STRAIN ANALYSIS BASED ON EYE BLINKING

UG PROGECT PHASE-1

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

Submitted by

SRIYA GUDLA	**19UK1A0518**
VAMSHI MAMIDALA	**19UK1A0516**
RAMYA BOLLEPELLI	**19UK1A0553**

Under the esteemed guidance of

Mr. A. ASHOK KUMAR

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VAAGDEVI ENGINEERING COLLEGE
(Affiliated to JNTUH, Hyderabad)
Bollikunta, Warangal –
506005 **2019– 2023**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINNERING

VAAGDEVI ENGINEERING COLLEGE

BOLLIKUNTA, WARANGAL – 506005

2019 – 2023

CERTIFICATE OF COMPLETION

UG PROJECT PHASE-1

This is to certify that the UG Project Phase-1 entitled “**STRAIN ANALYSIS BASED ON EYE BLINKING**”Isbeingsubmittedby***GUDLASRIYA(H.NO:19UK1A0518),MAMIDALAVAMSHI(H.NO:19UK1A0516),BOLL EPELLI RAMYA(H.NO:19UK1A0553),***in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering to Jawaharlal Nehru Technological University Hyderabad during the academic year 2022-23, is a record of work carried out by them under the guidance and supervision.

Project Guide	**Head of the Department**
Mr. A. ASHOK KUMAR	**Dr. R. NaveenKumar**
(Assistant Professor)	(Professor)

External

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr.P.PRASAD RAO,** Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this UG Project Phase-1 in the institute.

We extend our heartfelt thanks to **Dr.R.NAVEEN KUMAR**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG Project Phase-1.

We express heartfelt thanks to Smart Bridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the UG Project Phase-1 and for their support in completing the UG Project Phase-1.

We express heartfelt thanks to the guide, **A.ASHOK KUMAR,** Assistant professor, Department of CSE for his constant support and giving necessary guidance for completion of this UG Project Phase-1.

Finally, we express our sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experience throughout the thesis.

SRIYAGUDLA	**(19UK1A0518)**
VAMSHI MAMIDALA	**(19UK1A0516)**
RAMYA BOLLEPALLY	**(19UK1A0553)**

ABSTRACT

Blinking is a reflex, which means your body does it automatically. Babies and children only blink about two times per minute. By the time you reach adolescence that increases to 14 to 17 times per minute.

Detecting eye blinks is important for instance in systems that monitor a human operator vigilance, e.g. driver drowsiness, in systems that warn a computer user staring at the screen without blinking for a long time to prevent the dry eye and the computer vision syndromes, in human-computer interfaces that ease communication for disabled people. There should be an application that monitors to let the user know that he might get strained.

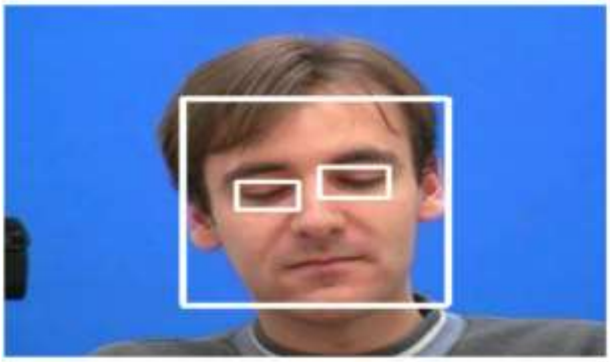
A neural network model is built which alerts the user if eyes are getting strained. This model uses the integrated webcam to capture the face (eyes) of the person. It captures the eye movement and counts the number of times a person blinks. If blink count deviates from the average value (if the number of blinks is less or more), then an alert is initiated by playing an audio message along with a popup message is displayed on the screen appropriately.

TABLE OF CONTENTS:-

1.INTRODUCTION

1.1.OVERVIEW

A neural network model is built which alerts the user if eyes are getting strained. This model uses the integrated webcam to capture the face (eyes) of the person. It captures the eye movement and counts the number of times a person blinks. If blink count deviates from the average value (if the number of blinks is less or more), then an alert is initiated by playing an audio message along with a popup message is displayed on the screen appropriately.



1.2.PURPOSE

It captures the eye movement and counts the number of times a person blinks. Based on our analysis we built a simple application **to measure the stress levels of a person**. We have found a threshold value for the blink rate to find out if a person is stressed or not. If the person's blink rate is more than the threshold value we conclude that the person is stressed.

PROJECT GOALS:

By the end of this project you will:

- know fundamental Computer vision, google text to speech.
- Gain a broad understanding of face landmark detection.
- know how to install necessary packages and setting up the environment.
- Calculate Eye aspect ration
- Work with google text to speech
- Work with Tkinter

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

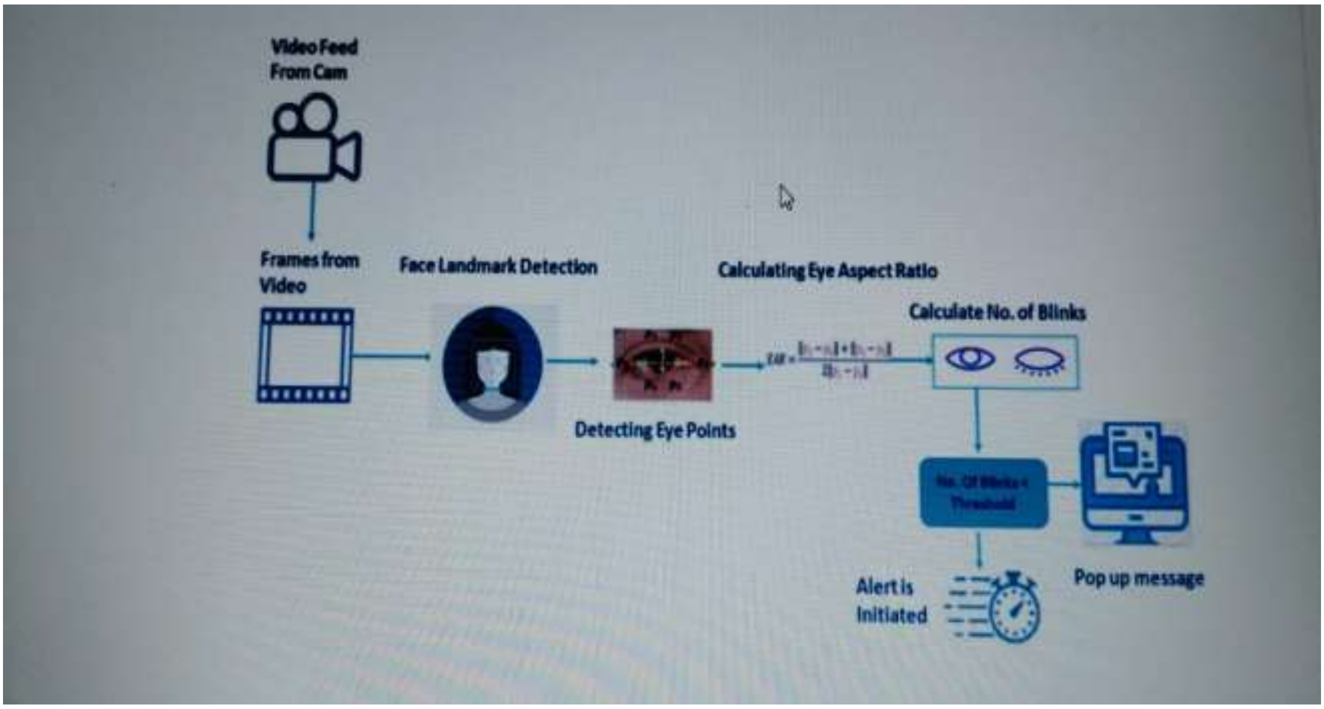
Uncorrected vision problems and eye strain. Uncorrected short or long-sightedness can cause excessive blinking as you try to focus**.** StressIt isn't uncommon for adults and children to develop excessive or forceful blinking habits in response to stress, anxiety or other triggers.

2.2 PROPOSED SOLLUTION

presents an approach for real-time remote detection of eye- blink parameters. First, a combination of boosted classifiers and Lucas- Kanade tracking is used to follow the movement of face and eyes. Then, detailed eye movement is described by normal flow. Finally, a discrete finite state machine is used to detect eye-blinks. The proposed approach is evaluated on a series of short video sequences. It shows promising eye- blink detection capabilities that could be used for software-based prevention of workplace-related disorders. In the European Community more than 40% of the today's working population use computers in their daily work. Computer use is related with static work, constrained sitting and vision problems. For example, approximately 70% of computer workers worldwide are reported to having vision problems leading towards Computer Vision Syndrome. The number of computer-related jobs is expected to increase significantly in the next decade, along with the number of workplace-related illnesses. One of primary causes of vision problems during computer use is insufficient eye movement, caused by long periods of gazing at computer screen. Distance between the screen and the user's head usually doesn't change much and as a result the muscles involved in adaptation of the eye are not exercised for long periods of time, leading to their weakening. This is usually accompanied by decrease in eye blinking frequency, which leads to excessive dryness of the eye surface (cornea and sclera) and can be harmful to the eye. Chronic dry eyes can eventually lead to scarring of the cornea and sight loss. With preventive measures like regular breaks, eye exercises and relaxational activities most of those disorders can be avoided. Unfortunately, the majority of population is reluctant to change their workplace habits until first signs of health issues appear. To help users become aware of the problem and assist them in prevention, we propose to use a simple monitor mounted camera (webcam) for capturing video of user at his workplace and estimating his eye blinking patterns. When potentially harmful behaviour is detected, the user can be alerted and informed about suitable actions.

3.THEORITICAL ANALYSIS

3.1 BLOCK DIAGRAM



3.2 HARDWARE / SOFTWARE DESIGNING

The following is the Hardware/software required to complete this project:

- Internet connection to download and activate
- Administration access to install and run Anaconda Navigator
- A **neural network model** is built which alerts the user if eyes are getting strained. This model uses the **integrated webcam** to capture the face (eyes) of the person.
- Creating a virtualenvironment for installing required libraries like ..dlib,gTTS,scipy,imutils , playsound Opencv_python,numpy ..

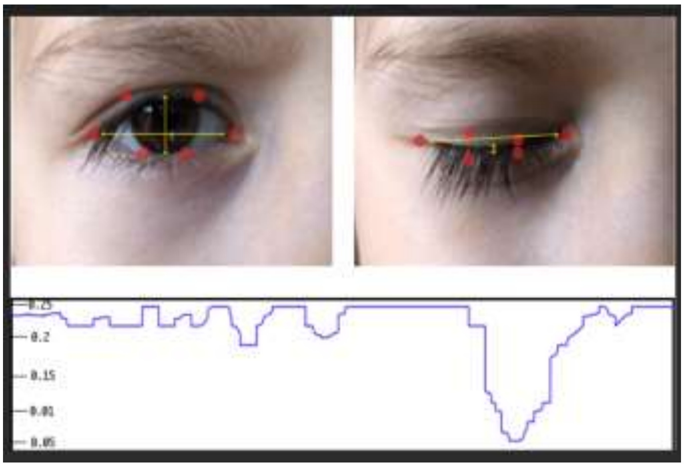
Application builing:

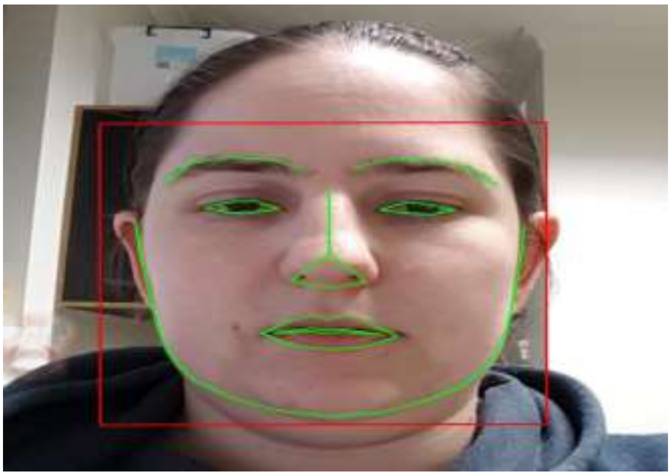
- Python code
- Templates

4.EXPERIMENTAL INVESTIGATION

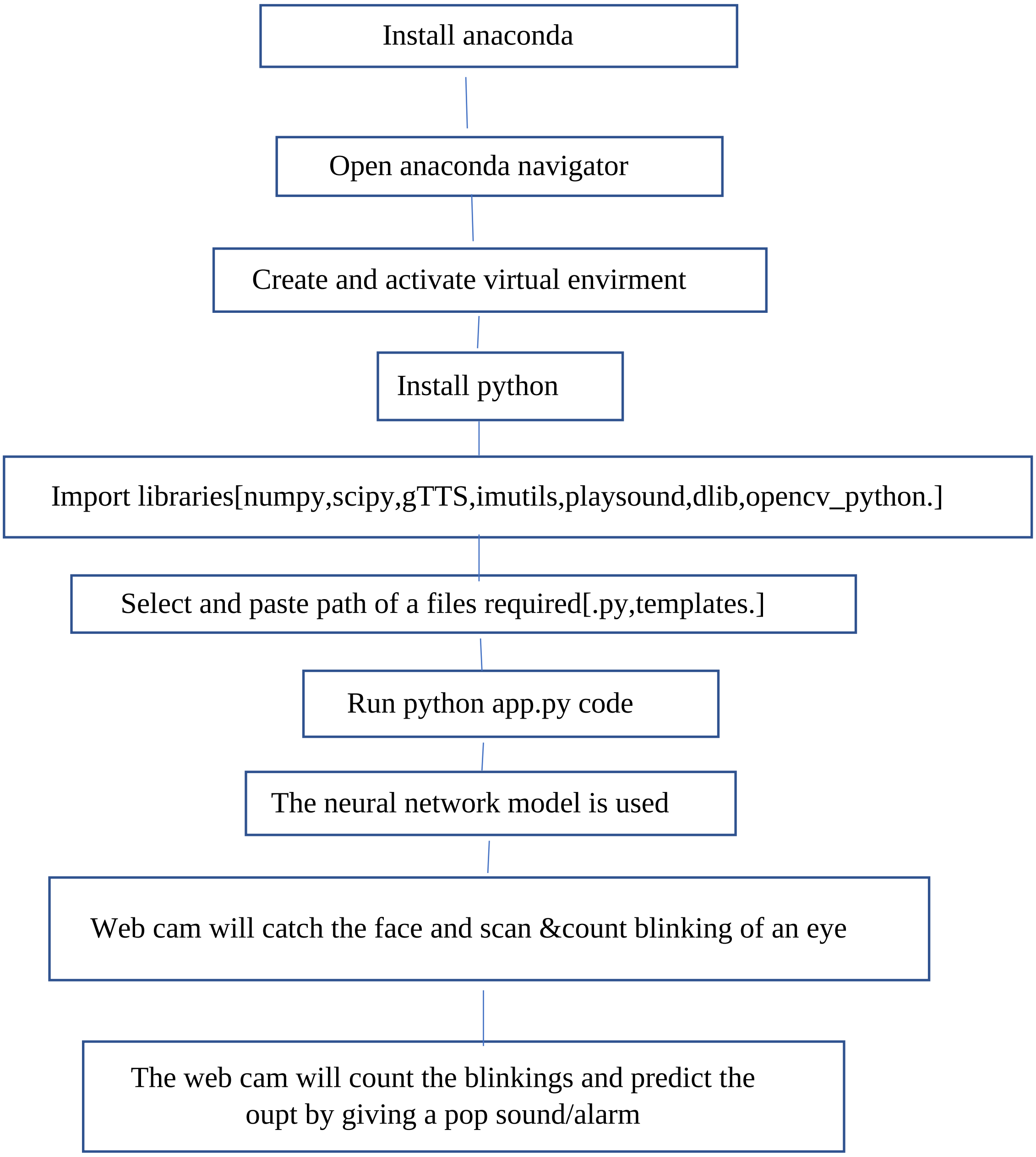
To create an eye blink detector, eyes will be the area on the face that we are interested in. We can divide the process of developing an eye blink detector into the following steps:

1. Detecting the face in the video
2. Detecting facial landmarks of interest (the eyes)
3. Calculating eye width and height
4. Calculating eye aspect ratio (EAR) – relation between the width and the height of the eye
5. Displaying the eye blink counter in the output video
6. Based on the blinks, an alert is initiated to the user with an audio message and popup message.





5.FLOWCHART


Install anaconda
Open anaconda navigator
Create and activate virtual envirment
Install python
Import libraries[numpy,scipy,gTTS,imutils,playsound,dlib,opencv_python.]
Select and paste path of a files required[.py,templates.]
Run python app.py code
The neural network model is used
Web cam will catch the face and scan &count blinking of an eye
The web cam will count the blinkings and predict the oupt by giving a pop sound/alarm

Flow chart description

Tsk 572 (Importing Necessary Libraries)

All the necessary libraries have been successfully imported in the main application file “app_eye.py” . As previously mentioned in the project prerequisites, Jupyter notebook has been used to accomplish this task and later the file has been extracted as a python file with a (.py) extension.

Tsk 573 (Defining Necessary Functions)

All the necessary functions have been successfully defined in the file named “app_eye.py”. The functions are described below,

a. playaudio(text)

In this function, we are translating the text input to a speech by using gTTS and saving the translated speech to the output1.mp3 file. We are returning the output1.mp3 to the calling function.

b. popupmsg(msg)

- Creating an instance of Tk initializes the interpreter and creates the root window
- Giving a title and style to the popup window and configuring it using the configure function.
- A label is used to display text messages Generally the content is static, but your program can change the text.
- pack() is geometry manager organizes widgets in blocks before placing them in the parent widget.
- We are creating an “Okay” button and showing the window using mainloop() function .

c. eye_aspect_ratio(eye) :

We can Calculate Eye Aspect Ration this using the below code

- Compute the Euclidean distances between the two sets of vertical eye landmarks (x, y)-coordinate.
- Compute the Euclidean distance between the horizontal eye landmark (x, y)-coordinates.
- Compute the eye aspect ratio using the above formula and then return ear to the calling function

Tsk 574 (construct argparser)

Our app_eye.py script requires a single command-line argument, followed by a second optional one.

We are using argparse library to parse command-line arguments.ArgumentParser() is a predefined class. let us create an object ‘ap’ for it so that we can access it in our program.

Tsk 575 (Defining important constants)

Let us define two important constants

- **EYE_AR_THRESH**

- By using ear value, we determine if a blink is taking place in the video stream.
- A "blink" is registered when ear value falls below a certain EYE_AR_THRESH and then rises above the **EYE_AR_THRESH.**
- We default it to a value of 0.3 as this is what has worked best for my applications, but you may need to tune it for your application.

- **EYE_AR_CONSEC_FRAMES**

  This value is set to 3 to indicate that three successive frames with an eye aspect ratio less than EYE_AR_THRESH must happen for a blink to be registered.

Tsk 576 (Get The Face Land Marks Using Dlib)

We first load the detector using the g**et_frontal_face_detector(**) and facial landmark predictor dlib.shape_predictor from dlib library.

By using face_utils.FACIAL_LANDMARKS_IDXS we can determine the starting and ending array slice index values for extracting (x, y)-coordinates for both the left and right eye .

Tsk 577 (Capturing the input frames)

There are two ways we can capture the input video:

1.using in-built webcam

2. using video file residing on the disk

we use the VideoStream module to capture a live video.

we use the VideoCapture module in OpenCV to capture a video that is residing on the disk.

Tsk 578 (Converting frames to grayscale channels)

The frame we have captured is a 3-channel RGB colored image. We detect face and eyes in the frame. dlib's face detection works perfectly fine on grayscale images as well as colored images. As a grayscale image is a single channel image, we convert the frame to grayscale to reduce the processing time required by further steps of the algorithm.

Tsk 579 (Calculate Eye Aspect Ratio)

Now let us loop over each of the faces in the frame and then apply facial landmark detection to each of them:

We'll be computing ear value for both the left and right eye, respectively by using the predefined function eye_aspect_ratio(eye).

Computing convex hull for eyes

Compute the convex hull for the left and right eye using cv3 inbuilt function convexHull, then visualize each of the eyes using the drawCountours.

Tsk 579 (Detect Blinks)

The aspect ratio will be approximately constant while the eye is open, and it will quickly fall to zero when a blink occurs. We need to determine the threshold for a blinking ratio that is near to zero. We will assume that every blinking ratio below that threshold will be detected as a blink, and the blinking ratio above the threshold will not be detected.

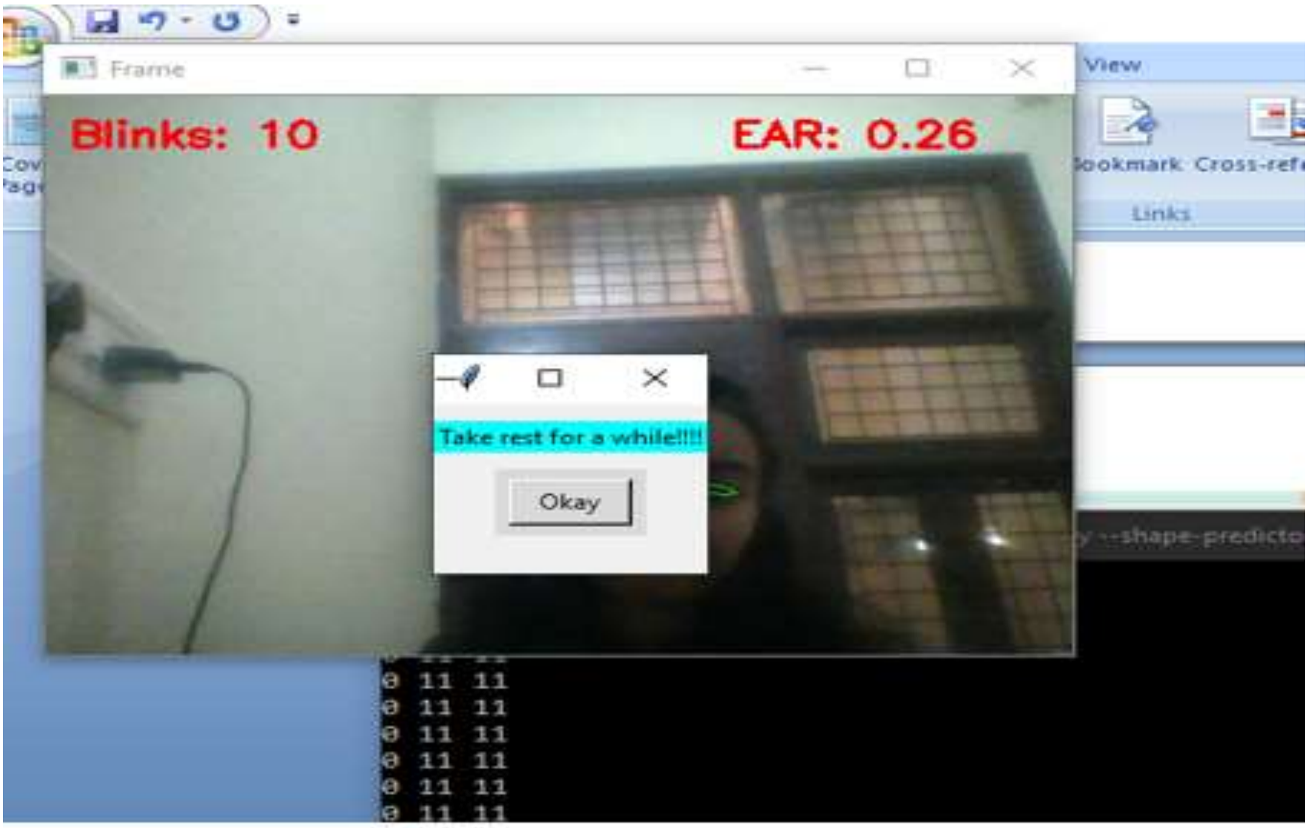
Tsk 579 (Alerting the users)

We’ll be calculating the average number of blinks per minute . If the total blink count is less or more than the average blink count for the stipulated time(calculated for every minute incrementally), then an alert is initiated using audio and popup messages to take rest by calling the playaudio and popup functions.

Tsk 579 (Display the result)

The cv2.putText function displays the number of blink and ear on the OpenCV window once a blink count is detected. The cv2.imshow() function always takes two more functions to load and close the image. These two functions are cv2.waitKey() and cv2.destroyAllWindows(). Inside the cv2.waitKey() function, we can provide any value to close the image and continue with further lines of code.

6.RESULT



Take rest for a while!!!!

```
0 11 11
0 11 11
0 11 11
0 11 11
0 11 11
0 11 11
```

To access the built-in webcam execute the following command in anaconda prompt

python app_eye.py --shape-predictor shape_predictor_68_face_landmarks.dat

To access video file residing on the disk execute the following command in anaconda prompt

python app_eye.py --shape-predictor shape_predictor_68_face_landmarks.dat –video filename.mp3

7.ADVANTAGES AND DISADVANTAGES

ADVANTAGES

During the normal course of a day, a person blinks an average of 15 times a minute to keep the eyes healthy.

Blinking cleans the ocular surface of debris and flushes fresh tears over the ocular surface.

Each blink brings nutrients to the eye surface structures keeping them healthy.

The flow of tears is responsible for wetting the lower third of the cornea.

DISADVANTAGES

Activities that require intense focus and concentration for long periods

- Exposure to bright lights
- Trying to see in dim lighting.
- Needing glasses or an updated glasses prescription
- Poor accommodation (the eye muscles have difficulty focusing on near work) These causes may lead to symptoms like:
- Tired eyes
- Eye pain

8.APPLICATIONS

If you work at a desk and use a computer, these self-care steps can help take some of the strain off your eyes.

Blink often to refresh your eyes.

Many people blink less than usual when working at a computer, which can contribute to dry eyes.

Blinking produces tears that moisten and refresh your eyes.

Reading without pausing to rest your eyes.

Driving long distances and doing other activities involving focusing for a long time.

Being exposed to bright light or glare.

Straining to see in very dim light.

9.CONCLUSION

It has been reported that the normal spontaneous blink rate is between 12 and 15/min. Other studies showed that the interval between blinks ranges from 2.8 to 4 and from 2 to 10 s. A mean blink rate of up to 22 blinks/min has been reported under relaxed conditions.

Most commonly, increased eye blinking results from **eye irritation caused by bright light, dust, smoke, or a foreign body in the eye**. Allergies, infections, and dry eye may also increase the rate of blinking. Conditions of stress, anxiety or fatigue may lead to increased blinking.

10.FUTURE SCOPE

The most important benefit of blinking is that it **cleans the surface of your eye of any debris and washes it out with fresh tears**. The coating of tears helps sharpen your vision, clearing and brightening the image your retina receives.

Blinking is something that we all do, but did you know about the importance of blinking? The truth is, **blinking helps to clean the surface of the eye**. The eye is very easily infected, and if we didn't have the eyelid there to help keep it clean we would develop eye infections much more often

11.BIBILOGRAPHY

- **Anitha J, Mani G, Venkata Rao K. 2020**. Driver drowsiness detection using viola jones algorithm. *Smart Innovation, Systems and Technologies* **159**:583-592

- **Al-Gawwam S, Benaissa M. 2018**. Robust eye blink detection based on eye landmarks and Savitzky-Golay filtering. *Information* **9**(4):93-104

- **Al Tawil L, Aldokhayel S, Zeitouni L, Qadoumi T, Hussein S, Ahamed SS. 2020**. Prevalence of self-reported computer vision syndrome symptoms and its associated factors among university students. *European Journal of Ophthalmology* **30**(1):189-195

- **Awais M, Badruddin N, Drieberg M. 2013**. Automated eye blink detection and tracking using template matching. In: Proceeding - 2013 IEEE student conference on research and development, SCOReD, 2013. PIscataway. IEEE. 79-83

- **Bergasa LM, Nuevo J, Sotelo MA, Barea R, Lopez ME. 2006**. Real-time system for monitoring driver vigilance. *IEEE Transactions on Intelligent Transportation Systems* **7**(1):63-77

- **Cao J, Chen L, Hu D, Dong F, Jiang T, Gao W, Gao F. 2021**. Unsupervised eye blink artifact detection from EEG with gaussian mixture model. *IEEE Journal of Biomedical and Health Informatics* **25**(8):2895-2905

- **Čech J, Franc V, Uřičář M, Matas J. 2016**. Multi-view facial landmark detection by using a 3D shape model. *Image and Vision Computing* **47**:60-70

12.APPENDIX

SOURCE CODE:

FLASK CODE

```
	#python app_eye.py --shape-predictor shape_predictor_68_face_landmarks.dat

# -*- coding: utf-8 -*-

"""

Created on Tue Sep 15 16:08:26 2020



@author: Tulasi

"""

# import the necessary packages

from scipy.spatial import distance as dist

from imutils.video import FileVideoStream

from imutils.video import VideoStream

from imutils import face_utils

import numpy as np

import argparse

import imutils

import time

import dlib

import cv2
import datetime

from gtts import gTTS

import tkinter as tk

from tkinter import ttk

from playsound import playsound



def playaudio(text):

    speech=gTTS(text)

    print(type(speech))

    speech.save("../output1.mp3")

    playsound("../output1.mp3")

    return



LARGE_FONT= ("Verdana", 12)

NORM_FONT = ("Helvetica", 10)

SMALL_FONT = ("Helvetica", 8)



def popupmsg(msg):

    popup = tk.Tk()

    popup.wm_title("Urgent")

    style = ttk.Style(popup)
    style.theme_use('classic')

    style.configure('Test.TLabel', background= 'aqua')

    label = ttk.Label(popup, text=msg,style= 'Test.TLabel')

    label.pack(side="top", fill="x", pady=10)

    B1 = ttk.Button(popup, text="Okay", command = popup.destroy)

    B1.pack()

    popup.mainloop()



def eye_aspect_ratio(eye):

    # compute the euclidean distances between the two sets of

        # vertical eye landmarks (x, y)-coordinates


        A = dist.euclidean(eye[1], eye[5])

        B = dist.euclidean(eye[2], eye[4])


        # compute the euclidean distance between the horizontal

        # eye landmark (x, y)-coordinates

        C = dist.euclidean(eye[0], eye[3])

    # compute the eye aspect ratio
```

```
            ear = (A + B) / (2.0 * C)

    # return the eye aspect ratio

            return ear

 # construct the argument parse and parse the arguments

ap = argparse.ArgumentParser()

ap.add_argument("-p", "--shape-predictor", required=True,

            help="path to facial landmark predictor")

ap.add_argument("-v", "--video", type=str, default="",

            help="path to input video file")

args = vars(ap.parse_args())



# define two constants, one for the eye aspect ratio to indicate

# blink and then a second constant for the number of consecutive

# frames the eye must be below the threshold


EYE_AR_THRESH = 0.3

EYE_AR_CONSEC_FRAMES = 3


#COUNTER  is the total number of successive frames that have an eye aspect ratio

#less than EYE_AR_THRESH
```

```
#TOTAL  is the total number of blinks that have taken place while
# the script has been running

COUNTER = 0
TOTAL = 0

## initializing dlib's face detector (HOG-based) and then create
# the facial landmark predictor

print("[INFO] loading facial landmark predictor...")
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor(args["shape_predictor"])
print(type(predictor),predictor)

# grab the indexes of the facial landmarks for the left and
# right eye, respectively

(lStart, lEnd) = face_utils.FACIAL_LANDMARKS_IDXS["left_eye"]
(rStart, rEnd) = face_utils.FACIAL_LANDMARKS_IDXS["right_eye"]
```

```
eye_thresh = 10

before =datetime.datetime.now().minute



if not args.get("video", False):

    # Taking input from web cam

    print("[INFO] starting video stream...")

    vs = VideoStream(src=0).start()

    time.sleep(1.0)

    #before =datetime.datetime.now().minute



else:

    print("[INFO] opening video file...")

    #Taking input as video file

    vs = cv2.VideoCapture(args["video"])

    time.sleep(1.0)

    #before =datetime.datetime.now().minute



# grab the frame from the threaded video file stream, resize

            # it, and convert it to grayscale

            # channels)
```

```
while True:

    frame = vs.read() #for taking input from web cam

    #_,frame = vs.read() #for taking input from video file

    frame = imutils.resize(frame, width=450)

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # detect faces in the grayscale frame

    rects = detector(gray, 0)

    # loop over the face detections

    for rect in rects:

        shape = predictor(gray, rect)

        shape = face_utils.shape_to_np(shape)

        leftEye = shape[lStart:lEnd]

        rightEye = shape[rStart:rEnd]

        leftEAR = eye_aspect_ratio(leftEye)

        rightEAR = eye_aspect_ratio(rightEye)



        ear = (leftEAR + rightEAR) / 2.0

        # compute the convex hull for the left and right eye, then

                        # visualize each of the eyes

        leftEyeHull = cv2.convexHull(leftEye)

        rightEyeHull = cv2.convexHull(rightEye)
```

```
cv2.drawContours(frame, [leftEyeHull], -1, (0, 255, 0), 1)

cv2.drawContours(frame, [rightEyeHull], -1, (0, 255, 0), 1)

# check to see if the eye aspect ratio is below the blink

                # threshold, and if so, increment the blink frame counter

if ear < EYE_AR_THRESH:

    COUNTER += 1

# otherwise, the eye aspect ratio is not below the blink

                # threshold

else:


    if COUNTER >= EYE_AR_CONSEC_FRAMES:

        TOTAL += 1

    # reset the eye frame counter

    COUNTER = 0


now = datetime.datetime.now().minute

no_of_min = now - before

print(no_of_min, before, now)

blinks = no_of_min * eye_thresh


if(TOTAL < blinks - eye_thresh):
```

```
            playaudio("Take rest for a while as your blink count is less than average count")
            popupmsg("Take rest for a while!!!! :D")
            cv2.putText(frame, "Take rest for a while!!!! :D", (70, 150),
                                    cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
        elif(TOTAL > blinks + eye_thresh):
            playaudio("Take rest for a while as your blink count is more than average count")
            popupmsg("Take rest for a while!!!! :D")
            cv2.putText(frame, "take rest for a while!!!! :D ", (70, 150),
                                    cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)


        cv2.putText(frame, "Blinks: {}".format(TOTAL), (10, 30),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
        cv2.putText(frame, "EAR: {:.2f}".format(ear), (300, 30),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)


    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF
    if key == ord("q"):
        break
cv2.destroyAllWindows()
vs.stop()
```