OPTIGUARD EYE PROTECTION SYSTEM

A MINI PROJECT REPORT

Submitted by

ABIKASH SHIVAM G (913120104003)

ARUNNACHALAM R S (913120104014)

KALILA C N A (913120104042)

MOHAMED IMRAN KHAN I (913120104055)

in partial fulfillment for the award of the degree
of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

VELAMMAL COLLEGE OF ENGINEERING AND TECHNOLOGY

ANNA UNIVERSITY: CHENNAI 600 025

JUNE 2023

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"Optiguard Eye Protection System"** is the bonafide work of **"ABIKASH SHIVAM G (913120104003), ARUNNACHALAM R S (913120104014), KALILA C N A (913120104042), MOHAMED IMRAN KHAN I (913120104055)."** who carried out the project work under my supervision.

SIGNATURE Dr. P.ALLI, MS, Ph.D.	**SIGNATURE** Mrs. K. AZARUDEEN, B.E.., M.E.,
HEAD OF THE DEPARTMENT, DEAN & PROFESSOR	**SUPERVISOR,** ASSISTANT PROFESSOR,
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, VELAMMAL COLLEGE OF ENGINEERING AND TECHNOLOGY, MADURAI-9	DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, VELAMMAL COLLEGE OF ENGINEERING AND TECHNOLOGY, MADURAI-9

Submitted for the university viva- voce held on__________________at Velammal College of Engineering and Technology.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

Optiguard Eye Protection System uses Python, OpenCV and MediaPipe to detect the user's face and the distance between the eye and the display is calculated and rotate the display screen to reduce eye strains and other problems. Anyone who spends a lot of time on their digital devices will find this system very useful as it is easy to use. This system was tested between the distance range of 45 to 50 cm. This system is very affordable and does not require any proprietary sensor or powerful and expensive hardware. The results show that the average accuracy of 100% and 90% are achieved in the iris detection.

Keywords: Eye protection, Optiguard, Facial recognition, Python, MediaPipe, Computer vision (CV), Machine learning.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

1.1 PROBLEM DEFINITION

Modern life is dominated by digital screens, and many of us are glued to our smartphones, computers, and other devices for hours each day. However, prolonged exposure to the blue light these screens produce can damage your eyes, headaches, and other problems. This study describes a technique to protect the eyes from the harmful effects of prolonged screen use. Our system uses Python, OpenCV and MediaPipe to detect the user's face and the distance between the eye and the display is calculated and rotate the display screen to reduce eye strains and other problems. Anyone who spends a lot of time on their digital devices will find this system very useful as it is easy to use. This system was tested between the distance range of 45 to 50 cm. This system is very affordable and does not require any proprietary sensor or powerful and expensive hardware.

1.2 OBJECTIVE

The main objective of the OptiGuard Eye Protection System using Python CV and MediaPipe is to provide an advanced solution for promoting healthy eye habits and preventing eye strain and other eye-related issues. This system aims to achieve this by utilizing advanced eye-tracking capabilities, real-time monitoring, and customizability features.

- Track eye movements and detect
- Monitor the user's screen time and provide alerts when the user exceeds a certain amount of time or takes insufficient breaks.
- Provide customizable settings.
- Integrate with other software and tools.

2. LITERATURE SURVEY

In our survey we researched for multiple sites for Optiguard eye protection system and came across few which were similar to what we needed. By using this survey we havebeen able to add and make changes to our project step by step. The project has been equally divided and completed by all four members of the group.

Below is the details of the sites for Optiguard Eye protection system:

- **Development of Real-Time Eye Tracking Algorithm.** Published in: **2021 4th International Conference on Computing & Information Sciences (ICCIS)** **-** Siti Nuradlin Syahirah Sheikh Anwar; Azrina Abd Aziz; Syed Hasan Adil**.** the facial landmark detector is implemented to detect the eyes as the Region of Interest (ROI), track the eyes and identify the gaze by classifying the eye positions to left, right and middle.
- **Design of Myopia Prevention System Based on Machine Vision** Published in: **2022 IEEE 2nd International Conference on Electronic Technology, Communication and Information (ICETCI)** **-** Wanting Wang; Yuxia Lin; Zhenyu Huang; Xin Liu The proposed work created equipment which has visual monitoring, sound source reminder, automatic light source generation, infrared distance survey, time timing and other performance.
- **REAL-TIME EYE BLINK DETECTION BASED ON PYTHON.** Published in: **The 8th International Symposium on Test Automation & Instrumentation (ISTAI 2020)** **- P. Ran**; **H. Wang.** In this article, Python,

OpenCV, Dlib and other third-party libraries are used to obtain the driver's face video through the mobile phone camera and detect the changes of eye.

- **Research on Eyesight Protection-Oriented Teaching Aids Using Machine Vision Technology**. Published in: **2021 IEEE International Conference on Educational Technology (ICET) -** Jian Xu.

This paper adopts the technical idea of "face detection-eye recognition-spatial positioning", from coarse to fine, and constructs an adaptive vision protection desk system based on machine vision. Finally, the vertical distance between the human eyes and the desktop is obtained through the optimized disparity calculation formula combined with the height calculation equation.

- **Image Recognition based Driver Drowsiness Detection using Python**. Published in: **2022 International Conference on Electronics and Renewable Systems (ICEARS).** K. Srisabarimani; R. Arthi; Pattabiraman B; Sabarish Suburamani; Surya A. E; Sudarssh Narrayan K**.** This paper provides a detailed study to detect the langour of the driver using python programming language and haar training algorithm to identify the eye movements of the driver by capturing image respectively.

3. ABOUT THE SYSTEM

3.1 EXISTING SYSTEM

There are a number of existing systems that utilize Python CV for eye protection, but they typically lack the advanced features and real-time monitoring capabilities of the OptiGuard Eye Protection System. These systems often focus on basic image processing techniques, such as adjusting screen brightness or color temperature, to reduce eye strain and fatigue.

One example of an existing system is F.lux, which is a free software that adjusts the color temperature of a computer screen based on the time of day. F.lux reduces blue light emissions from screens, which can interfere with circadian rhythms and disrupt sleep patterns. However, F.lux does not provide eye-tracking or real-time monitoring capabilities, and it only addresses one aspect of eye protection.

Another example is Iris, which is a software that uses various image processing techniques to reduce screen flicker and blue light emissions. Iris offers customizable settings, such as brightness and contrast adjustments, and can be used on various devices. However, like F.lux, Iris does not offer advanced eye-tracking capabilities or real-time monitoring.

While these existing systems can be helpful in reducing eye strain and fatigue, they lack the advanced features and functionality of the OptiGuard Eye Protection System. OptiGuard offers real-time monitoring and alerts based on advanced eye-tracking capabilities, allowing users to maintain healthy eye habits and prevent eye-related issues. Additionally, OptiGuard provides customizable settings, such as screen brightness and color temperature, and can be integrated with other software and tools for a more comprehensive eye protection solution.

DISADVANTAGES OF EXISTING SYSTEM:

- Limited eye-tracking accuracy.
- Limited customization options.
- Limited compatibility with certain devices.
- Difficulty to handle.
- systems may be costly.
- Some systems may not offer real-time monitoring and alerting features.
- Not Efficient.

3.2 NEED FOR NEW SYSTEM

Eye strain and fatigue are common problems associated with prolonged computer use, and can cause discomfort and reduce productivity. The OptiGuard Eye Protection System can help reduce the risk of these issues by reminding users to take breaks or adjust their posture.

The new system can offer more advanced eye-tracking capabilities than existing systems. It can be designed to integrate with other software or tools, providing a more comprehensive eye health and wellness solution. It can include more accurate and reliable eye-tracking, the ability to detect specific eye conditions, and real-time monitoring and alerting features. It can offer greater customization options to meet the specific needs and preferences of users.

3.3 PROPOSED SYSTEM

The proposed system uses a webcam to detect the user's eyes and track their movements. It uses the Mediapipe framework to accurately detect eye and facial features. It then analyzes the color spectrum of the digital screen to detect the intensity of blue, ultraviolet and infrared light. Based on this input, the system determines whether the user is at risk of damaging their eyes and warns the user with appropriate warnings and suggestions.

The first step of the system is to detect the user's eyes and track their movements. The Mediapipe framework provides a face detection model that can detect the user's face and locate the eyes accurately. We use the OpenCV library to capture the video stream from the webcam and feed it into the Mediapipe face detection model. The model returns the coordinates of the facial landmarks, including the eyes. we can also implement a feature that rotates the screen when the user's eyes are too close to the screen.

The main advantages of the proposed system are –

- Accurate Eye detection.
- Easy Customization.
- Cost-Effective.
- Easy Integration with other tools.
- Real Time Monitoring.

4. REQUIREMENT ANALYSIS

4.1 SOFTWARE REQUIREMENTS

Operating system	-	Windows 7 or more
Tools	-	Pycharm or other tools for running python.
Libraries	-	Mediapipe and Python OpenCV.
Language	-	Python 3.x or more.

4.2 HARDWARE REQUIREMENTS

Processor	-	Core i3, 10th gen, 1.19 GHz
RAM	-	12 GB
Hard disk	-	1 TB
System Type	-	64-bit operating system
Monitor	-	11.2’
Mouse	-	104 keys US Key Serial, USB or PS/2
Webcam	-	Compactible with laptop and Pc.

5. SYSTEM ARCHITEXTURE

5.1 UML DIAGRAM:

USE CASE DIAGRAM:

In this use case diagram, the main actor is the User, who interact with the OptiGuard Eye Protection System. The Python CV and Mediapipe libraries are responsible for detecting the user's face and eyes in real-time, tracking their movement, and providing real-time feedback to the user or supervisor.

6. WORKING ENVIRONMENT

PYTHON:

Python is a popular high-level programming language that is known for its simplicity, readability, and ease of use. Python is a versatile language that can be used for a wide range of applications, including web development, scientific computing, **data analysis**, artificial intelligence, machine learning, and automation, among others. It has a large and active community of developers who contribute to its open-source libraries and tools, making it easy to find solutions to common problems. One of the main reasons for Python's popularity is its vast ecosystem of third-party libraries and frameworks. These libraries provide developers with pre-built modules and functions that can be easily integrated into their applications. Some popular Python libraries include NumPy, Pandas, Matplotlib, SciPy, and TensorFlow.

Python also has a large and active community of developers who contribute to its open-source libraries and tools, making it easy to find solutions to common problems. The language has a strong reputation for being easy to learn, and there are many resources available online for those who are new to programming.

In summary, Python is a popular high-level programming language that is known for its simplicity, readability, and ease of use. It is versatile, with a large ecosystem of libraries and tools, making it ideal for a wide range of applications. It also has a large and active community of developers who contribute to its **open-source libraries** and tools, making it easy to find solutions to common problems.

MEDIAPIPE:

MediaPipe is an open-source framework developed by Google that provides a pipeline of customizable building blocks for building real-time multimodal applications for mobile, web, and desktop. It is based on computer vision and machine learning techniques and can handle tasks such as video and audio processing, object detection and tracking, hand and **facial recognition**, and pose estimation, among others. MediaPipe also supports the development of custom graphs through a high-level Python API or a low-level C++ API. This allows developers to create their own processing nodes or customize existing ones to fit their specific needs. Developers can also leverage MediaPipe's machine learning models or integrate their own models to improve the accuracy of their applications. MediaPipe provides a set of pre-built graphs for common tasks, such as face detection, hand tracking, object detection, and pose estimation, among others. These pre-built graphs are optimized for performance and can be easily integrated into applications without the need for specialized knowledge or skills in **computer vision** or machine learning.

PYTHON CV:

Python CV (Computer Vision) is a branch of computer science that deals with the ability of machines to interpret and understand visual information from the world around us. Python CV is built using the Python programming language, and it has a wide range of applications, including object detection, **facial recognition**, image segmentation, and image classification.

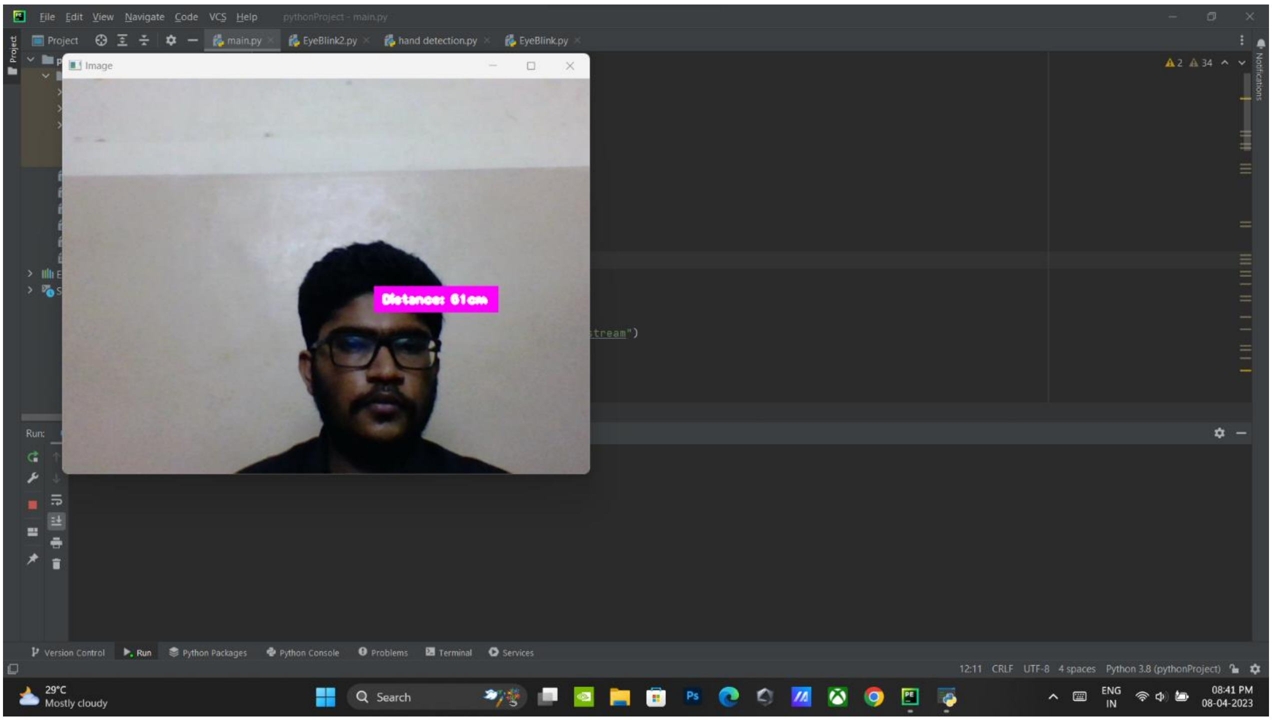
One of the key strengths of Python CV is its ability to leverage machine learning algorithms to improve the accuracy and effectiveness of visual processing. **Machine learning** algorithms can be used to train models that can detect patterns in visual data and make predictions about the objects or people in a given scene. These models can be trained using a wide range of machine learning algorithms, including neural networks and decision trees.

Python CV is a powerful tool for solving problems in the field of computer vision. It is built using the Python programming language and leverages a wide range of tools and libraries, including OpenCV, to provide developers with a wide range of functionality. Python CV can be used for a wide range of applications, including healthcare, automotive, manufacturing, and security, and it is **highly customizable**, allowing developers to create their own algorithms and tools to solve specific problems.

7. MODULE DESCRIPTION

7.1 PYTHON EYE DETECTION MODULE:

The first step of the system is to detect the user's eyes and track their movements. The Mediapipe framework provides a face detection model that can detect the user's face and locate the eyes accurately. We use the OpenCV library to capture the video stream from the webcam and feed it into the Mediapipe face detection model.



7.2 PYTHON ROTATE SCREEN MODULE:

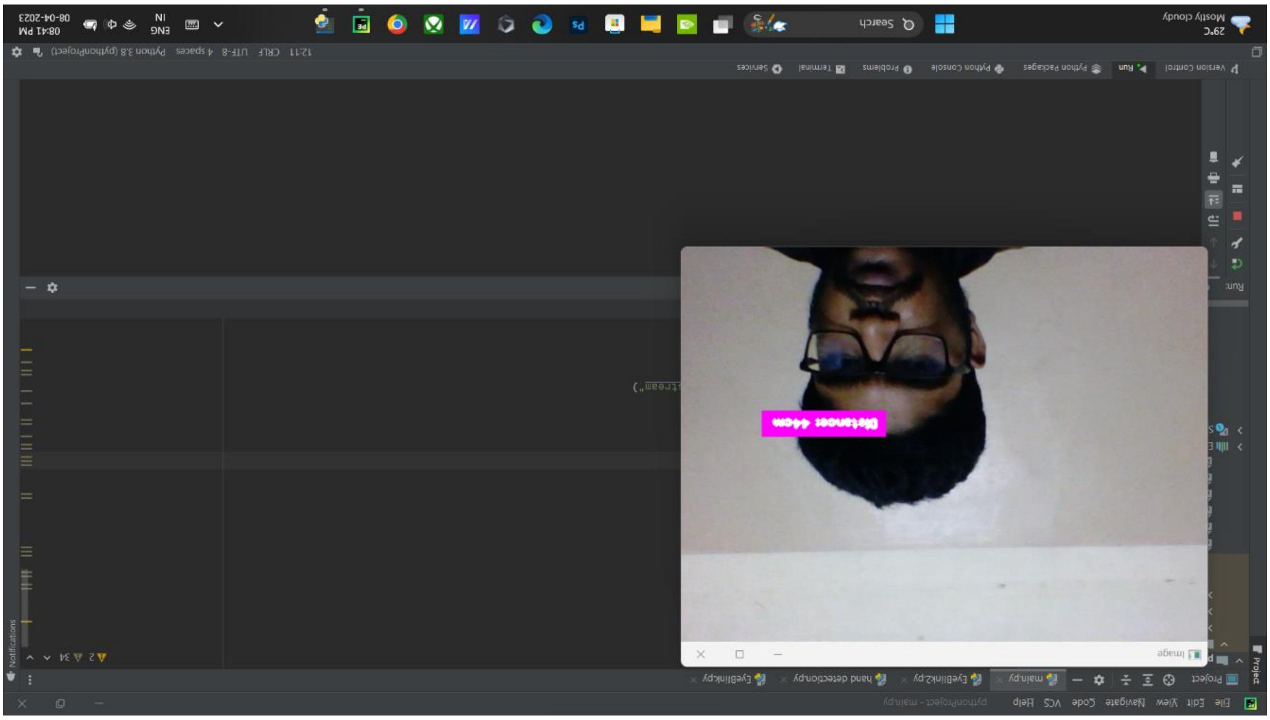
We can use the distance formula to calculate the distance between the user's eyes in pixels. The distance formula is as follows

pointLeft = face[145] pointRight = face[374]

w = detector.findDistance(pointLeft,pointRight) W=6.3

d = (W*f)/w

Once we have calculated the distance, we can compare it to a minimum threshold. If the distance is below the threshold, we can rotate the screen by 90 degrees to reduce eye strain and prevent other issues. This system was tested between the distance range of 45 to 50 cm. This system is very affordable and does not require any proprietary sensor or powerful and expensive hardware. The results show that the average accuracy of 100% and 90% are achieved in the iris detection.



8. SYSTEM IMPLEMENTATION

Implementation is the stage in the project where the theoretical design is turned into a working system and is giving confidence on the new system for the users that it will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the change over, an evaluation of change over methods. Apart from planning major task of preparing the implementation are education and training of users. The implementation process begins with preparing a plan for the implementation of the system. According to this plan, the activities are to be carried out, discussions made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system. In network backup system no additional resources are needed. Implementation is the final and the most important phase. The most critical stage in achieving a successful new system is giving the users confidence that the new system will work and be effective. The system can be implemented only after thorough testing is done and if it is found to be working according to the specification. This method also offers the greatest security since the old system can take over if the errors are found or inability to handle certain type of transactions while using the new system.

9. SYSTEM TESTING

As the part of system testing we execute the program with the intent of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied. The ultimate aim is quality assurance. Tests are carried out and the results are compared with the expected document. The various tests performed are unit testing, integration testing and user acceptance testing.

9.1 UNIT TESTING

As a part of unit testing we executed the program for individual modules independently. This enables, to detect errors in coding and logic that are contained within each of the three module. This testing includes entering data that is filling forms and ascertaining if the value matches to the type and entered into the database. The various controls are tested to ensure that each performs its action as required.

9.2 INTEGRATION TESTING

Integration testing is a systematic testing to discover errors associated within the interface. The objective is to take unit tested modules and build a program structure. All the modules are combined and tested as a whole. Here the admin module, sec module and student module options are integrated and tested. This testing provides the assurance that the application is well integrated functional unit with smooth transition of data.

9.3 USER ACCEPTANCE TESTING

User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keep the records of applicants and making changes to the details and password whenever required.

10. SAMPLE CODING

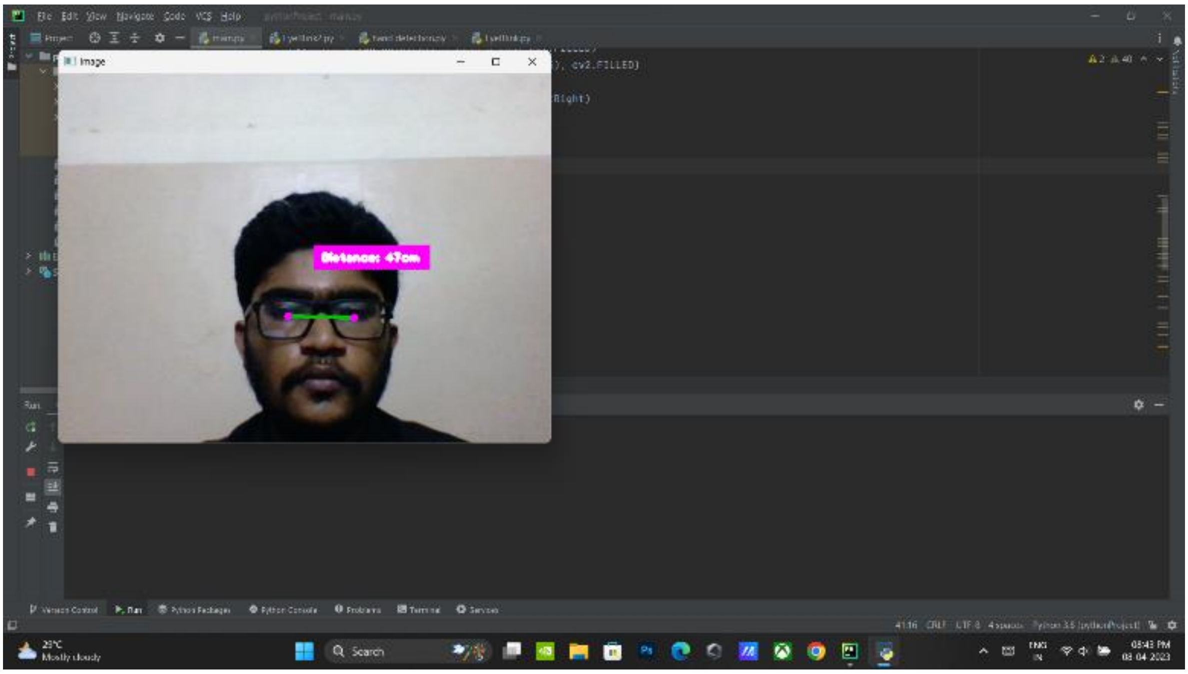
PYTHON:

```
import time

import cv2
import cvzone
import rotatescreen

import serial
from cvzone.FaceMeshModule import FaceMeshDetector

# esp32 = serial.Serial('COM3',115200)
time.sleep(1)
cmd = "ON"

control = 1
if(control==0):
 cap = cv2.VideoCapture("http://192.168.43.247:81/stream")
else:
 cap=cv2.VideoCapture(0)
detector=FaceMeshDetector(maxFaces=1)



while True:
    success,img = cap.read()
    img,faces = detector.findFaceMesh(img,draw=False)
    # time.sleep(0.3)
```

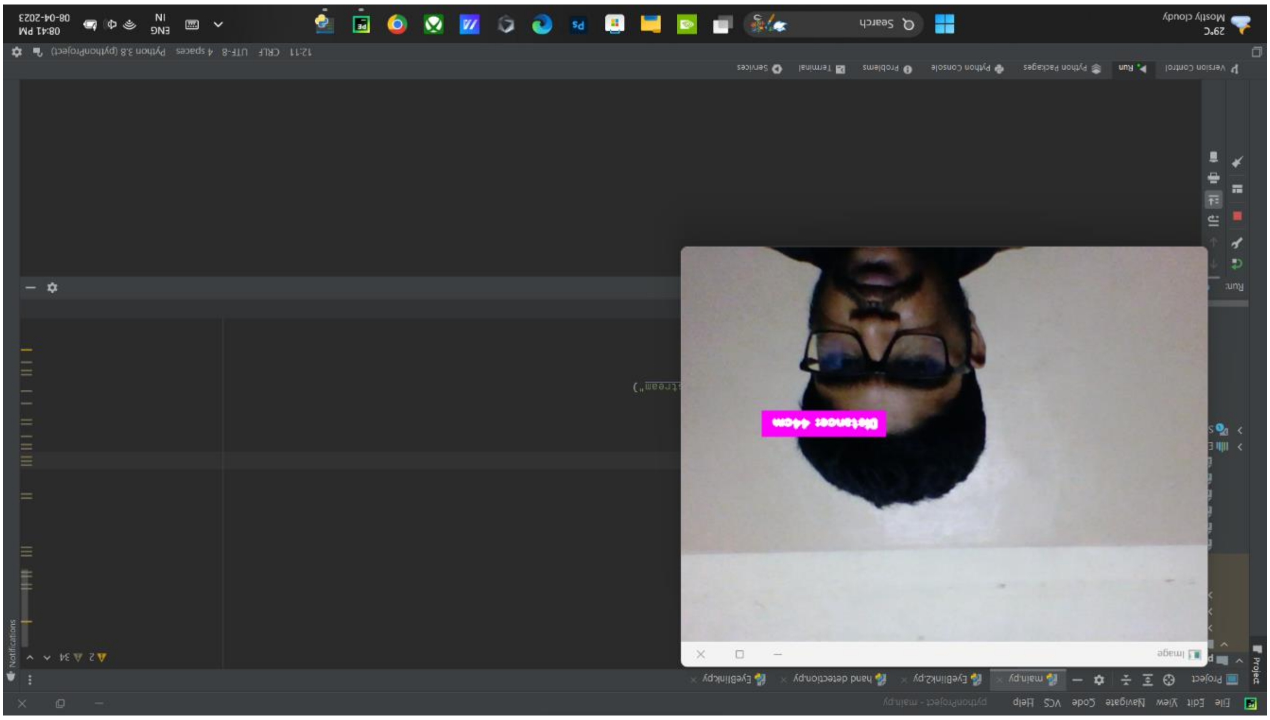
```
if faces:
    face = faces[0]
    pointLeft = face[145]
    pointRight = face[374]

    cv2.line(img, pointLeft, pointRight, (0, 200, 0), 3)
    cv2.circle(img,pointLeft,5,(255,0,255),cv2.FILLED)
    cv2.circle(img, pointRight, 5, (255, 0, 255), cv2.FILLED)

    w,_ = detector.findDistance(pointLeft,pointRight)
    W=6.3 # average distance between human eyes

    # #focal length
    # W=6.3
    # d=50
    # f= (w*d)/W

    #distance
    if(control == 0):
      f=300 #650
    else:
      f=650
    d =  (W*f)/w

    #print(d)

    screen = rotatescreen.get_primary_display()
    if (d <= 50 ):
        cmd ="ON"
        screen.rotate_to(180)
        else:
            cmd = "OFF"
            screen.rotate_to(0)
        cmd=cmd+'\r'

        # esp32.write(cmd.encode())

        cvzone.putTextRect(img, f'Distance: {int(d)}cm', (face[10][0] ,
face[10][1])#100
                       , scale=1)

    cv2.imshow("Image",img)

    if cv2.waitKey(1) == ord("q"):
        screen.rotate_to(0)
        break
```

11. RESULT

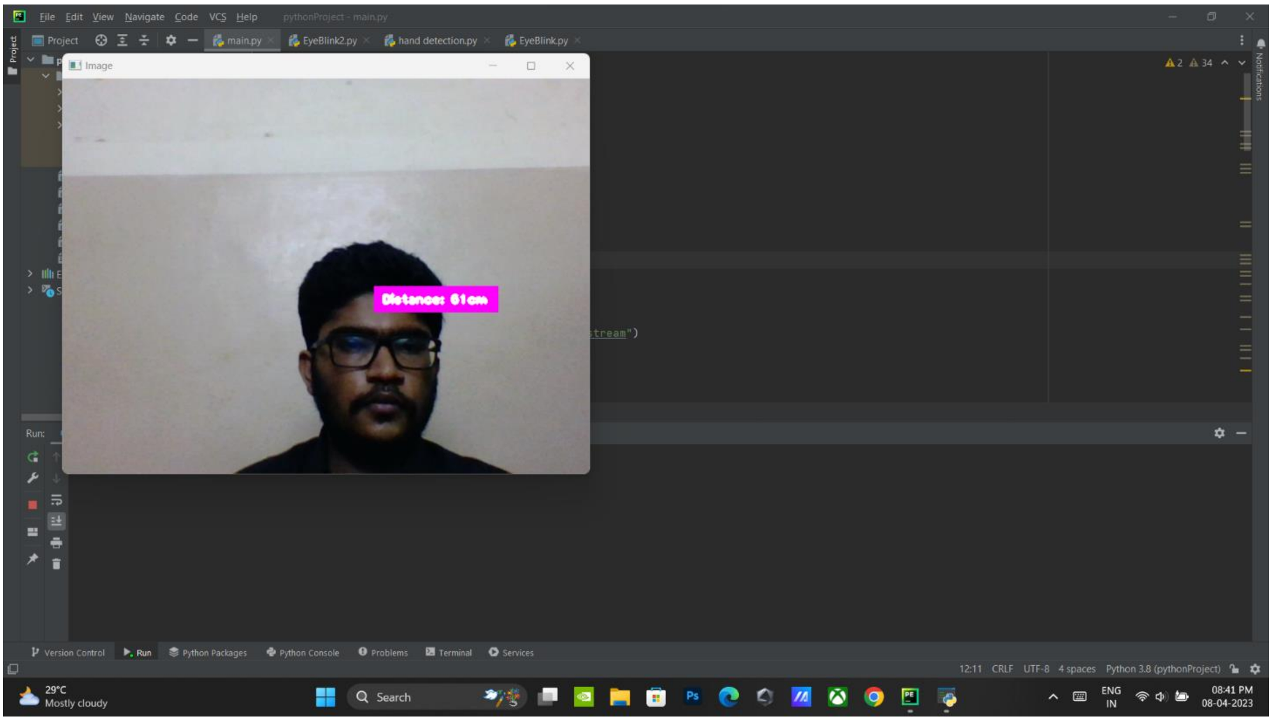
1. DETECTION OF EYE VIEW DISTANCE AND DISPLAY SCREEN :



2. ROTATION OF SCREEN:



3. RETURNS TO NORMAL STATE IF DISTANCE IS LESS:



12. CONCLUSION

In conclusion, an eye protection system using Python is a valuable tool for preventing eye strain, fatigue, and other negative effects of prolonged digital screen use. By detecting the eyes in the video input, the system can track the user's gaze and provide alerts or suggestions for taking breaks. Python offers several libraries and frameworks for implementing an eye protection system, such as OpenCV, Mediapipe. By utilizing the power of Python and computer vision technology, we can improve the well-being of individuals and promote a healthier digital lifestyle.

13. REFERENCES

- **Research on Eyesight Protection-Oriented Teaching Aids Using Machine Vision Technology.** : **2021 IEEE International Conference on Educational Technology (ICET)** **-** Jian Xu; Yabo Luo**.**
- **A Solution for Monitoring the Eye's Viewing Distance from Electronical Display. 2022 International Conference on Communication, Computing and Internet of Things (IC3IoT). K. Srisabarimani; R. Arthi; Pattabiraman B; Sabarish Suburamani; Surya A. E; Sudarssh Narrayan K.**
- **Research on real-time distance measurement of mobile eye tracking system based on neural network.** : **2022 IEEE 2nd International Conference on Electronic Technology, Communication and Information (ICETCI)** **-** Wanting Wang; Yuxia Lin; Zhenyu Huang; Xin Liu