide situating system. This application is important for the face area, features acknowledgment, format age and cursor advancement. As we are entirely remarkable about the debilitations.
- In any case the individual can't use their body parts they are to be consider as the cripple individual for such people here we will familiarize a methodology with partner them with this current reality. To offer people with unbelievable inadequacies, an opportunity to control a PC fundamentally by moving his/her eyes or head. To design an insignificant exertion joined eye and head worldwide situating structure for individuals with insufficiency of their upper extremities.
- Gives smoother movement
- Unable ordinary people to use this application to increase well-being without the trouble and need to use the mouse and so we will not need to use the mouse in our lives again and we will not need any other devices to implement the application and all our dependence is on the webcam that we have in the device is only for tracking and executing eye movements.

Fig 1.2 detection of the iris

1.5 Contributions section

1- We developed this program with the intention to help people in serious need of it.
2- People with disabilities struggle daily to perform tasks and developing an application that would allow them to use the computer will be of great effect on their daily life.
3- By giving them the opportunity to control the mouse cursor through the movement of the iris we hope that would give them the chance to perform any computer task they need or get work done.
4- Since using computers is inevitable nowadays.
5- It's also suitable and available to be used by everyone.
6- Any person can use this application to multitask.
7- You could be doing something while doing another task on the computer and all you need to do is move your iris to control the cursor on the screen.
8- It saves time, effort and is very efficient to use.

1.6 Project scope

It is a program to replace the mouse by moving the cursor by moving the eye of the human being it performs all the functions of the mouse such as (scrolling up and down, Opening and closing a folder clicking once and double-clicking ... etc.) And applying this program on a computer equipped with a webcam contained in the device. No additional hardware is required Users: for people with special needs and those who do not have the ability to move hands, and this program can also be used for the normal person to increase well-being without the need to hold the mouse. children can use ai-eye for interactive computer gaming. This program can be applied to smart phones and tablets at the future work.

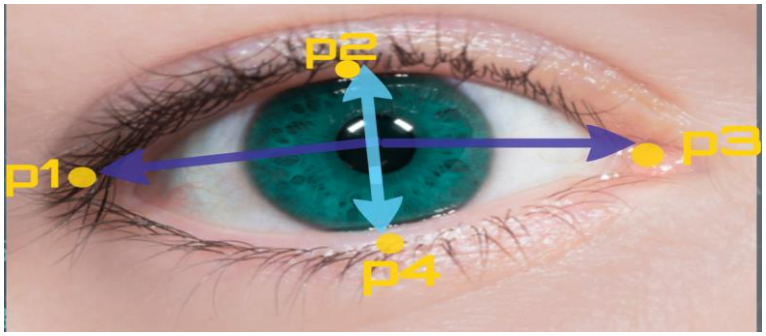


Fig 1.3 how the camera read the iris' movement

Chapter 2

Literature Review

2.1 **Introduction:**

In this chapter, we will review the previous related work to our project, and we will discuss the problems that faced the previous ones, how we will solve these problems, and add more features to this project. At the end of this chapter, we aim to see how we will enhance our project and make it better by decreasing the project's disadvantages and making it easier for the users.

2.2 **Review of relevant projects:**

The following are some projects and programs whose goals are to help the disabled in using computers without needing the help of others by moving the mouse cursor by their head's reactions and eyes' movement. we will talk about abstracts and their pros and cons and how we will avoid them. we will use the previous mistakes to make our project easier and more useful to our users

2.2.1 **Smyle Mouse:**

Smyle Mouse is a software package that uses your webcam to track your head and face movement, giving you hands-free cursor control with no need for additional hardware or accessories and it is suitable for users with acquired and progressive conditions that limit their movements, such as muscular dystrophy, spinal cord injury or MND/ALS.

Fig 2.1 smyle mouse interface

2.2.2 **Camera Mouse**

Camera Mouse is a free program that enables you to control the mouse pointer on your computer screen just by moving your head (or foot or finger).

The "Camera Mouse" system has been developed to provide computer access for people with severe disabilities.

The system tracks the computer user's movements with a video camera and translates them into the movements of the mouse pointer on the screen.

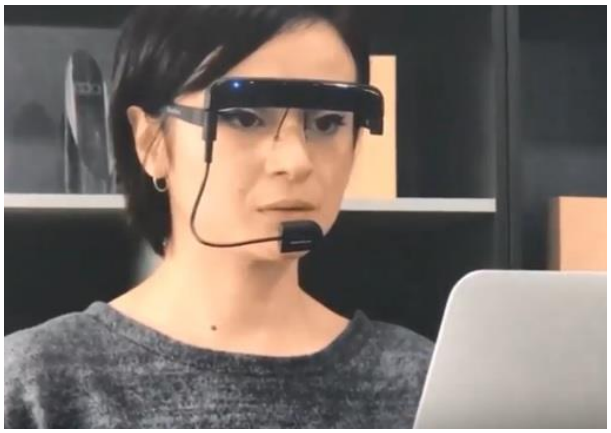


Fig 2.2 moving your head for cursor controlling

2.2.3 **Replacement of the Standard Computer Keyboard and Mouse by Eye Blinks:**

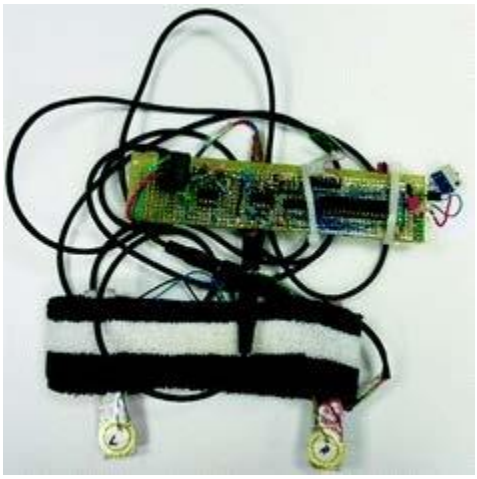


Fig 2.4 electric circuit to detect blinks

Individuals with physical disabilities faced a lot of difficulties regarding this, and that's why hundreds of research, theses, and projects were made to solve this problem. In this project, they mainly focused on replacing the physical keyboard and mouse with almost a virtual one. They developed a software tool called On Screen Dual Scribe, it's a headband with sensors and physical keys. This tool does not only give the ability to type but also to do some mouse functions on the screen like clicking on objects and selecting them.

2.3 projects' pros and cons

we can now after describing an abstract about each project describe pros and cons so we can know more about their project, we can improve our project and learn how to solve these problems if we face them.

Project	Pros	Cons
Smyle mouse	Many of the people who use this software program say about it many positive feedbacks. for example, 1- it gives chance tocompete with other applicants for the disabled. 2- Too easy to learn compared to the other eye-tracking programs.	1- Users face some mouse display issues wherein the display of the mouse cursor degraded after every smile 2- E.g., the mouse cursor would shrink in size or become blurry after smile clicking. 3- The problem of face tracking is when it is lost upon the user turning their head away from the camera. 4- Click Options window takes a huge space in the monitor, and it will limit the face's actions as shown in Fig (1). 5- those were some of the problems that faced me when I tried this program, and we will avoid these problems as much as we can.
Camera mouse	1- Camera mouse guaranteed 100% clean this mean no virus on software program and it achieved by soft-pedal. 2- Camera Mouse has a built-in facility for clicking using "dwell time". That is, if you hold the mouse pointer over a button or icon for a set amount of time, usually a second, then the Camera Mouse program will click on it.	1- Not very accurate in detectface'smovement whencameraismoving even a little so camera should be in table or desk and that limit personal freedom to user. 2- There is no version to mac users. 3- If people cannot move their head this will be problem because it is tracks head movement 4- so they will need eye control system, so it makes this program not very professional.

Replacement of the standard computer keyboard and mouse by eye blink	1- It got the job done, for replacing both the computer keyboard and mouse. so, some individuals could take advantage of it according to their needs. 2- It has an evaluation software to measure the accuracy and speed of writing, which in return raises the quality of the project.	1- Even though they used software to evaluate the accuracy, as this tool depends on the movement of muscles solely it cannot be very accurate, so the error rate cannot be ignored. 2- It's not virtual, but mainly depends on a physical tool to detect the eyes' movement, and even though the headband doesn't use that many sensors. 3- it's still big for a tool that would be placed on the head of the users considering that it will be used daily. 4- It also has about 16 physical keys and a circuit. 5- And taking into consideration that this product was directed towards disabled individuals 6- , the idea of using a headband doesn't seem like the best choice.

2.3 **Summary:**

The previous section shows the most important projects and programs that were applied in our life for helping disabled people in their daily life. section 2.1 shows the chapter's introduction, section 2.2 discusses the relevant projects and how developers make an effort to make disabled people's lives easier on the level of the technical side of their life, advantages and disadvantages, and the basic tools that help them to make their projects and how it will benefit us in programming our project.
We seek to help the disabled and we will spare no effort to make this project -as much as possible- to make our users happy and satisfied with this opportunity that brought us together.

Chapter 3

System Analysis

3.1 Introduction:

The main objective of this chapter is to provide a concise idea of the system. A comprehensive description of the system's functional and non-functional requirements, along with the user requirement. Furthermore, we will mention the tools and the languages used to develop this system.

3.2 User and system requirements:

It is divided into two parts:

1) Functional requirements.
2) Non-functional requirements.

3.2.1 functional requirement:

Is a list that describes the services the system must offer and how it's supposed to behave and function.

3.2.1.1 User functional requirement:

- The user should be able to register on the system and make a personal account.
- The user should be able to login back to the system at any time smoothly.
- The user's login data and credentialed should be saved.
- The user should stay logged in as long as they did not deliberately log out of the system.
- In case the user forgot their password, they should be able to retrieve it.
- The user should be able to use the software as long as they want without any malfunctions in the system.
- The user's eye movements should be easily and rapidly recognized by the system.

3.2.2 System non-functional requirements:

- This section mentions the non-functional requirements that should be considered while working on this application, to enhance the overall quality of the system.

The non-functional requirements of the system.

- Availability: This application should be available all the time constantly.

- Reliability: It should be always reliable and be reliable to all possible users.

- Security: It should be secure enough for the users to use, so confidential data like the user's login credentials should be secured and to be able to prevent any unauthorized access to the data.

- Internationalization: this application should be user-friendly to people from different backgrounds and recognize people from all ethnicities.

- Scalability: the applications should be flexible and adapt to new changes and updates.

- Speed: the system should function quickly and not take too long to process the data.

- Error-handling: The system should be able to handle and survive unexpected situations.
- Data recovery: the system should be able to recover the data after a blackout or a disconnection.
- Logging tracking: the system should monitor and keep up with its login activity, as this can be used for further purposes.

3.3 System architecture:

The project is a program for human-computer interaction applications, which is how to move the mouse pointer through eye movement, especially the iris of the eye, and control its movement on a screen and motion of iris of the eye is done by six extraocular muscles.

3.3.1 System requirement:

Function	
Description	the project is a program for human-computer interaction applications and replace the mouse by moving the cursor by moving the eye of the human
Input	The eyes of the human
source	1. Camera 2. Eyes of the human
Output	Do all mouse tasks (Implement the actions of the mouse and its functions)

Action	1. We start with a slight movement of the eye (the iris) in all directions, which leads to moving the cursor in all directions and the places we have on the screen. 2. Right and left clicking of the mouse and double-clicking we will replace them with the movements of closing the right eye and the left eye (winking) or closing one of the eyes twice in order. 3. Simulation of the screen to scroll up, Focus, and long look at the bottom of the screen to scroll down

3.4 Development methodology:

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

3.4.1 Use Case Diagram:

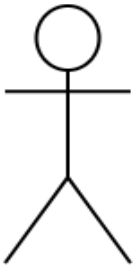
Describes the function side of the project which mean the actors and how they will interact with the program in the theoretical part

The use case does not expect the exact of how the user interacts with the program but specify the expected behavior

The main three components of the use case diagram are:

3.4.1.1 Actors:

They are the users who will use our software program who our program directed to and by users, I mean individuals persons, or organizations.



3.4.1.2 Use Case:

In this section, we discuss what is the action that users can take toward the program how to deal with and learn the system and what is the goal form that interacts between the user and the system.

3.4.1.3 Relationships

It describes the relation between actor and use case, there are types, and they are: -

-Include relation describes one use case.

-Extend relation describes the hidden relation in the use case.

- Generalization relation between two use cases.

-Association relation the role of the actor in the use case.

3.4.1.4 Use Case Diagram of The System:

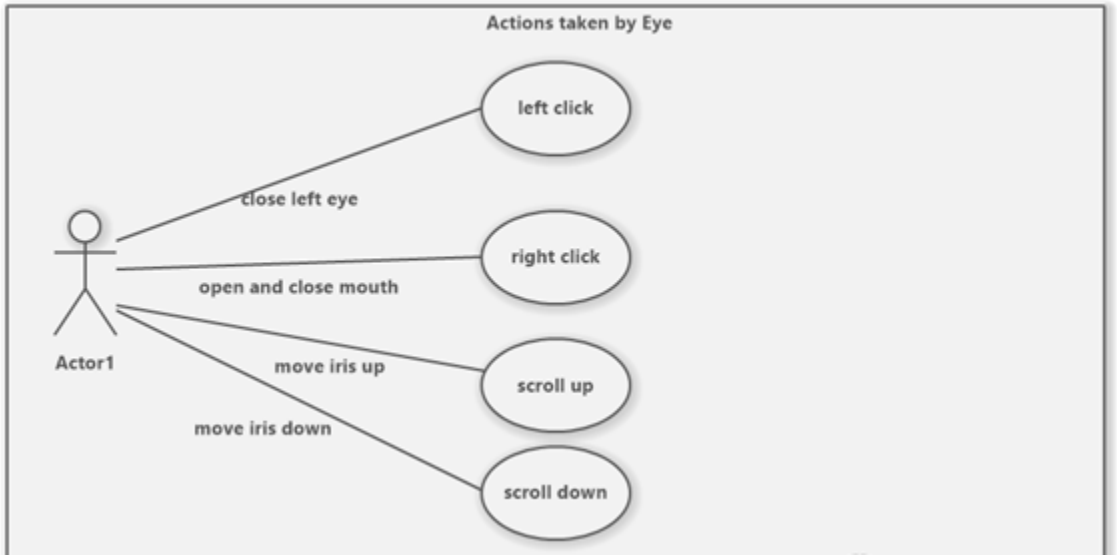


Fig 3.4.1.1 system use case diagram

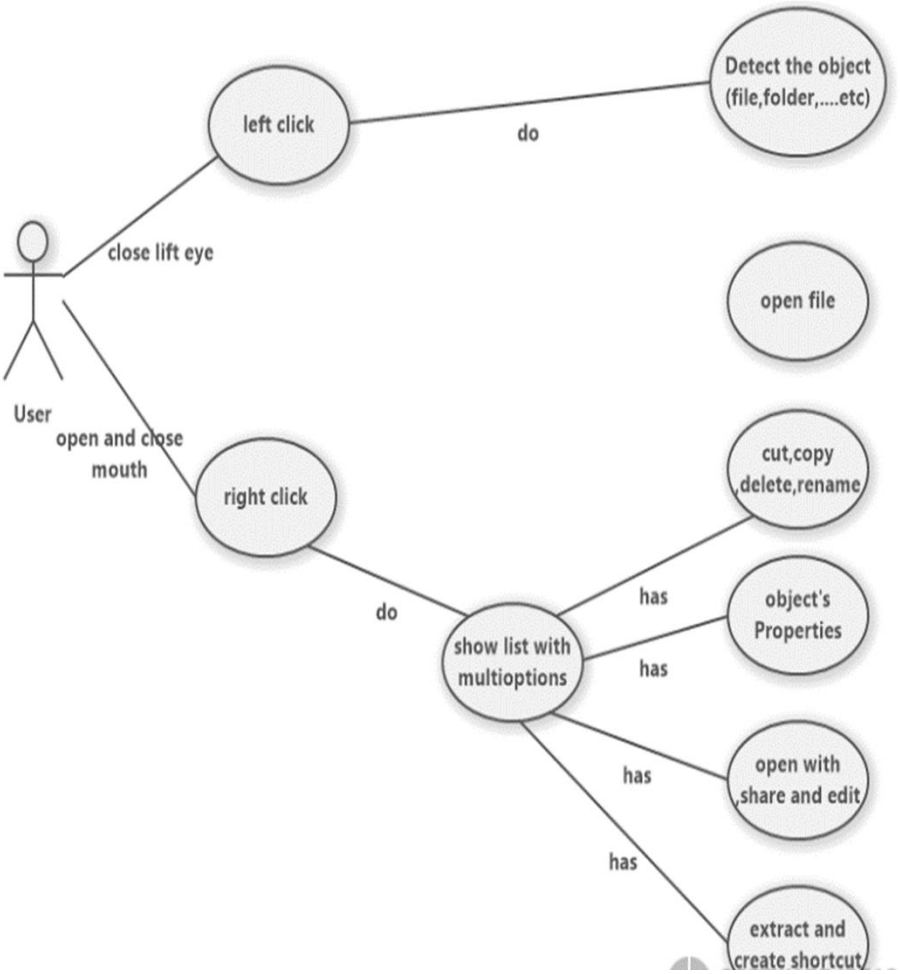


Fig 3.4.1.2 right/left click function use case diagram

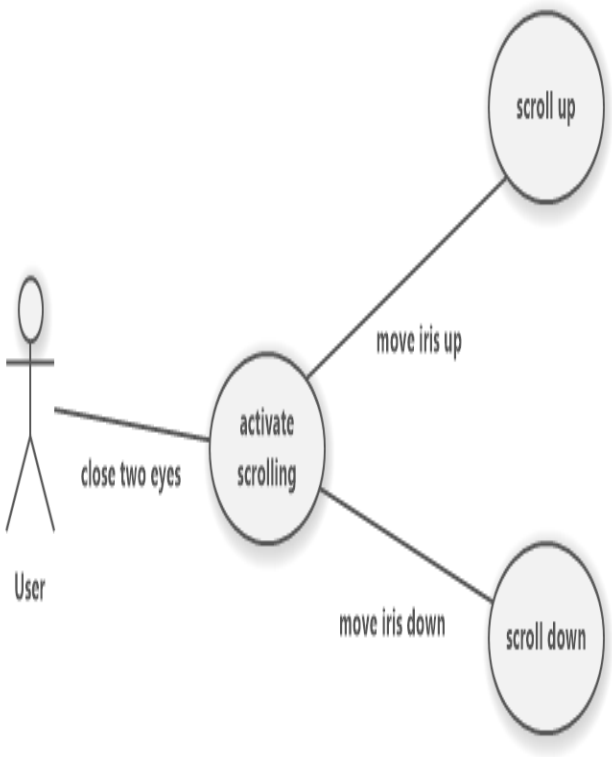


Fig 3.4.1.3 scrolling function use case diagram

3.4.2 Sequence Diagram

A Sequence diagram is an interaction diagram that shows the relation between the objects participating in a particular interaction and the messages that are exchanged in the time sequence.

This diagram is also known as Event diagrams which helps in understanding the objects, we use It has two dimensions time which is represented in vertical and objects which is in Horizontal.

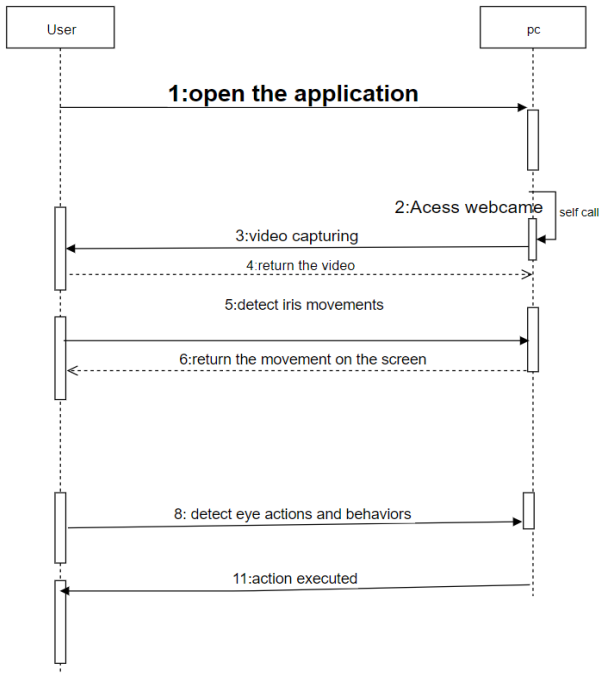


Fig 3.4.2.1 system sequence diagram

In this Fig 3.4.2.1, it shows the simple initial operations of dealing with the program, which begins with opening the program and accessing the webcam Then the program tracks and determines the movement of the eye And then it performs the actions.

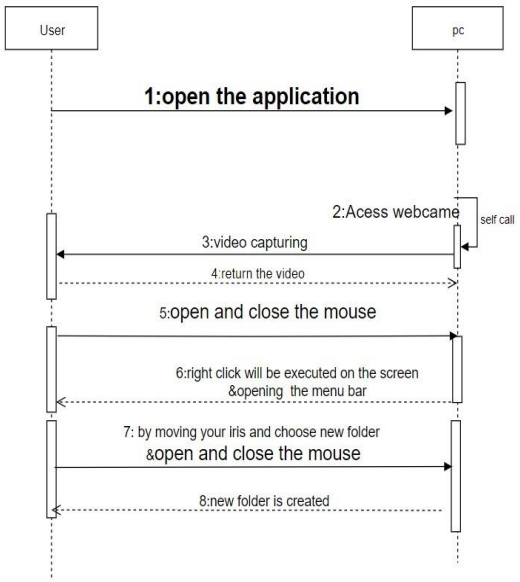


Fig 3.4.2.2 system sequence diagram

Right &left click

Fig 3.4.2.2. shows how to perform left and right clicking using the left and right eye, which is by how to create a folder, for example

We open and close the mouth, which we represent in the mouse by right-clicking, and thus the menu bar window will open and move the eye towards the window and choose (new folder) by winking with the left eye, and thus a folder will be created by eye.

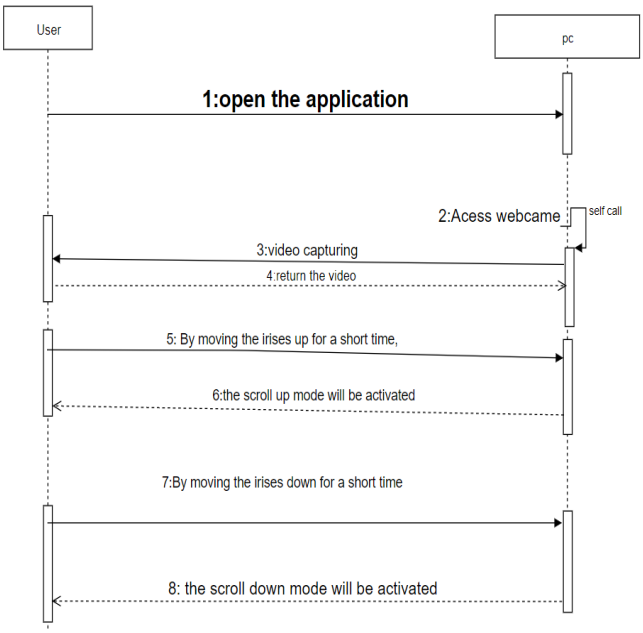


Fig 3.4.2.3 system sequence diagram

Scroll up & scroll down

Figure 3.4.2.3. shows how to scroll up and scroll down using the eyes. It is by moving the eye and looking up for a short period to activate the scrolling mode up. These operations are performed by looking down to activate the scroll down mode.

Business Canvas Model

Defined as plan fro the successful operationof a buisness,identifying sources of revenue,the target customers base,products and details of financing.
we know that buisness canvas model provide us way to create a clear buisness model using just a single sheet of paper and how the key drivers of the buisness fit together, so in this section we will discuss how to promote our product which represent in application desktop, how to gain money from this app, how we will deal with our customers and ways help customer to contact us if they need any inquiry.
The best thing about business canvas model it can used by any company in the world which make it a global scale.

So why would we use business canvas model?
Because of what it holds of many advantages and including:

- Easy to understand.
- Focussed removes all the fluff that mighet have been present in a traditional buisness.
- Flexible quick and easy to make changes and sketch out new ideas .
- Customer focused because make you think about the value of customers.
- Shows connection .
- Easy to comminucate that it is easy to be able share and explain with your team.

Business canvas model consist of 9 blocks:

1. Customer segments (who are the customers?).
2. Value proposition (why Customer deal and buy from you?).
3. Channels (how are products and servers delivered to the market?).
4. Customer relationships (how do you get and grow up your customers?).
5. Revenue streams (how does business earn mony?).
6. Key resources (what unique strategic resources does buisness have?).
7. Key activities (what unique strategic activities the business perform to deliver value propostion?).

8. Key partnerships (what non key activities can make you outsource to enable you to focus more on your key activities?).
9. Cost structure (what are the major costs incurred by the business?).

The left hand side of model represent costs to the business and right hand side represent revenue and customer realtionships for the business.

Firstly, we will discuss the problem that can be said in two points:-

1) people who can't use their hands for using the mouse and this make some problems when they deal with any compute.
2) also, normal people can use it when they stand at far away from the computer by moving their iris only in their place.
3) Help children in easy interactive with computer when they want play games and surf the internet.
4) Old people want to use computers without any extra effort or more movement.

Secondly, people who will use desktop application and we said they are two types:-

1)the disables who want to use computer more comfortable and easier without help of anyone else
2) normal people who want deal with computer with more comfortable way without needing mouse in case the mouse is not working well

The software project focuses in the first type (the disables) who is considered the most important class of people who want really to deal with computer without any obstruction hinder their usage of computer.

The goal of the project help disabled to use computer and this by moving their iris, they can move the cursor in the screen and simulate the mouse actions just by that.

The IBA's Global Business Model Canvas	*Designed for:*	*Designed by:*	*Date:*	*Version:*
	Disabled	AI eye team	20/5/22	1.1

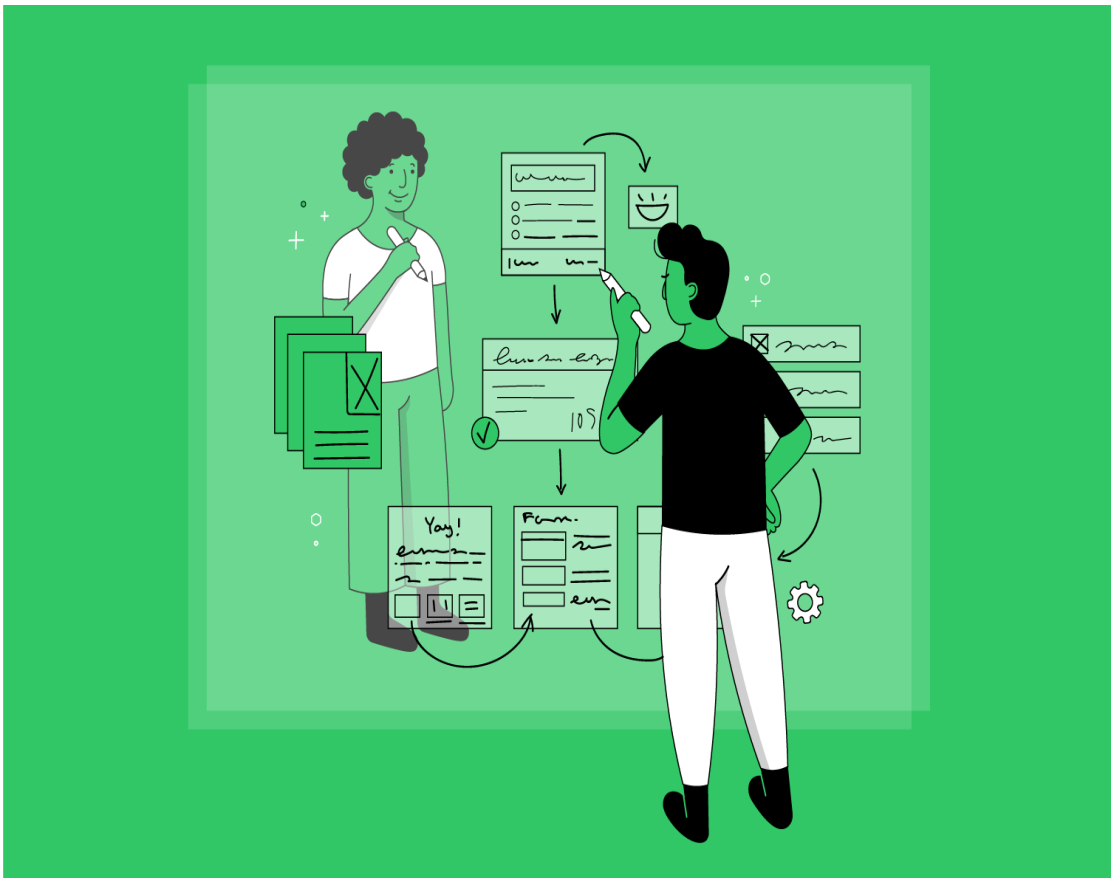
Key Partners	**Key Activities**	**Value Propositions**	**Customer Relationships**	**Customer Segments**
Companies help in build up and maintain the application Adverisment caompanies to sponsore to our product	Develop and maintaince the app regularly Marketing and digital marketing Platform development Customer services Creating technilogical infrastrucutre	AI eye provide:- Helping disable in their daily life when dealing with computer Cursor respond to your iris faster and more accuracy in detect the objects Moving cursor with just iris no need to move your head and cause aheadache or pain in your neck Ability to customize and price Simulator in Interactive computer games	Personal assistance Automated and self services Raised the morale of cutomers when the purchased Monthly newsletter of latest services to grow customers	Disabled who want to use computer without help of another people in any location Normal people who want to use computer in easier way without any extra effort Children old people who want to use computer eaiser without nedding to move alot
	Key Resources Language programming (python) Modules(media pipe,opencv,cv zone,pyautogui) Team members		**Channels** Website Social media Events to Increase awarness between customers about the benefit of the product Providing post-purchase customer support	

Cost Structure	**Revenue Streams**
Technical set-up and opertional costs	Licensing
Salaries for permanent employees	Sponsored ads
Performance of the software	Usage fee(every year)
	Upgrade to premium

Table 1.1.1 Business Canvas Model

Chapter 4

System Design



4.1 Introduction:

The key objective of this chapter is to provide a clear understanding of our system, including the system design, structure, and behavior to provide a thorough view of our system. In addition to providing diagrams and user interface storyline to illustrate how the system looks and functions.

4.2 Class diagram:

This is a type of diagram defined by the unified modeling language (UML), which is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the methods use. It's not only used to describe and visualize different aspects of the system from different views but also used for constructing executable code for the software application. The classes in a class diagram are arranged in groups, and each group shares some common characteristics as they all are interrelated to each other in some ways.

4.2.1 Relationships:

Relationships in UML diagrams are connections between model elements, so each relationship includes a different type of logical connection, and they are divided into categories as shown below.

4.2.2 Dependency:

It is a connection between dependent and independent model elements. A dependency relationship means that one element uses or depends on another element.

4.2.3 Association:

It represents a logical connection between two or more classes and is represented by a link between the connected peer classes. It covers the grounds of a relationship and the rules governing the relationship. And it could be either directed or reflexive.

4.2.4 Aggregation:

It's a special type of association, and it represents a part of the relationship. It's usually represented as a link between a parent class to a child class.

4.2.5 Composition:

Is a special type of aggregation relationship. The key difference between a composition and aggregation relationship is that the composition one, emphasizes the dependency of the contained class to the whole life cycle of the container class. To put it simply if the parts are destroyed then the whole is destroyed. As the contained class can't live nor stand by itself. Class-level relationships.

4.2.6 Generalization:

This relationship is also known as the inheritance relationship, in which a model element is based on another model element. Like the child being totally based on the parent, inheriting all of its attributes and functionalities.

4.2.7 Realization:

In UML it is defined as the relationship between two model elements. In which one element realizes the behavior that the other element specifies, like between the client and the supplier. It's represented as an unfilled link between the class that defines the functionality of the class that implements the function.

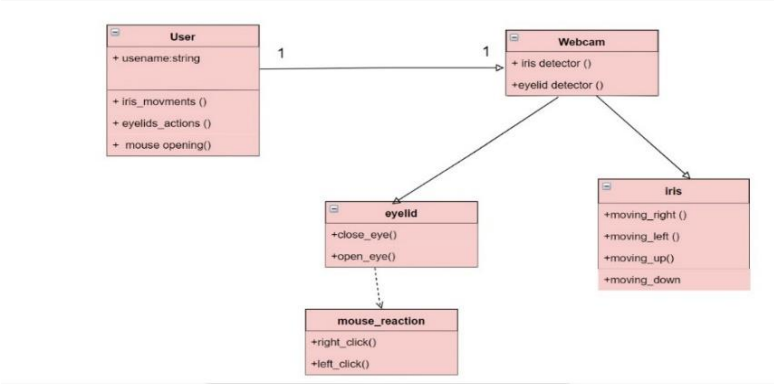


Fig 4.2.1 class diagram for the program

4.3 System Design:

Our system design can represent in logo whereas it is just application desktop and that is in early designs for the project so we can enhance it more and more in the future.

Fig 4.3.1 design of system logo

Chapter 5

System Implementation

pyaotugui install steps:

5.1.7.1 installation:

To install PyAutoGUI, install the pyautogui package from PyPI by running pip install pyautogui (on Windows) or pip3 install pyautogui (on macOS and Linux). (On macOS and Linux, pip refers to Python 2's pip tool.) OS-specific instructions are below.

5.1.7.1.1 windows:

On Windows, you can use the py.exe program to run the latest version of Python:
py -m pip install pyautogui If you have multiple versions of Python installed, you can select which one with a command line argument to py. For example, for Python 3.8, run:
py -3.8 -m pip install pyautogui (This is the same as running pip install pyautogui.).

5.7.1.1.2 macOS :

On macOS and Linux, you need to run python3:
python3 -m pip install pyautogui If you are running El Capitan and have problems installing pyobjc try: MACOSX_DEPLOYMENT_TARGET=10.11 pip install pyobjc.

5.1.7.1.3 Linux:

On macOS and Linux, you need to run python3:
python3 -m pip install pyautogui On Linux, additionally you need to install the scrot application, as well as Tkinter:
sudo apt-get install scrot
sudo apt-get install python3-tk
sudo apt-get install python3-dev

Pyaotugui install the modules it depends on, including PyTweening, PyScreeze, PyGetWindow, PymsgBox, and MouseInfo.

OpenCV install steps:

1. Installing OpenCV from prebuilt binaries

1. Below Python packages are to be downloaded and installed to their default locations.
    a. Python 3.x (3.4+) or Python 2.7.x .
    b. Numpy package (for example, using pip install numpy command).
    c. Matplotlib (pip install matplotlib) (*Matplotlib is optional but recommended since we use it a lot in our tutorials*).
2. Install all packages into their default locations. Python will be installed to C:/Python27/ in case of Python 2.7.
3. After installation, open Python IDLE. Enter **import numpy** and make sure Numpy is working fine.
4. Download latest OpenCV release from GitHub or SourceForge site and double-click to extract it.
5. Goto **opencv/build/python/2.7** folder.
6. Copy **cv2.pyd** to **C:/Python27/lib/site-packages**.
7. Open Python IDLE and type following codes in Python terminal.

```
>>> import cv2 as cv
>>> print( cv.__version__ ).
```

2. Install openCV for ubuntu

We need **CMake** to configure the installation, **GCC** for compilation, **Python-devel** and **Numpy** for building Python bindings etc.

```
sudo apt-get install cmake
sudo apt-get install gcc g++
```

to support python2:

```
sudo apt-get install python-dev python-numpy
```

to support python3:

```
sudo apt-get install python3-dev python3-numpy
```

Next we need **GTK** support for GUI features, Camera support (v4l), Media Support (ffmpeg,gstreamer) etc.

```
sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev
sudo apt-get install libgstreamer-plugins-base1.0-dev libgstreamer1.0-dev
```

to support gtk2:

```
sudo apt-get install libgtk2.0-dev
```

to support gtk3:

```
sudo apt-get install libgtk-3-dev
```

3. install OpenCV for windows using pip

n pip package management, there are 4 different OpenCV packages all using the same namespace, cv2. Although they are not officially supported by OpenCV.org, they are commonly used in developers' community. You could install any of them using the following command:

pip install PACKAGE_NAME
where PACKAGE_NAME can be

- opencv-python (only contains main modules)
- opencv-contrib-python (contains both main and contrib modules)
- opencv-python-headless (same as opencv-python but without GUI functionality)
- opencv-contrib-python-headless (same as opencv-contrib-python but without GUI functionality)

Convert code to executable desktop application:

Firstly, we need setup python file to declare the version of the project, to make the file we need to install package index by using command pip install inda and then we create the file setup.py which contain every all the information about the desktop application.

We need to cx_Freeze library to work directly with modules setup and Executable to produce normally folder containing an executable file.

python modules which used in whole project are stored in Zip file and the packages by default in the file system and can be stored in Zip file too.

In the setup python file we declare the name of the app, options which contain the packages used in the program, the version, the executable file (code file).

```
from cx_Freeze import setup, Executable

base = None

executables = [Executable("main.py", base=base)]

packages = ["idna"]
options = {
    'build_exe': {
        'packages':packages,
    },
}

setup(
    name = "AI Eye",
    options = options,
    version = "1.0",
    description = 'AI Eye desktop application version 1.0',
    executables = executables
)
```

Fig 5.1 Setup python file

Secondly by using the command python setup.py build we can now execute the source code and turn it into desktop application which user can open the program normally in his/her computer at any time.

So how we convert using that command:

- we make sure that code is run normally and that by checking all libraries and modules are installed in the environment (place which we build the program and supply it with all modules program need) and to install package we can use command pip install package name if you use windows, for MAC software we use pip3 install package name and if you use anaconda use the command condo install package name , there is simple way to install packages if you use PyCharm and that by install package by clicking the "+" symbol and you can find it in settings >>interpreter and then search the package you want to install

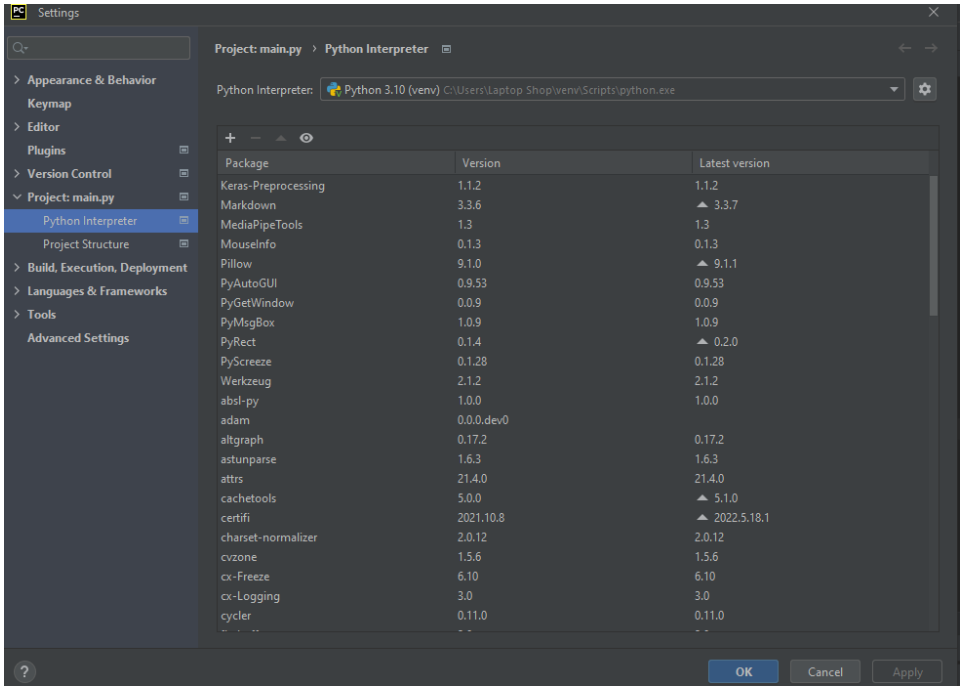


Fig 5.2 Install packages manually in pycharm

- after we run code and work successfully, we start to run the command we mentioned before python setup.py build (use the setup.py file to make configuration to the application) after that command run successfully, we found one more file added to project file named build and that contain

    1. lib file contains every library used in the program code.
    2. AI Eye file the application.
    3. Python3.dll and python310.dll application extensions files acronym for Dynamic Link Library file and used to create extensions libraries and found in system32 on the computer.

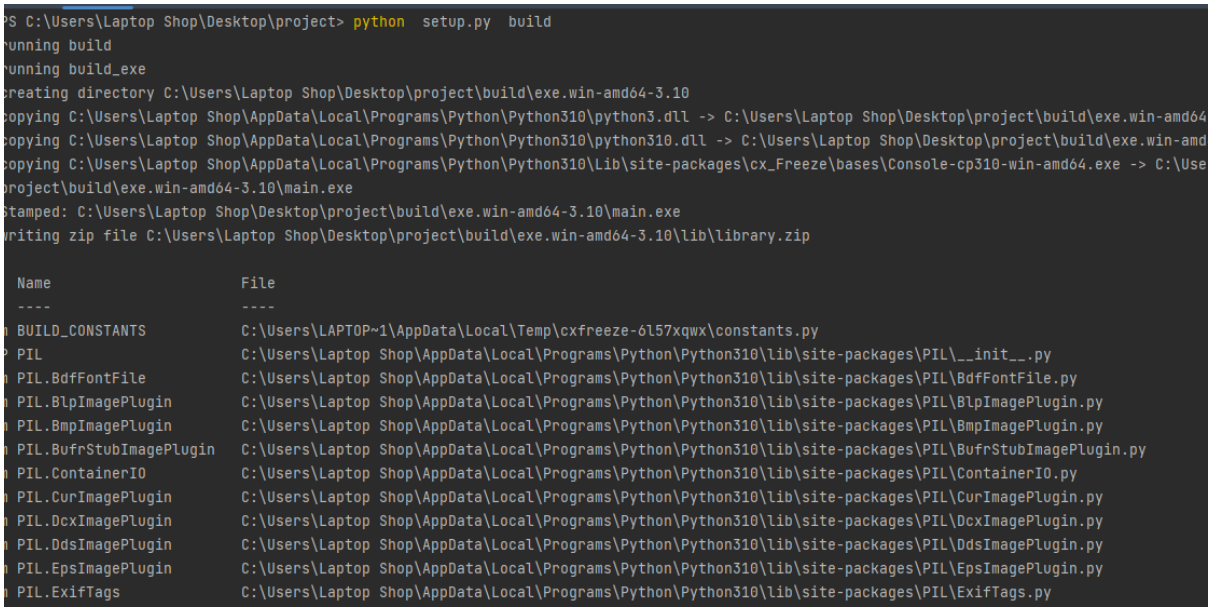


Fig 5.3 Python setup.py build command run in template

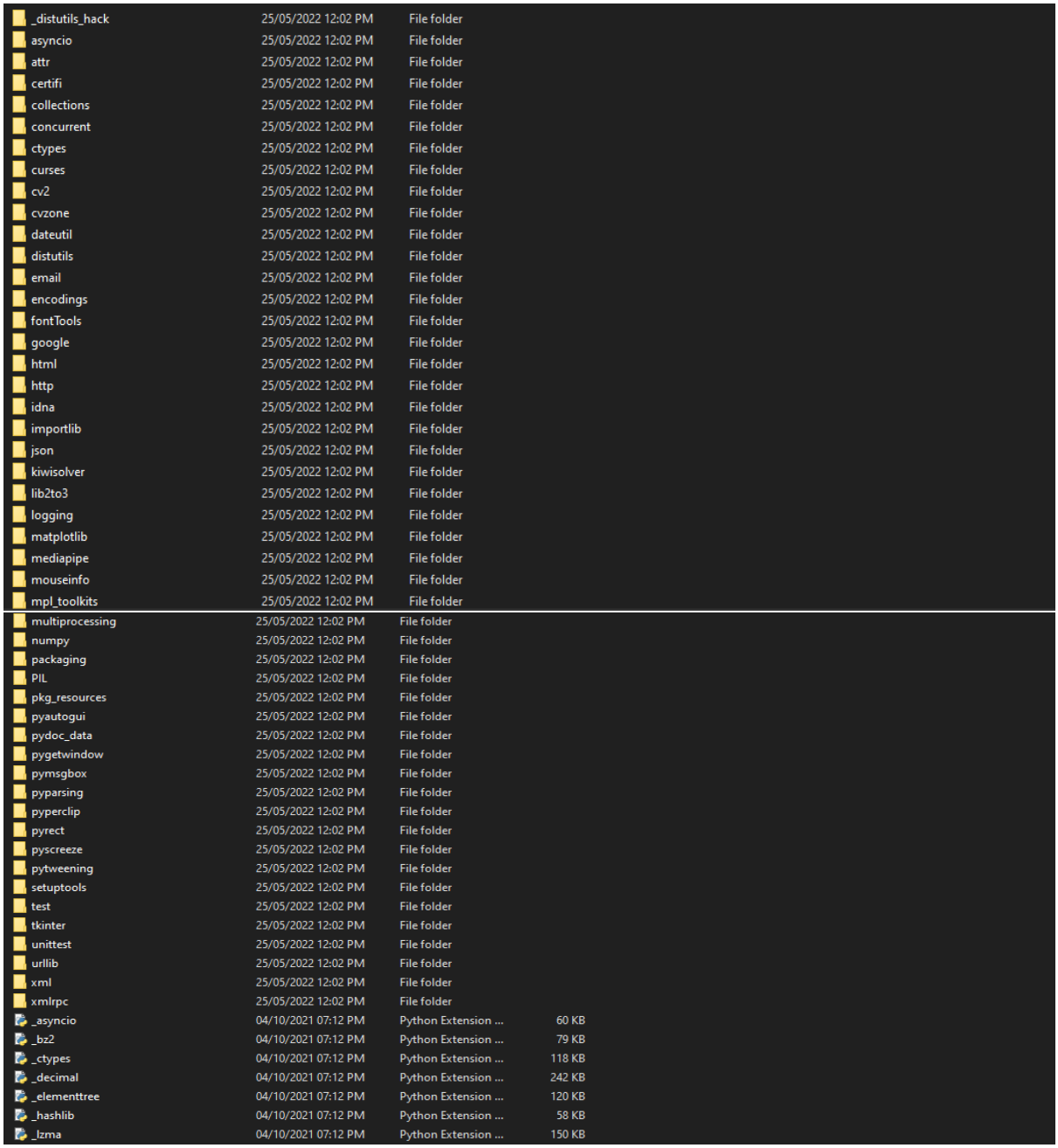


Fig 5.4 Libraries used in program

5.1 Introduction:

The algorithm for controlling the cursor by the eye iris movement was achieved through the following steps:

5.1.1 Face detection:

in order to capture the face image accurately, the user sat upright with the eye level parallel to the webcam as shown in the Fig. 5.1.1. The image of the user's face is captured using a, using which images can be captured with the help of a webcam or any other imaging tools attached to the PC. Fig. 5.1.1.4 shows the image captured by this tool

Fig. 5.1.1: System setup for eye mouse.

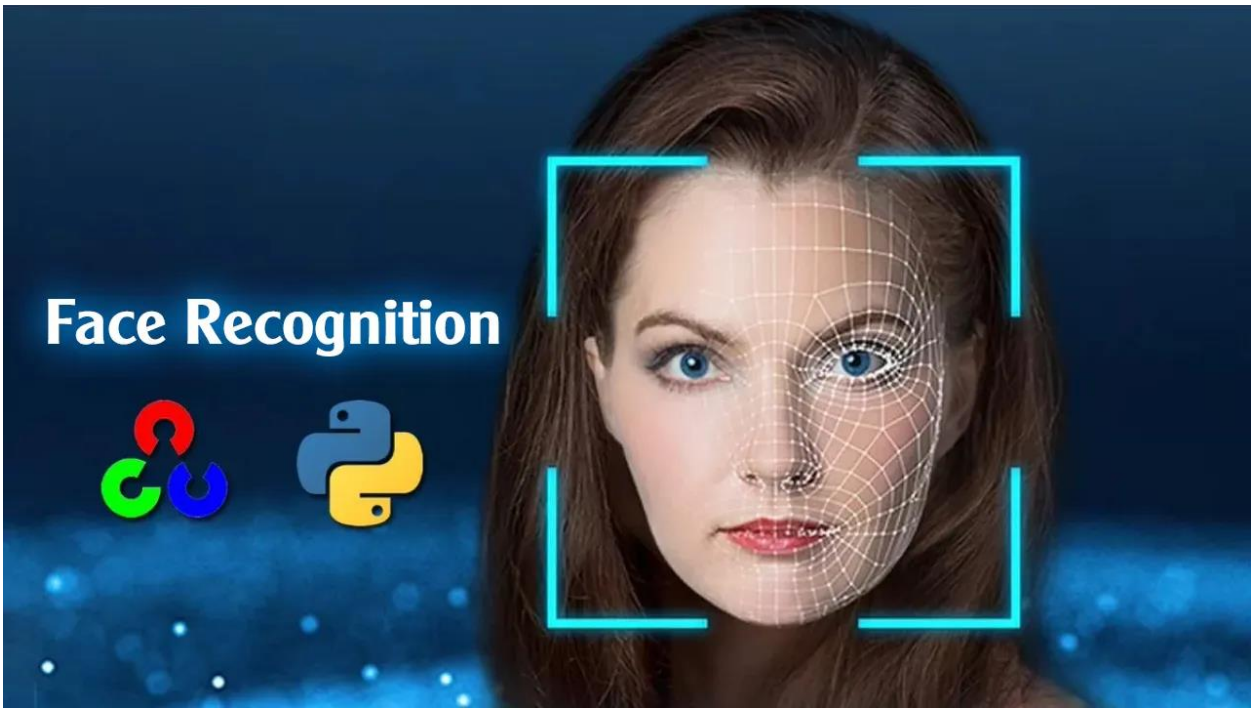


Figure 5.1.1.2 Face detection by python

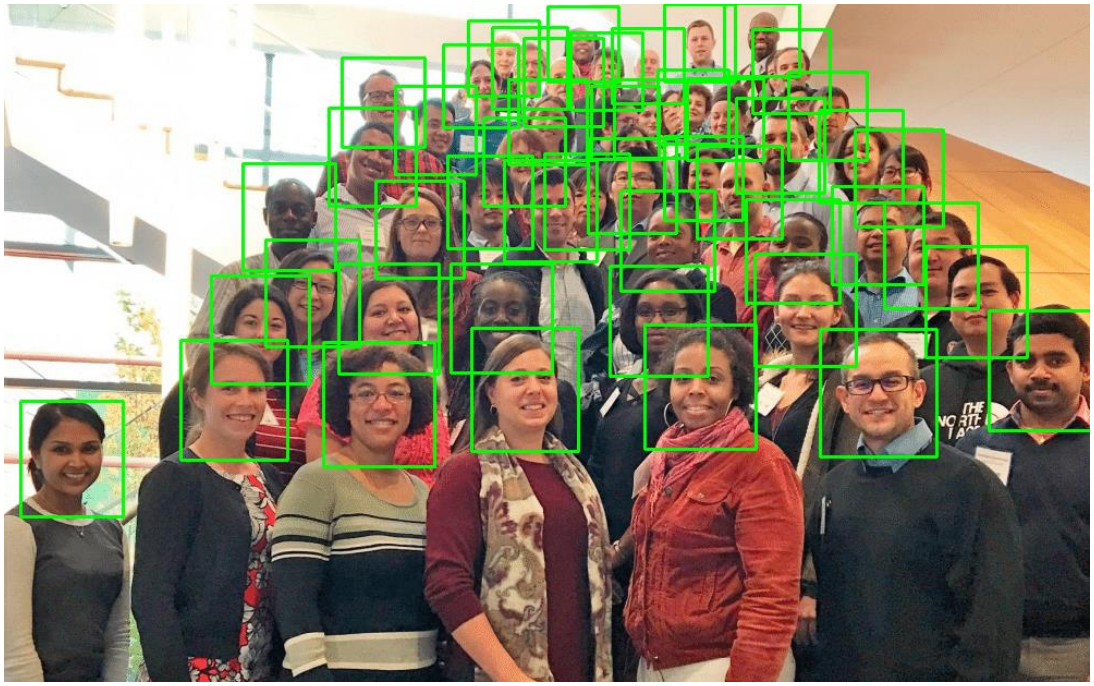


Figure 5.1.1.3 Multi face detection in python

Fig. 5.1.1.4 Image detection and image capture

5.1.2 Extracting Face mesh:

In order to extract the face mesh, by using Media pipe for face detection and recognize facial landmarks . Media Pipe Face Mesh is a solution that estimates 468 3D face landmarks in real-time even on mobile devices. It employs machine learning (ML) to infer the 3D facial surface, requiring only a single camera input without the need for a dedicated depth sensor. Utilizing lightweight model architectures together with GPU acceleration throughout the pipeline.

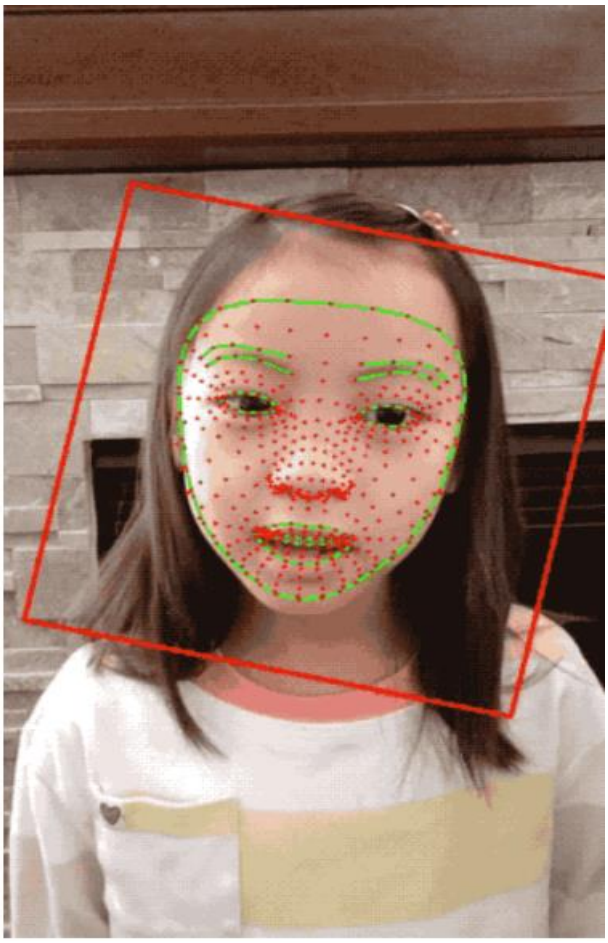


Fig. 5.1.2 *Face landmarks:*

the red box indicates the cropped area as input to the landmark model, the red dots represent the 468 landmarks in 3D, and the green lines connecting landmarks illustrate the contours around the eyes, eyebrows, lips and the entire face.

5.1.3Iris Extracting

By using Media Pipe Iris is a ML solution for accurate iris estimation, able to track landmarks involving the iris, pupil and the eye contours using a single RGB camera, in real-time, without the need for specialized hardware. Through use of iris landmarks, the solution is also able to determine the metric distance between the subject and the camera with relative error less than 10%. Note that iris tracking does not infer the location at which people are looking, nor does it provide any form of identity recognition. as fig 5.1.3 show

Fig. 5.1.3 Normalized pixels values of left eye image

5.1.4 Iris Landmark Model:

Since The iris model takes an image patch of the eye region and estimates both the eye landmarks (along the eyelid) and iris landmarks (along this iris contour).as shown in the Fig. 5.1.4

Fig. 5.1.4 *Eye landmarks (red) and iris landmarks (green).*

5.1.5 iris **pose estimation**

In computer vision the pose of an object refers to its relative orientation and position with respect to a camera. You can change the pose by either moving the object with respect to the camera, or the camera with respect to the object.

The pose estimation problem described as **Perspective-n-Point** problem or PNP in computer vision jargon in this problem the goal is to find the pose of the iris when we have a calibrated

camera, and we know the locations of **n** 3D points on the object and the corresponding 2D projections in the image.

The new approach identifying people by iris recognition. Two major contributions are suggested in our system. The first contribution was to locate the iris images by a noninvasive method. This method is based on capturing an eye image by rectifying the initial position of the user's head. This correction was calculated by the proposal of a new method that calculates head rotation angles from three interest points. This approach allows us to alert the user to rectify their position in front of the camera. Our second contribution was to cleverly use Meyer wavelets. This contribution allowed extracting an iris signature and improving the authentication results. These two contributions are detailed in the rest of the paper with the proposition of a new iris-authentication scheme.

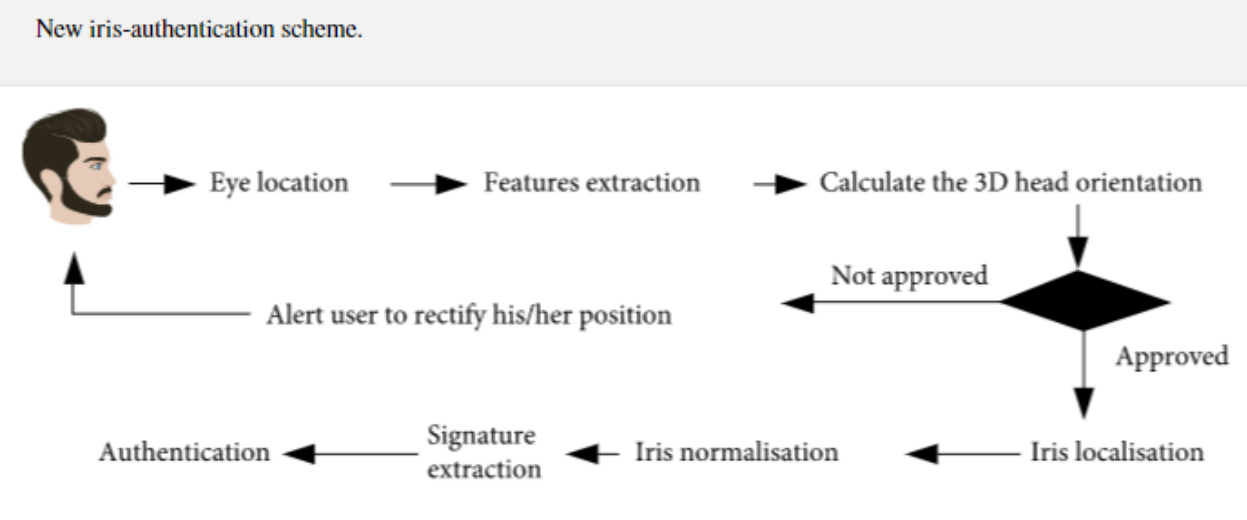


Fig .5.1.5 Steps of iris authentication scheme

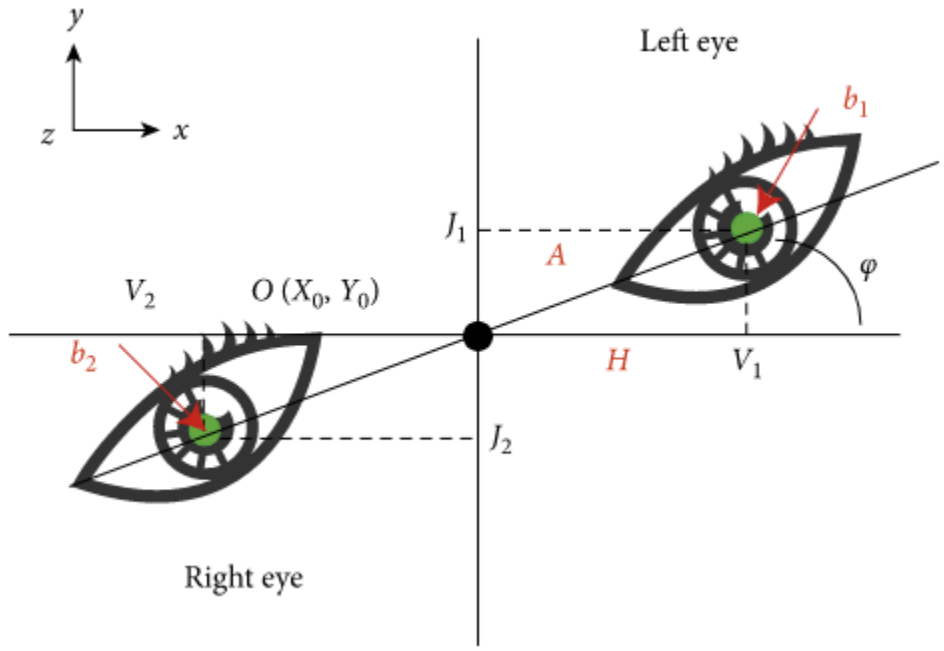


Fig 5.1.5 Calculating alpha angle

To calculate the 3D pose of an iris in an image you need the following information

1. 2D coordinates of a few points: You need the 2D (x,y) locations of points in the iris.

2. **3D locations of the same points**: You also need the 3D location of the 2D feature points.

3. Intrinsic parameters of the camera: (focal length, center, lens distortion coefficients)

In the absence of radial distortion, the coordinates (x, y) of point p in the image coordinates is given by:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = s \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

where, f_x and f_y are the focal lengths in the x and y directions, and (c_x, c_y) is the optical center.

. 5.1.6 OpenCV solvePnP

OpenCV is a huge open-source library for computer vision, machine learning and image processing.

OpenCV supports a wide variety of programming languages like python, C++, java,.etc.

In OpenCV the function **solvePnP** and **solve PnP Ransac** can be used to estimate pose.

Solve PnP implements several algorithms for pose estimation which can be selected using the parameter **flag**

Python: **cv2.solvePnP**(object Points, image Points, camera Matrix, discoes[, rvec[, tvec[, useExtrinsicGuess[, flags]]]]) → retval, rvec, tvec **Parameters:**

object Points – Array of object points in the world coordinate space. I usually pass vector of N 3D points. You can also pass Mat of size Nx3 (or 3xN) single channel matrix, or Nx1 (or 1xN) 3 channel matrix. I would highly recommend using a vector instead.

image Points – Array of corresponding image points. You should pass a vector of N 2D points. But you may also pass 2xN (or Nx2) 1-channel or 1xN (or Nx1) 2-channel Mat, where N is the number of points.

camera Matrix – Input camera matrix $A = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$. Note that f_x, f_y can be approximated by the image width in pixels under certain circumstances, and the c_x and c_y can be the coordinates of the image center.

Dist. Coeff's – Input vector of distortion coefficients (k_1, k_2, p_1, p_2[, k_3[, k_4, k_5, k_6],[s_1, s_2, s_3, s_4]]) of 4, 5, 8 or 12 elements. If the vector is NULL/empty, the zero distortion coefficients are assumed. Unless you are working with a Go-Pro like camera where the distortion is huge, we can simply set this to NULL. If you are working with a lens with high distortion, I recommend doing a full camera calibration.

rvec – Output rotation vector.

tvec – Output translation vector.

useExtrinsicGuess – Parameter used for SOLVEPNP_ITERATIVE. If true (1), the function uses the provided rvec and tvec values as initial approximations of the rotation and translation vectors, respectively, and further optimizes them.

flags –
Method for solving a PnP problem:

SOLVEPNP_ITERATIVE Iterative method is based on Levenberg-Marquardt optimization. In this case, the function finds such a pose that minimizes reprojection error, that is the sum of squared distances between the observed projections imagePoints and the projected (using projectPoints()) objectPoints .

SOLVEPNP_P3P Method is based on the paper of X.S. Gao, X.-R. Hou, J. Tang, H.-F. Chang "Complete Solution Classification for the Perspective-Three-Point Problem". In this case, the function requires exactly four object and image points.

SOLVEPNP_EPNP Method has been introduced by F.Moreno-Noguer, V.Lepetit and P.Fua in the paper "EPnP: Efficient Perspective-n-Point Camera Pose Estimation".

The flags below are only available for **OpenCV 3**

SOLVEPNP_DLS Method is based on the paper of Joel A. Hesch and Stergios I. Roumeliotis. "A Direct Least-Squares (DLS) Method for PnP".

SOLVEPNP_UPNP Method is based on the paper of A.Penate-Sanchez, J.Andrade-Cetto, F.Moreno-Noguer. "Exhaustive Linearization for Robust Camera Pose and Focal Length Estimation". In this case the function also estimates the parameters f_x and f_y assuming that both have the same value. Then the cameraMatrix is updated with the estimated focal length.

Fig 5.1.6 Opencv library

5.1.7 pyaotugui :

Pyaotugui lets your Python scripts control the mouse and keyboard to automate interactions with other applications. The API is designed to be simple. Pyaotugui works on Windows, macOS, and Linux, and runs on Python 2 and 3

PyAutoGUI has several features:

- Moving the mouse and clicking in the windows of other applications.
- Sending keystrokes to applications (for example, to fill out forms).
- Take screenshots, and given an image (for example, of a button or checkbox), and find it on the screen.
- Locate an application's window, and move, resize, maximize, minimize, or close it (Windows-only, currently).
- Display alert and message boxes.

In this project we used some functions like:

1. pyautogui.size():

Locations on your screen are referred to by X and Y Cartesian coordinates. The X coordinate starts at 0 on the left side and increases going right. Unlike in mathematics, the Y coordinate starts at 0 on the top and increases going down.

The pixel at the top-left corner is at coordinates 0, 0. If your screen's resolution is 1920 x 1080, the pixel in the lower right corner will be 1919, 1079 (since the coordinates begin at 0, not 1).

The screen resolution size is returned by the size() function as a tuple of two integers. The current X and Y coordinates of the mouse cursor are returned by the position() function.

For example:

```
>>> pyautogui.size()
(1920, 1080)
>>> pyautogui.position()
(187, 567)
```

2.pyautogui.moveTo(x, y):

The moveTo() function will move the mouse cursor to the X and Y integer coordinates you pass it. The None value can be passed for a coordinate to mean "the current mouse cursor position".

For example:

```
>>> pyautogui.moveTo(100, 200)   # moves mouse to X of 100, Y of 200.
>>> pyautogui.moveTo(None, 500)  # moves mouse to X of 100, Y of 500.
>>> pyautogui.moveTo(600, None)  # moves mouse to X of 600, Y of 500.
```

Normally the mouse cursor will instantly move to the new coordinates. If you want the mouse to gradually move to the new location, pass a third argument for the duration (in seconds) the movement should take. For example:

3. pyautogui.click():

The click() function simulates a single, left-button mouse click at the mouse's current position. A "click" is defined as pushing the button down and then releasing it up. For example:

```
>>> pyautogui.click()  # click the mouse
```

To combine a moveTo() call before the click, pass integers for the x and y keyword argument:

```
>>> pyautogui.click(x=100, y=200)  # move to 100, 200, then click the left mouse button.
```

To specify a different mouse button to click, pass 'left', 'middle', or 'right' for the button keyword argument:

```
>>> pyautogui.click(button='right')  # right-click the mouse
```

To do multiple clicks, pass an integer to the clicks keyword argument. Optionally, you can pass a float or integer to the interval keyword argument to specify the amount of pause between the clicks in seconds. For example:

```
>>> pyautogui.click(clicks=2)  # double-click the left mouse button
>>> pyautogui.click(clicks=2, interval=0.25)  # double-click the left mouse button, but with a quarter second pause in between clicks
>>> pyautogui.click(button='right', clicks=3, interval=0.25)  ## triple-click the right mouse button with a quarter second pause in between clicks
```

As a convenient shortcut, the doubleClick() function will perform a double click of the left mouse button. It also has the optional x, y, interval, and button keyword arguments. For example:

```
>>> pyautogui.doubleClick()  # perform a left-button double click
```

There is also a tripleClick() function with similar optional keyword arguments.

The rightClick() function has optional x and y keyword arguments.

4. pyautogui.scroll(10) :

The mouse scroll wheel can be simulated by calling the scroll() function and passing an integer number of "clicks" to scroll. The amount of scrolling in a "click" varies between platforms. Optionally, integers can be passed for the the x and y keyword arguments to move the mouse cursor before performing the scroll. For example:

```
>>> pyautogui.scroll(10)   # scroll up 10 "clicks"
>>> pyautogui.scroll(-10)  # scroll down 10 "clicks"
>>> pyautogui.scroll(10, x=100, y=100)  # move mouse cursor to 100, 200, then scroll up 10 "clicks"
```

On OS X and Linux platforms, PyAutoGUI can also perform horizontal scrolling by calling the hscroll() function. For example:

```
>>> pyautogui.hscroll(10)   # scroll right 10 "clicks"
>>> pyautogui.hscroll(-10)   # scroll left 10 "clicks"
```

The scroll() function is a wrapper for vscroll(), which performs vertical scrolling.

5. the write() function:

The primary keyboard function is write(). This function will type the characters in the string that is passed. To add
a delay interval in between pressing each character key, pass an int or float for the interval keyword argument.
For example:

```
>>> pyautogui.write('Hello world!') # prints out "Hello world!"
‹→instantly
>>> pyautogui.write('Hello world!', interval=0.25) # prints out "Hello world!" with
‹→a quarter second delay after each character
```

You can only press single-character keys with write(), so you can't press the Shift or F1 keys, for example.

6. press(),keydown() and keyuP():

To press these keys, call the press() function and pass it a string from the pyautogui.KEYBOARD_KEYS such
as enter, esc, f1. See KEYBOARD_KEYS.

For example:

```
>>> pyautogui.press('enter') # press the Enter key
>>> pyautogui.press('f1') # press the F1 key
>>> pyautogui.press('left') # press the left arrow key
```

The press() function is really just a wrapper for the keyDown() and keyUp() functions, which simulate pressing.

a key down and then releasing it up. These functions can be called by themselves. For example, to press the left arrow.

key three times while holding down the Shift key, call the following:

```
>>> pyautogui.keyDown('shift') # hold down the shift key
>>> pyautogui.press('left') # press the left arrow key
>>> pyautogui.press('left') # press the left arrow key
>>> pyautogui.press('left') # press the left arrow key
>>> pyautogui.keyUp('shift') # release the shift key
```

To press multiple keys similar to what write() does, pass a list of strings to press(). For example:

```
>>> pyautogui.press(['left', 'left', 'left'])
```

Or you can set how many presses left:

```
>>> pyautogui.press('left', presses=3)
```

To add a delay interval in between each press, pass an int or float for the interval keyword argument.

7. The hold() context manager

To make holding a key convenient, the hold() function can be used as a context manager and passed a string from
the pyautogui.KEYBOARD_KEYS such as shift, ctrl, alt, and this key will be held for the duration of the
with context block. See KEYBOARD_KEYS.

```
>>> with pyautogui.hold('shift'):
pyautogui.press(['left', 'left', 'left'])
. . . is equivalent to this code:
>>> pyautogui.keyDown('shift') # hold down the shift key
>>> pyautogui.press('left') # press the left arrow key
>>> pyautogui.press('left') # press the left arrow key
>>> pyautogui.press('left') # press the left arrow key
>>> pyautogui.keyUp('shift') # release the shift key
```

8. The hotkey() function

To make pressing hotkeys or keyboard shortcuts convenient, the hotkey() can be passed several key strings which
will be pressed down in order, and then released in reverse order. This code:

```
>>> pyautogui.hotkey('ctrl', 'shift', 'esc')
. . . is equivalent to this code:
>>> pyautogui.keyDown('ctrl')
>>> pyautogui.keyDown('shift')
>>> pyautogui.keyDown('esc')
>>> pyautogui.keyUp('esc')
>>> pyautogui.keyUp('shift')
>>> pyautogui.keyUp('ctrl')
```

To add a delay interval in between each press, pass an int or float for the interval keyword argument.

5.1.8 CVzone

cvzone makes it easy to run image processing and AI functions and for requirement we need to install opencv and mediapipe as we declare above how to install each other.

Used in face detection, hand tracking and pose estimation and that exactly we need in our software project to reach the final form of the application which made user use the application as we planned without any obstacles

To install it for windows, use pip install cvzone and again we need first before install cvzone that mediapipe and opencv packages must be in tour environment.

Fig 5.1.8 CVZone library logo

5.1.9 Blink detecting :

Fig 5.1.8 blinking action

By using matplotlib library we import live plot graph to explain the occurrence of blinking

At Fig 5.1.9.1 Top-left: A visualization of eye landmarks when then the eye is open. Top-right: Eye landmarks when the eye is closed. Bottom: Plotting the eye aspect ratio over time. The dip in the eye aspect ratio indicates a blink

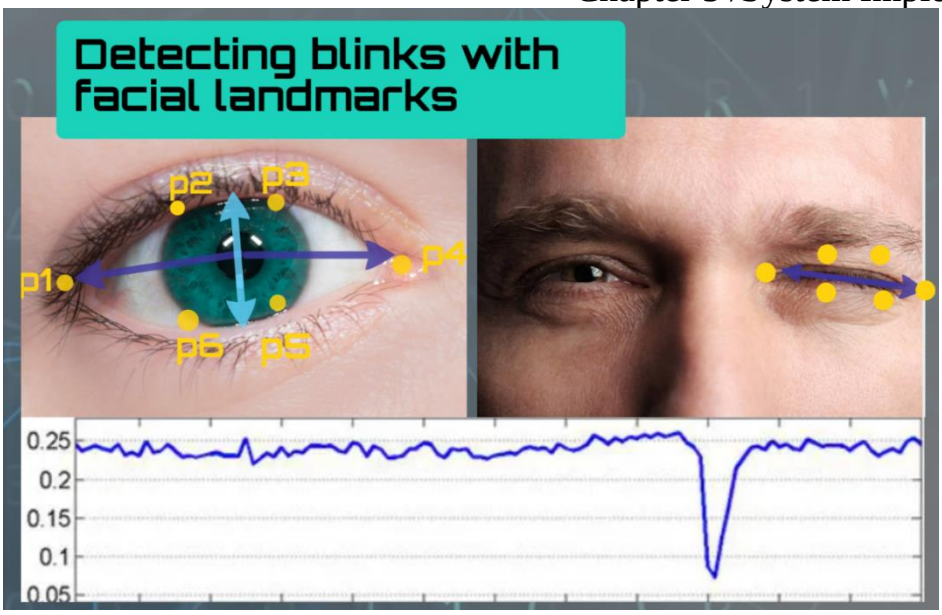


Fig 5.1.9.1 detecting blinks

Chapter 6

Conclusion & Future Work

6.1 Conclusion:

A system that enables a disabled person to interact with the computer was successfully developed and tested. The method can be further enhanced to be used in many other applications. The system can be adapted to help the disabled to control home appliances such as TV sets, lights, doors etc. The system can also be adapted to be used by individuals suffering from complete paralysis, to operate and control a wheelchair. The eye mouse can also be used to detect drowsiness of drivers in order to prevent vehicle accidents. The eye movement detection and tracking have also potential use in gaming and virtual reality.

Finally, we note that the project is operable in variable environmental conditions, only a few tweaks in the brightness and contrast setting need to be applied for it to maintain its robustness. This is an impressive feat for such a low-cost eye-tracking system.

6.2 Future Work:

It's never too early or too late when we think about improving the project so in this section we discuss what will happened to the project through the maintenance, upgrade and marketing.

First, I want to make you sure that this software program will be better in the future with the cooperation of team members so every member should improve from his\her skill in problem solving and coding so we can make a project we proud of it in the future and that what exactly we do right now to make user more comfortable and to be good experience to him\her.

Firstly seek in the near future to help people with special needs who do not have the ability to move and walk, that is, people who have muscle atrophy, to control the wheelchair through the movement of their eyes, by making our program as an embedded system, which is by using a tablet installed in the wheelchair facing the face of the sick person and about Through the movements of the iris of the eye, moving the wheelchair, which is moving the eye to the right or left, moving the eye up to the front movement, and moving the eye down to go back.

Dr. Stephen Hawking was diagnosed with Amyotrophic Lateral Sclerosis، commonly referred to in the U.S. as Lou Gehrig's disease. As ALS progresses, the degeneration of motor neurons in the brain interferes with messages to muscles in the body. Eventually, muscles atrophy and voluntary control of muscles is lost. As fig 6.2.1 shows

FIG 6.2.1 Steven hawking

Secondly , Using our software in the metaverse and the virtual world where a disabled person can move using the metaverse through their avatar and make purchases and sales as if they were in the real world. As fig 6.2.2shows :

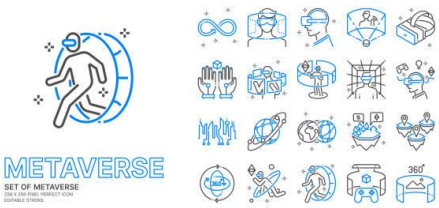
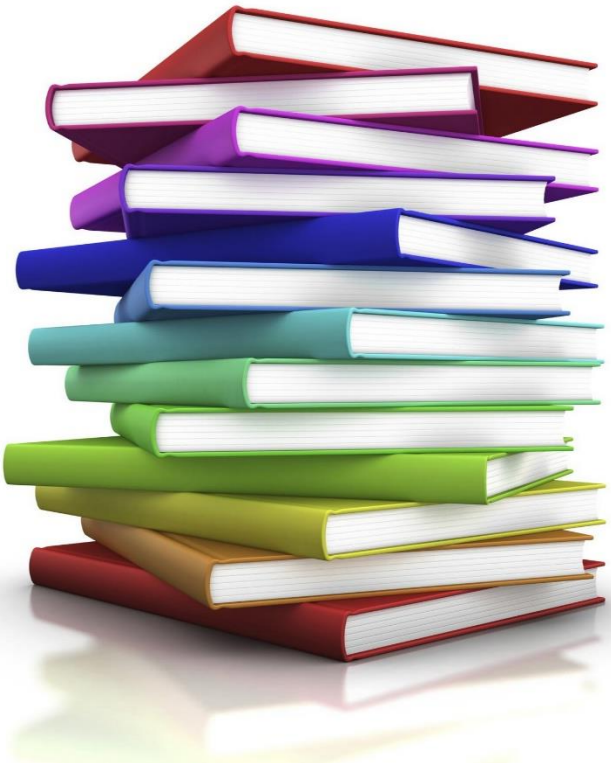
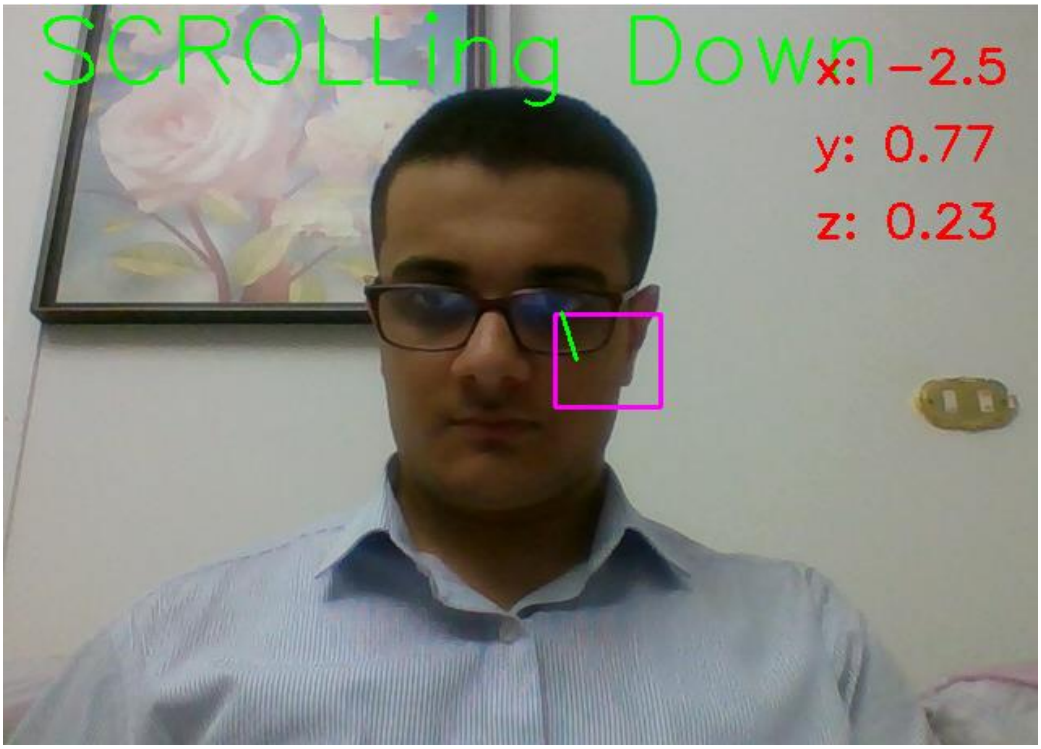


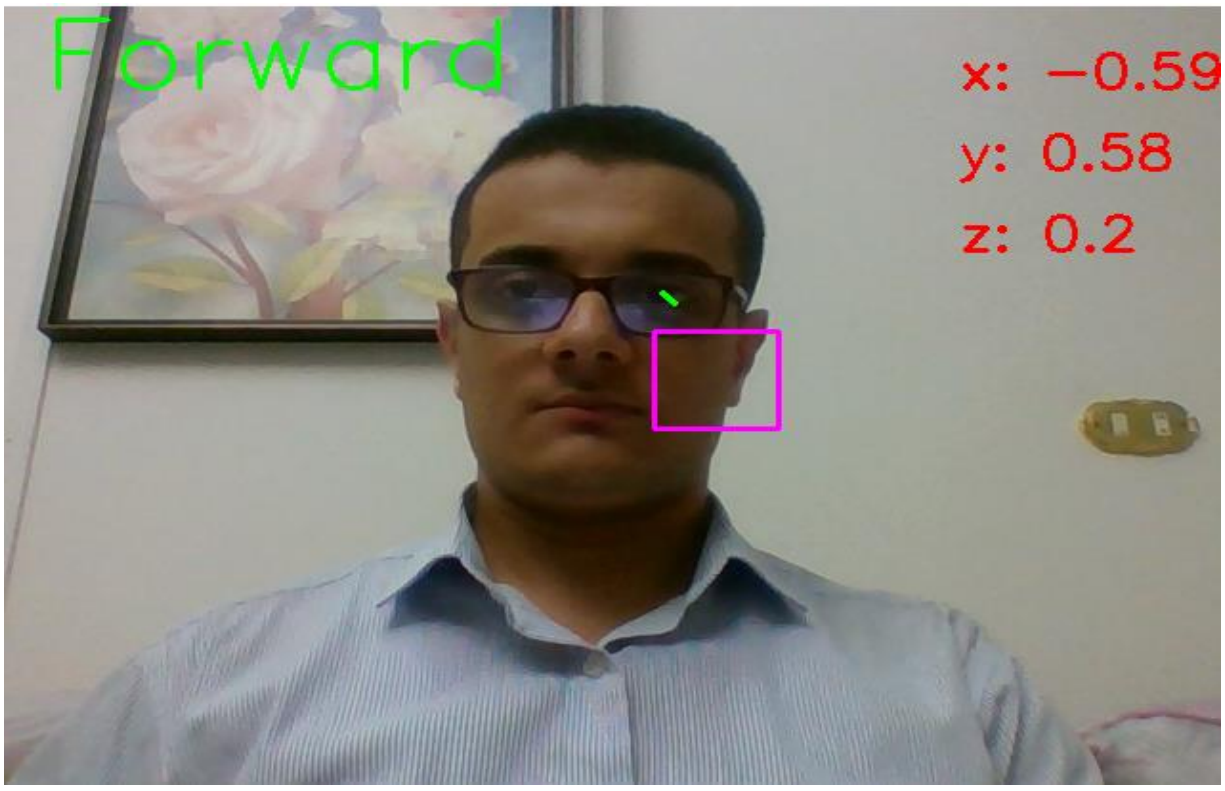
Fig 6.2.2

Appendix



1- Program screenshots:




Forward
x: −0.59
y: 0.58
z: 0.2

2- Codes used in the program:

```
import cv2
import mediapipe as mp
import numpy as np
import time
import pyautogui
from cvzone.FaceMeshModule import FaceMeshDetector
from cvzone.PlotModule import LivePlot
import cvzone
from time import sleep

############################ blink
detector = FaceMeshDetector(maxFaces=1)
plotY = LivePlot(640, 360, [20, 50], invert=True)
############################ left blink
LI = [22, 23, 24, 26, 110, 157, 158, 159, 160, 161, 130, 243]
LIratio = []

############################ blink

############################ right blink
RI = [253, 252, 256, 254, 339, 257, 258, 286, 259, 260, 359, 463]
RIratio = []
############################ blink
# # # # # # # # # # mouse
MOUSE = [11, 72, 73, 302, 303, 16, 85, 180, 315, 404, 61, 291]
Mratio = []

# # # # # # # # # # mouse
##########################

##########################
wCam, hCam = 640, 360
frameX = 340
frameY = 200
smoothening = 7
##########################

plocX, plocY = 0, 0
clocX, clocY = 0, 0


pyautogui.FAILSAFE = False
wscreen, hscreen = pyautogui.size()


mp_face_mesh = mp.solutions.face_mesh
face_mesh = mp_face_mesh.FaceMesh(refine_landmarks=True,
min_detection_confidence=0.5, min_tracking_confidence=0.5)
```

```
cap = cv2.VideoCapture(0)


# Read logo and resize
logo = cv2.imread('image.png')
size = 100
logo = cv2.resize(logo, (size, size))
# Create a mask of logo
img2gray = cv2.cvtColor(logo, cv2.COLOR_BGR2GRAY)
ret, mask = cv2.threshold(img2gray, 1, 255, cv2.THRESH_BINARY)

while cap.isOpened():
    # blink
    # Region of Image (ROI), where we want to insert logo


    if cap.get(cv2.CAP_PROP_POS_FRAMES) == cap.get(cv2.CAP_PROP_FRAME_COUNT):
        cap.set(cv2.CAP_PROP_POS_FRAMES, 0)
        # blink
    success, img = cap.read()

    img = cv2.flip(img, 1)
    img, faces = detector.findFaceMesh(img, draw=False)
    roi = img [-size - 10:-10, -size - 10:-10]
    roi[np.where(mask)] = 0
    roi += logo

    results = face_mesh.process(img)

    img_h, img_w, img_c = img.shape
    face_3d = []
    face_2d = []
    # 473,467
    # blink
    if results.multi_face_landmarks:
        for face_landmarks in results.multi_face_landmarks:
            for idx, lm in enumerate(face_landmarks.landmark):

                if idx == 473 or idx == 468 or idx == 168 or idx == 6 or idx
== 291 or idx == 199:
                    if idx == 473:
                        IRIS_2d = (lm.x * img_w, lm.y * img_h)
                        IRIS_3d = (lm.x * img_w, lm.y * img_h, lm.z * 3000)

                    x, y = int(lm.x * img_w), int(lm.y * img_h)

                    face_2d.append([x, y])

                    face_3d.append([x, y, lm.z])

            face_2d = np.array(face_2d, dtype=np.float64)
```

```
            face_3d = np.array(face_3d, dtype=np.float64)

            focal_length = 1 * img_w

            cam_matrix = np.array([[focal_length, 0, img_h / 2],
                                   [0, focal_length, img_w / 2],
                                   [0, 0, 1]])

            dist_matrix = np.zeros((4, 1), dtype=np.float64)
            success, rot_vec, trans_vec = cv2.solvePnP(face_3d, face_2d,
cam_matrix, dist_matrix)

            rmat, jac = cv2.Rodrigues(rot_vec)

            angles, mtxR, mtxQ, Qx, Qy, Qz = cv2.RQDecomp3x3(rmat)

            x = angles[0] * 90
            y = angles[1] * 90
            z = angles[2] * 90

            if y < -10:
                text = "Looking Left"
            elif y > 10:
                text = "Looking Right"
            elif x < -1.5:
                text = "SCROLLing Down"
                pyautogui.scroll(-200)
            elif x > 0.6:
                text = "SCROLLing Up"
                pyautogui.scroll(200)
            else:
                text = "Forward"


            iris_3d_projection, jacobian = cv2.projectPoints(IRIS_3d,
rot_vec, trans_vec, cam_matrix, dist_matrix)

            p1 = (int(IRIS_2d[0]), int(IRIS_2d[1]))
            x2 = int(IRIS_2d[0])
            y2 = int(IRIS_2d[1])

            p2 = (int(IRIS_2d[0] + y * 12), int(IRIS_2d[1] - x * 12))
            x1 = int(IRIS_2d[0] + y * 12)
            y1 = int(IRIS_2d[1] - x * 12)
            cv2.rectangle(img, (frameX, frameY), (wCam - 235, hCam - 100),
                          (0, 255, 0), 2)

            x3 = np.interp(x1, (frameX, wCam - 235), (0, wscreen))
            y3 = np.interp(y1, (frameY, hCam - 100), (0, hscreen)
```

```
)

            cv2.line(img, p1, p2, (0, 0, 255), 2)

            clocX = plocX + (x3 - plocX) / smoothening
            clocY = plocY + (y3 - plocY) / smoothening

            pyautogui.moveTo(clocX, clocY)

            plocX, plocY = clocX, clocY

            cv2.putText(img, text, (20, 50), cv2.FONT_HERSHEY_SIMPLEX, 2, (0,
255, 0), 2)
            cv2.putText(img, "x: " + str(np.round(x, 2)), (500, 50),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)
            cv2.putText(img, "y: " + str(np.round(y, 2)), (500, 100),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)
            cv2.putText(img, "z: " + str(np.round(z, 2)), (500, 150),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)


    if faces:
        face = faces[0]
        ritop = face[257]
        ridown = face[253]
        rileft = face[463]
        riright = face[359]
        lenghtVer1, _ = detector.findDistance(ritop, ridown)
        lenghtHor1, _ = detector.findDistance(rileft, riright)

        ratio1 = int((lenghtVer1 / lenghtHor1) * 100)
        RIratio.append(ratio1)
        if len(RIratio) > 3:
            RIratio.pop(0)
        ratioAvg1 = sum(RIratio) / len(RIratio)

        if ratioAvg1 < 35:
            pyautogui.click(button='right')


        # # # # # # # # # # right
        litop = face[159]
        lidown = face[23]
        lileft = face[130]
        liright = face[243]
        lenghtVer, _ = detector.findDistance(litop, lidown)
        lenghtHor, _ = detector.findDistance(lileft, liright)
        ratio = int((lenghtVer / lenghtHor) * 100)
```

```
        LIratio.append(ratio)
        if len(LIratio) > 3:
            LIratio.pop(0)
        ratioAvg = sum(LIratio) / len(LIratio)

        if ratioAvg < 35:
            pyautogui.click(button='left')
        # # # # # # # # # # right
        # # # # # # # # # # mouse

        mtop = face[11]
        mdown = face[16]
        mleft= face[61]
        mright = face[291]
        mlenghtVer, _ = detector.findDistance(mtop, mdown)
        mlenghtHor, _ = detector.findDistance(mleft, mright)

        ratio2 = int((mlenghtVer / mlenghtHor) * 100)
        Mratio.append(ratio2)
        if len(Mratio) > 3:
            Mratio.pop(0)
        ratioAvg2 = sum(Mratio) / len(Mratio)

        if ratioAvg2 > 45:
             pyautogui.click(button='right')


    cv2.imshow('AI-EYE', img)

    if cv2.waitKey(5) & 0xFF == 27:
        break

cap.release()
```

• References

- [1] B. Rebsamen, C. L. Teo, Q. Zeng, M. Ang. Jr. "Controlling a wheelchair indoors using thought" IEEE Intelligent Systems, 2007, pp. 18-24.
- [2] C. A. Chin "Enhanced Hybrid Electromyogram / Eye gaze tracking cursor control system for hands-free computer interaction", Proceedings of the 28th IEEE EMBS Annual International Conference, New York City, USA, Aug 30-Sept 3, 2006, pp. 2296-2299.
- [3] J. Kirke's, J. Riana, J. Bergman's, "Using an Eye tracker for Accurate Eye Movement Artifact Correction", IEEE Transactions on Biomedical Engineering, vol. 54, no. 7, July 2007, pp. 1257-1267.
- [4] A. E. Kaufman, A. Bandyopadhyay, B. D. Shavit, "An Eye Tracking Computer User Interface", Research Frontier in Virtual Reality Workshop Proceedings, IEEE Computer Society Press, October 1993, pp. 78-84.
- [5] T. Kocejko, "Device which will allow people suffered from Lateral Amyotrophic Sclerosis to communicate with the environment", MSc thesis, January 2008. G. A. Myers, K. R. Sherman, L. Stark, "Eye Monitor", IEEE Computer Magazine, Vol. March 1991, pp. 14-21.
- [6] C. Collet, A. Finkel, R. Gharib, "A Gaze Tracking System in Man-Machine Interaction", Proceedings of IEEE International Conference on Intelligent Engineering Systems, September 1997.
- [7] B. Hu, M. Qiu, "A New Method for Human-Computer Interaction by using Eye-Gaze", Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, October 1994.
- [8] P. Ballard, G. C. Stockman, "Computer operation via Face Orientation", Pattern Recognition vol. 1. Conference A: Computer Vision and Applications, Proceedings., 11th IAPR International Conference, 1992.
- [9] https://www.mathworks.com/matlabcentral/fileexchange/ 247-vfm
- [10] https://google.github.io/mediapipe/getting_started/python.html
- [11] https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html
- [12] https://pythonlang.dev/repo/cvzone-cvzone/
- [13]https://www.lucidchart.com/pages/landing/uml-diagram-software?utm_source=google&utm_medium=cpc&utm_campaign=_chart_en_tier3_desktop_search_nb_exact_&km_CPC_CampaignId=2083826535&km_CPC_AdGroupID=76733736197&km_CPC_Keyword=uml%20diagram&km_CPC_MatchType=e&km_CPC_ExtensionID=&km_CPC_Network=g&km_CPC_AdPosition=&km_CPC_Creative=442345716870&km_CPC_TargetID=aud-552508845282:kwd-10399741&km_CPC_Country=1005393&km_CPC_Device=c&km_CPC_placement=&km_CPC_target=&gclid=CjwKCAjw5NqVBhAjEiwAeCa97ck7uiTcqAG1akctkQPKrKqgThwi7yqo5zK_u54u_7RUkR1XhMKZphoCh4YQAvD_BwE

- [14] https://www.topcoder.com/thrive/articles/python-for-gui-automation-pyautogui
- [15] https://docs.opencv.org/4.x/db/d05/tutorial_config_reference.html
- [16] https://pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/
- [17]https://www.neurotechnology.com/sentiveillance.html?gclid=CjwKCAjw5NqVBhAjEiwAeCa97da2xOHDT4VWuhCFnHQj_PIwYJQPI7NHLj3AtIlQecMBoFQTUwfaLBoCyswQAvD_BwE
- [18]https://www.megamatcher.online/?gclid=CjwKCAjw5NqVBhAjEiwAeCa97SKY3PRYCTHQ58D6plN5ONJdN8a8JW-MTo1uj76W8FmaMgZc3m4vjxoC5RYQAvD_BwE

Summary

هو برنامج الاستبدال الماوس بتحريك المؤشر عن طريق حركتي العين والرأس للإنسان ويقوم بجميع وظائف الماوس

مثل (التحريك العلى والسفل, فتح مجلد وإغلاقه, السحب والفالت الملف او المجلد, والنقر مرة والنقر وتطبيق هذا البرنامج على جهاز كمبيوتر مزود بكاميرا ويب الموجودة بالجهاز ال يلزم أجهزة)مرتين...الخ

إضافية للمستخدمين لذوي الاحتياجات الخاصة ومن ال يملكون القدرة على تحريك اليدين، ويمكن استخدام هذا البرنامج للإنسان العادي لزيادة الرفاهية بدون الحاجة الى المساك بالماوس ويمكن تطبيق هذا البرنامج على الهواتف والتابلت الذكية