INNOVATIVE PROJECT REPORT

A report submitted in partial fulfillment of the requirements for the Award of Degree of

Bachelor of Technology
in
Computer Science and Information Technology (Cyber Security)

By

Name of Student: Yeshu Wanjari
PRN: 2001106045

Under Supervision of
Project Guides:
Prof.Asheesh Dixit
Prof. Ashvini Bhongade

SCHOOL OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

Symbiosis Skills & Professional University
Kiwale, Adjoining Pune Mumbai Expressway, Pune - 412 101

SCHOOL OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

Symbiosis Skills & Professional University

Kiwale, Adjoining Pune Mumbai Expressway, Pune - 412 101

CERTIFICATE

This is to certify that the Innovative Project Report submitted by Yeshu Wanjari , **PRN:** 20011066045 is done by him and submitted during 2021 to 2022 academic year, in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Information Technology (Cyber Security).**

Project Guide

Director
CSIT

ACKNOWLEDGEMENT

The Innovative Project opportunity I had was a great chance for learning and professional development. Therefore, I consider myself as a very lucky individual as I was provided with an opportunity to be a part of it.

I would like to use this opportunity to express my deepest gratitude to **Prof. Asheesh Dixit, Dir CSIT** who whole heartedly welcomed me for the innovative project and guided and encouraged me through the internship period.

I would also like to convey my heartiest thanks to **Prof.Ashvini Bhongade,** who helped us during the innovative project period and taught us some new methodologies and applications.

Last but not least, I would like to thank all the department heads and the staff for their careful and precious guidance and creating a friendly environment while working which were extremely valuable for my study both theoretically and practically.

Table of Contents

- # Preliminary Investigation

An Eye Controlled mouse meaning the cursor can be controlled by the movment of the users iris and right click can be made by a blink of an eye.

To be very Honest I got this idea back in my 10th standard and I even researched for various apks to make it possible on a android device. I even got apps which worked on rooted device, I found it fun at that time.

Now, To make it possible on Windows Platform I knew python is simple to work on and still had huge possibilities, I researched for Python libraries to to control system utilities like mouse and keyboard via code input, later I searched for libraries to recognise and trace face and hands. I learned via youtube to interconnect these libraries

There are different reasons for which people need an artificial of locomotion such as a virtual keyboard. The number of people, who need to move around with the help of some article means, because of an illness. Moreover, implementing a controlling system in it enables them to move without the help of another person is very helpful.
Hands-free computing is popular nowadays because it caters to the needs of quadriplegics (people suffering from paralysis of all four limbs).

The software requirements for this technique include Python, OpenCV, NumPy, and a few more facial recognition tools, as well as a basic camera.

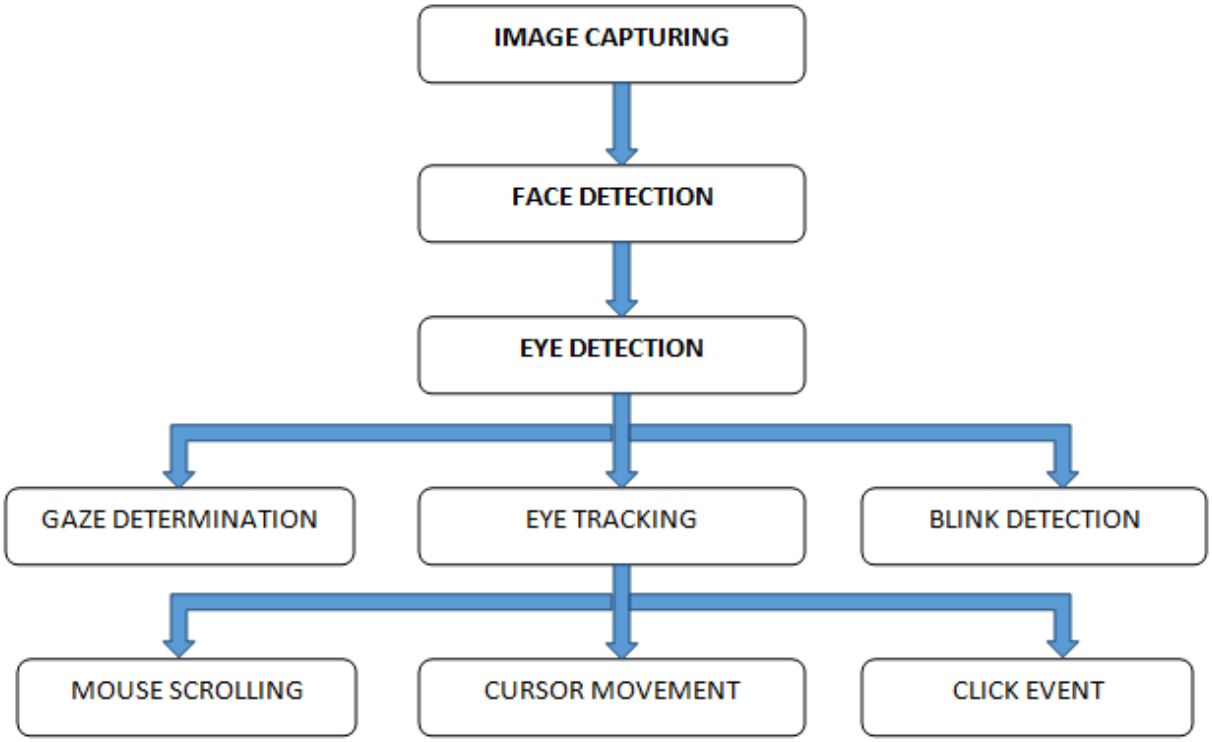
• Company Profile

The user has to sits in front of the display screen of private computer or pc, a specialised video camera established above the screen to study the consumer's eyes. The laptop constantly analysis the video photo of the attention and determines wherein the consumer is calling at the display screen. not anything is attached to the consumer's head or body. To "pick out" any key, the user seems at the key for a exact period of time and to "press" any key, the consumer just blink the eye. On this device, calibration procedure is not required. For this system enter is simplest eye. No outside hardware is connected or required.

Camera gets the input from the eye. After receiving these streaming movies from the cameras, it'll spoil into frames. After receiving frames, it will check for lights conditions because cameras require enough lighting fixtures from external sources in any other case blunders message will show at the screen. The captured frames which can be already in RGB mode are transformed into Black 'n' White. Five. Pics (frames) from the enter supply focusing the eye are analysed for Iris detection (middle of eye).After this, a mid point is calculated through taking the suggest of left and right eye centre point. Eventually the mouse will pass from oneposition to any other at the display and consumer will perform clicking with the aid of blinking their eyes for 5 seconds.

An 'eye-tracking mouse' is a gadget that tracks the user's eye movements. The project relies on mapping facial traits to the cursor to recognize and capture them in video. When the camera is opened, the application must extract all of the video's frames.

Diagramatical representation of the Cursor Movement Process :



• Manual System

No doubt the user can manually operate a normal mouse effectively. But the Eye Controlled mouse has its use cases in special conditions.

There will always be limitations of the mouse as the mouse is a hardware input the mouse is a hardware device like any other physical object even the mouse will have a durability time within which is functional and after its durability time we have to change the mouse

A Specially abled person cant use the conventional method efficiently or not at all The number of people, who need to move around with the help of some article means, because of an illness. Moreover, implementing a controlling system in it enables them to move without the help of another person is very helpful. The idea of eye controls of great use to not only the future of natural input but more importantly the handicapped and disabled.

• Innovative idea behind proposed system

The idea was to find a fun way to fully operate the computer system without even touching the system. The project is a mouse-like eye-based interface that converts eye movements like blinking, staring, and squinting into mouse cursor actions. As Eye controlled mouse has been an idea before so I have also modified the project for mouse to be controlled by hand gestures

An 'eye-tracking mouse' is a gadget that tracks the user's eye movements. The project relies on mapping facial traits to the cursor to recognize and capture them in video. When the camera is opened, the application must extract all of the video's frames.

Since the video's frame rate is typically around 30 frames per second, every frame will be processed in about 1/30th of a second. The application then goes through a series of steps to identify and map the characteristics of the video to the point .After the frame has been retrieved, the face areas must be identified. As a result, the frames will go through a set of image-processing routines to appropriately analyse the frame, allowing the algorithm to distinguish things like eyes, mouths, and noses.

• Proposed System

An Eye Controlled mouse meaning the cursor can be controlled by the movement of the users iris and right click can be made by a blink of an eye.

1. With the help of web camera, the video is said to be recorded

2. The recorded video is converted into Frames

3. From the frames it is further converted into Grayscale for the background elimination

4. After the elimination of background it takes a proper face image to find Counter and Edges in

5. From edges and counters it Identifies Eye and Mouth in the Frame

6. After identifying we calculate Aspect Ratio of Eye and Mouth

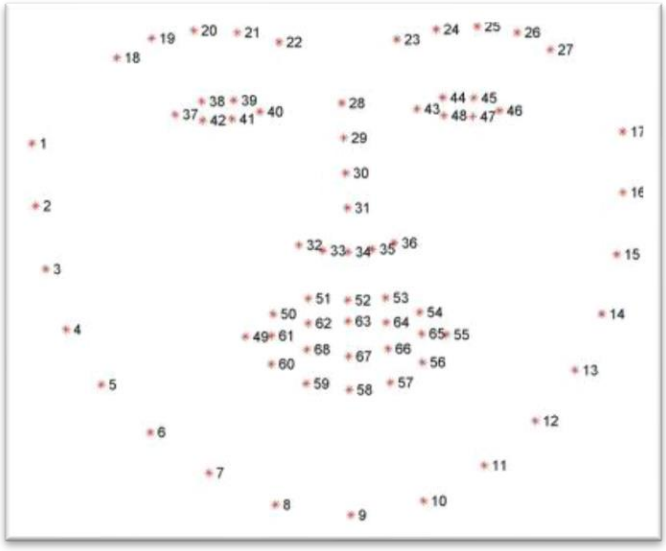
7. Eye Blink and Head Moment is Detected through Decision Algorithm

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

Dlib's prebuilt model, not only does a fast face-detection but also allows us to accurately predict 68 2D facial landmarks. Very handy.

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

programmed as triggers to control the mouse cursor. PyAutoGUI library was used to move the cursor around.



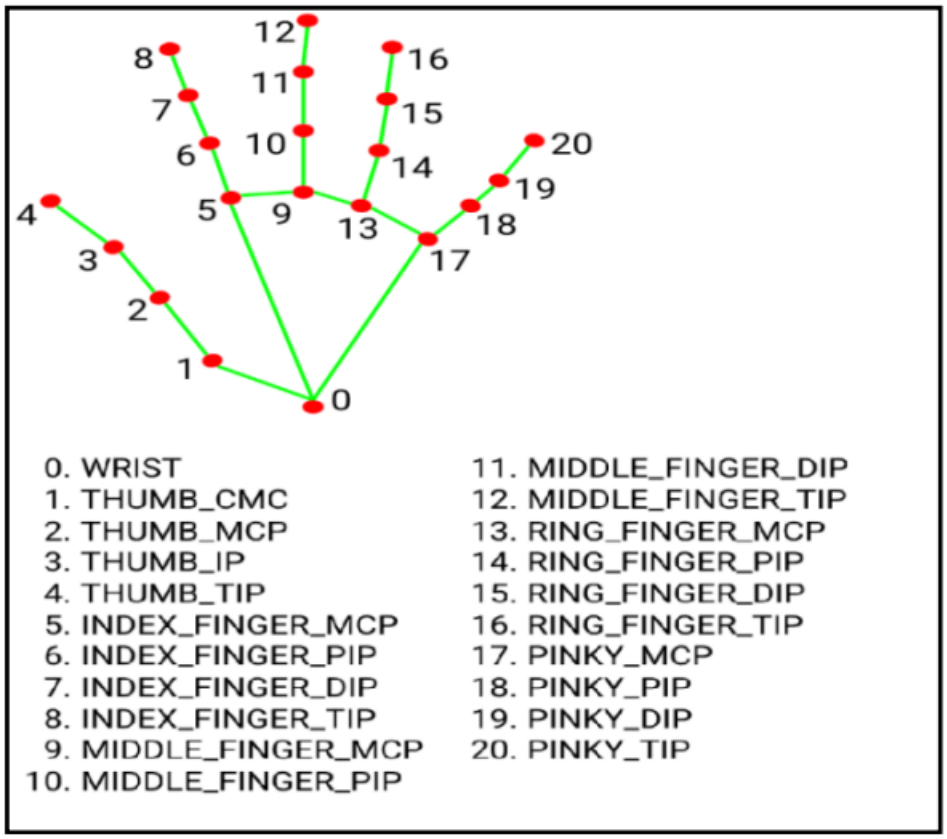
After detecting the face, eye region is detected with the help of facial landmark features. Using the face landmarks dataset, we can point out 68 landmarks on the face. each landmark is

assigned with an index. Using these indices, the desired region of the face is detected, . Point index for two eyes:

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

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

Hand landmarks by Mediapipe:

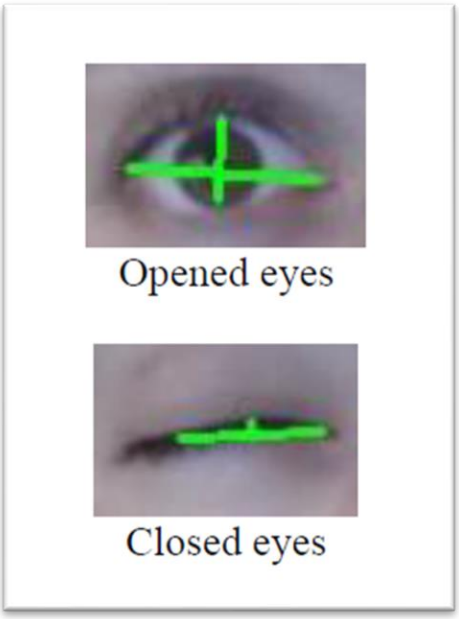


EYE BLINK DETECTION

With the exact eye region, we can detect the blinks with the help of two lines. One drawn horizontally and other drawn vertically splitting the eyes.Temporary closure of eyes along with the movement of eyelids is known as blink. It is a rapid natural process. We have to find out what happens when eye is blinked. We can conclude that the eye is closed/blinked when:

- Eyeball is not visible
- Eye lid is closed

- Upper and lower eyelids are connected together



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

LIBRARIES USED IN THIS PROJECT:

Numpy 1.13.3 : The NumPy is a multidimensional array used to store values of same datatype. These arrays are indexed just like Sequences, starts with zero.

Pyautogui 0.9.36 : mainly used to operate mouse events imutils: It is a series of OpenCV + convenience functions for translation, rotation, resizing, and skeletonization.

Dlib 19.4.0 :dlib is a generally written in the programming language C++ toolkit which consists of machine leaning algorithms in C to solve the real world problems

OpenCV2 3.2.0 :OpenCV is a cross-platform library. That shall be used to develop the real-time computer vision applications.

PyautoGUI: is a cross-platform GUI automation module that works on Python. In this, you can control the mouse and keyboard as well as you can perform basic image recognition to automate tasks on your computer.

• Fact Finding Techniques

I researched about Python libraries on different websites about OpenCV, Numpy and pyautogui.

I also went through youtube video lectures in order learn the uage and interconnection of those libraries

• Synopsis

Project Title: Eye Controlled Mouse

Project Description:
An Eye Controlled mouse meaning the cursor can be controlled by the movment of the users iris and right click can be made by a blink of an eye. The software requirements for this technique include Python, OpenCV, NumPy, and a few more facial recognition tools, as well as a basic camera.

An 'eye-tracking mouse' is a gadget that tracks the user's eye movements. The project relies on mapping facial traits to the cursor to recognize and capture them in video. When the camera is opened, the application must extract all of the video's frames.

System Requirement Specification:

Software Specifications

- Python
- Open CV
- Numpy

Hardware Specifications

- Intel Pentium 4 Processor
- 512 MB RAM
- 10 GB Hard Disk
- Webcam of minimum 720p Resolution

Team Size: Solo

Yeshu Wanjari

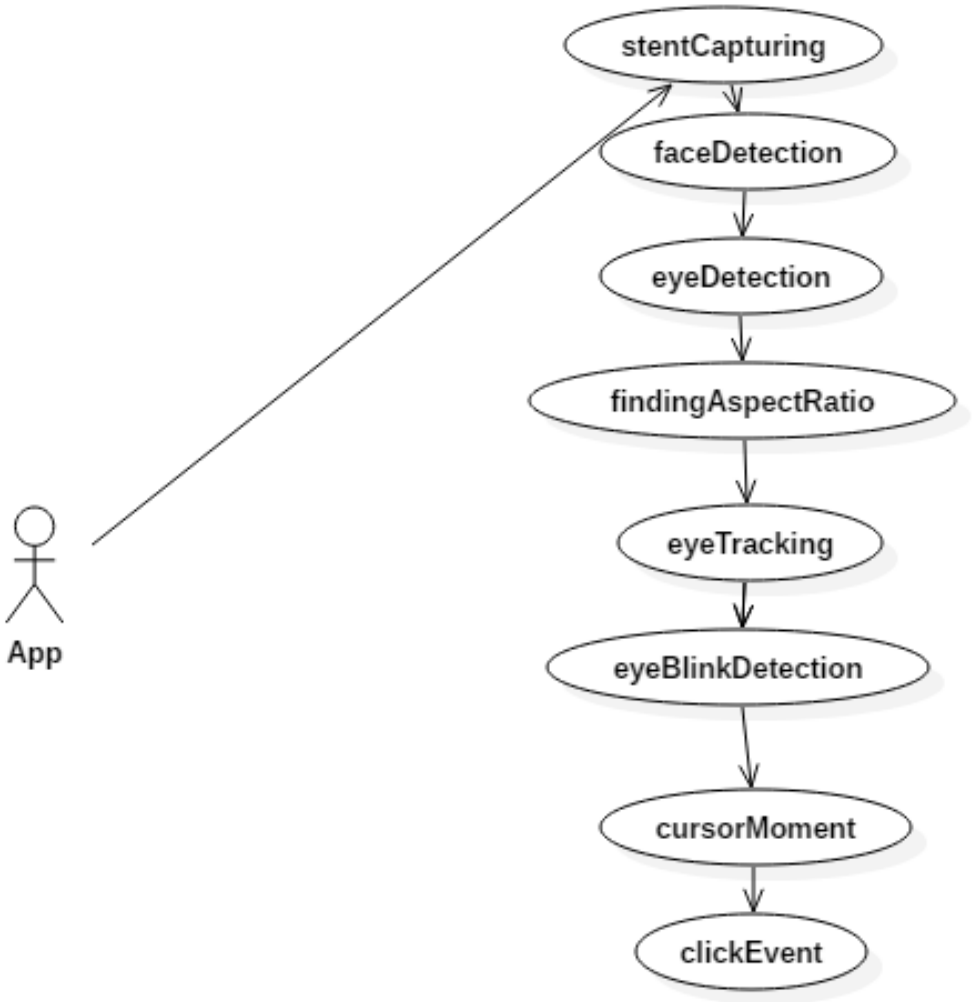
Estimated Time: 12 weeks

• System Analysis

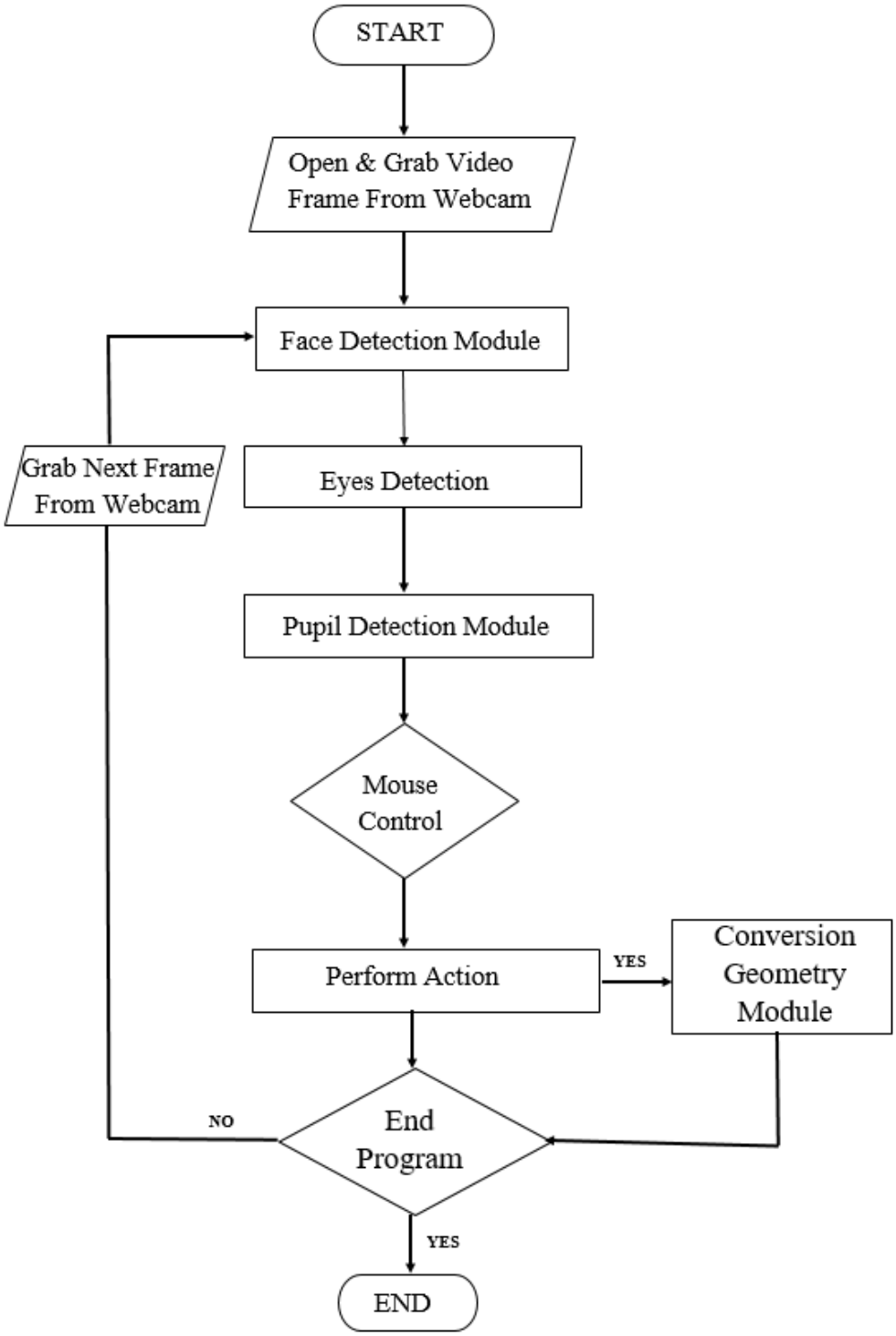
1. Software requirements to run The Eye controlled Mouse Project
    i. Python
    ii. OpenCV
    iii. Numpy
    iv. Webcam

• UML Diagrams

1. Use Case Diagram:



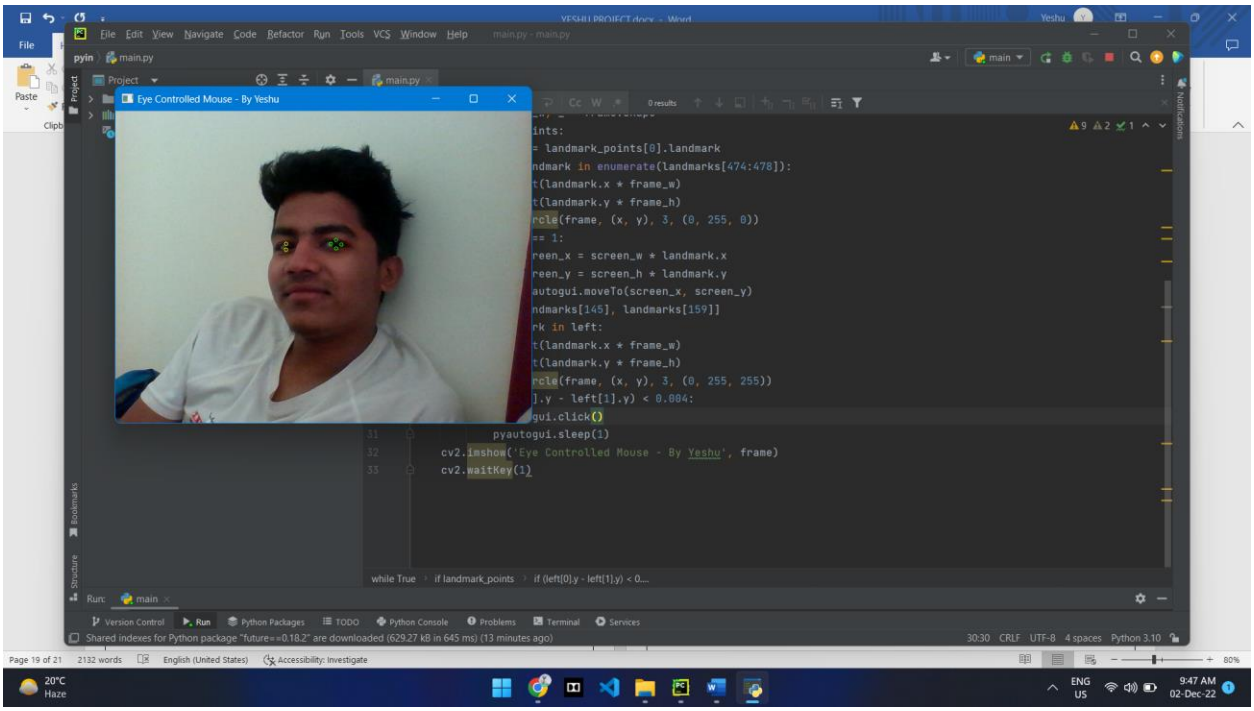
2. Activity Diagram

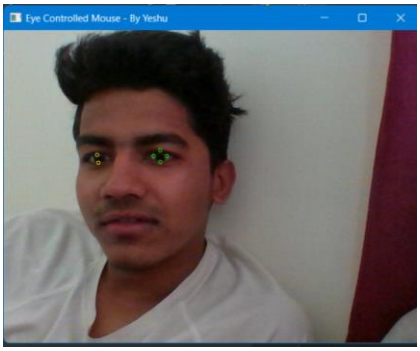


- **System Design & Coding/Development**

Code for Eye Controlled mouse

```
import cv2
import mediapipe as mp
import pyautogui
cam = cv2.VideoCapture(0)
face_mesh =
mp.solutions.face_mesh.FaceMesh(refine_landmarks=True)
screen_w, screen_h = pyautogui.size()
while True:
    _, frame = cam.read()
    frame = cv2.flip(frame, 1)
    rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    output = face_mesh.process(rgb_frame)
    landmark_points = output.multi_face_landmarks
    frame_h, frame_w, _ = frame.shape
    if landmark_points:
        landmarks = landmark_points[0].landmark
        for id, landmark in enumerate(landmarks[474:478]):
            x = int(landmark.x * frame_w)
            y = int(landmark.y * frame_h)
            cv2.circle(frame, (x, y), 3, (0, 255, 0))
            if id == 1:
                screen_x = screen_w * landmark.x
                screen_y = screen_h * landmark.y
                pyautogui.moveTo(screen_x, screen_y)
        left = [landmarks[145], landmarks[159]]
        for landmark in left:
            x = int(landmark.x * frame_w)
            y = int(landmark.y * frame_h)
            cv2.circle(frame, (x, y), 3, (0, 255, 255))
        if (left[0].y - left[1].y) < 0.004:
            pyautogui.click()
            pyautogui.sleep(1)
    cv2.imshow('Eye Controlled Mouse - By Yeshu', frame)
    cv2.waitKey(1)
```



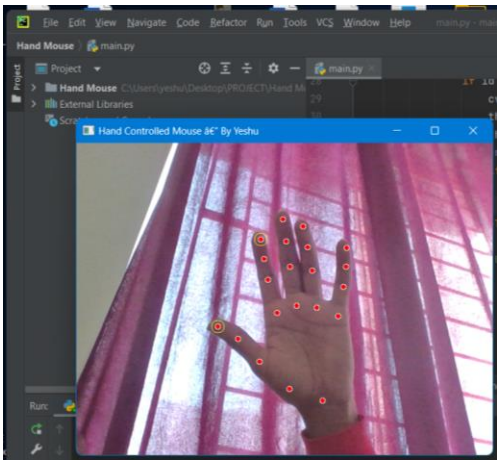


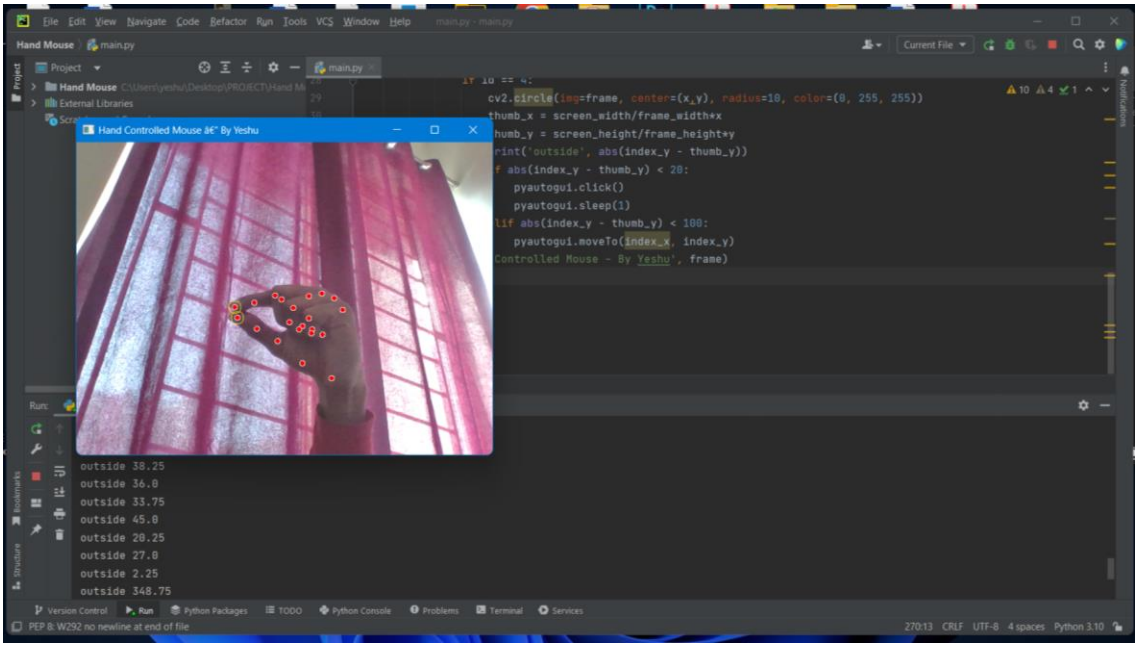
Code for Hand Controlled Mouse

```
import cv2
import mediapipe as mp
import pyautogui
cap = cv2.VideoCapture(0)
hand_detector = mp.solutions.hands.Hands()
drawing_utils = mp.solutions.drawing_utils
screen_width, screen_height = pyautogui.size()
index_y = 0
while True:
    _, frame = cap.read()
    frame = cv2.flip(frame, 1)
    frame_height, frame_width, _ = frame.shape
    rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    output = hand_detector.process(rgb_frame)
    hands = output.multi_hand_landmarks
```

```
    if hands:
        for hand in hands:
            drawing_utils.draw_landmarks(frame, hand)
            landmarks = hand.landmark
            for id, landmark in enumerate(landmarks):
                x = int(landmark.x*frame_width)
                y = int(landmark.y*frame_height)
                if id == 8:
                    cv2.circle(img=frame, center=(x,y),
radius=10, color=(0, 255, 255))
                    index_x = screen_width/frame_width*x
                    index_y = screen_height/frame_height*y

                if id == 4:
                    cv2.circle(img=frame, center=(x,y),
radius=10, color=(0, 255, 255))
                    thumb_x = screen_width/frame_width*x
                    thumb_y = screen_height/frame_height*y
                    print('outside', abs(index_y - thumb_y))
                    if abs(index_y - thumb_y) < 20:
                        pyautogui.click()
                        pyautogui.sleep(1)
                    elif abs(index_y - thumb_y) < 100:
                        pyautogui.moveTo(index_x, index_y)
    cv2.imshow('Hand Controlled Mouse - By Yeshu ', frame)
    cv2.waitKey(1)
```





• System Implementation & Evaluation

The Python libraries I used for this technique include Python, OpenCV, NumPy, and a few more facial recognition tools, as well as a basic camera.

• Advantages of The System

- An Ease of use for Disabled Population
- Easy access in case users user cannot touch the device
- Accurate
- Fast and Easy Cursor movement and clicking
- Real time Iris Tracking

• Future Enhancements

I will be taking usage feedbacks to make advancements.

We will try to make it more fluent and smooth.

The Face recognition and tracing can be improved by a greater extent

Scrolling movement and Left click can be added for Hand Controlled mouse

• System Maintenance

Latest Python version should be used. Python version lesser than 3.7 is incapable to run the libraries used in this project. All the used libraries should be imported. A system camera with decent clarity at least having a 720p resolution is required

• User Manual

For Eye controlled mouse

- The user has to sit in front of Camera with decent lighting conditions
- The mouse movement can be controlled by moving the head
- Right click can be simulated by a blink of left eye

For Hand Controlled mouse

- The index finger and thumb are joined and moved to move the cursor
- The fingers are tapped to simulate a right click