EYE blink and sleep detection using OpenCv and python drivers for safety

OPenCV

- Library to process image and used in computer vision

Eye Game

- Library to process image get the Eye location and movement

PYTHON

Most Commonly Used Library.

Face Recognition and Eye detection

File Edit Format Run Options Window Help

```
import face_recognition
import cv2
import numpy as np
import time
from espeak import espeak
import cv2

import eye_game
previous ="unkno"
count=0
video_capture = cv2.VideoCapture(0)

#frame = (video_capture, file)
file = 'image_data/image.jpg'

# Load a sample picture and learn how to recognize it.
ash_image = face_recognition.load_image_file("ash.jpg")
ash_face_encoding = face_recognition.face_encodings(ash_image)[0]


# Create arrays of known face encodings and their names
```

Video Capture and process process

```
#frame = (video_capture, file)
file = 'image_data/image.jpg'

# Load a sample picture and learn how to recognize it.
ash_image = face_recognition.load_image_file("ash.jpg")
ash_face_encoding = face_recognition.face_encodings(ash_image)[0]


# Create arrays of known face encodings and their names
known_face_encodings = [
    ash_face_encoding


]
known_face_names = [
    "Ashwini kumar"
]

# Initialize some variables
face_locations = []
face_encodings = []
```

```
process_this_frame = True

while True:
    # Grab a single frame of video

    ret, frame = video_capture.read()

    # Resize frame of video to 1/4 size for faster face recognition processing
    small_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)



    # Convert the image from BGR color (which OpenCV uses) to RGB color (which f
    rgb_small_frame = small_frame[:, :, ::-1]

    # Only process every other frame of video to save time
    if process_this_frame:
        # Find all the faces and face encodings in the current frame of video
        face_locations = face_recognition.face_locations(rgb_small_frame)
        face_encodings = face_recognition.face_encodings(rgb_small_frame, face_l
        cv2.imwrite(file, small_frame)

        face_names = []
        for face_encoding in face_encodings:
            # See if the face is a match for the known face(s)
            matches = face_recognition.compare_faces(known_face_encodings, face_
            name = "Unknown"


            # # If a match was found in known_face_encodings, just use the first
            # if True in matches:
            #     first_match_index = matches.index(True)
            #     name = known_face_names[first_match_index]

            # Or instead, use the known face with the smallest distance to the n
            face_distances = face_recognition.face_distance(known_face_encodings
            best_match_index = np.argmin(face_distances)
            if matches[best_match_index]:
```

```
            # # If a match was found in known_face_encodings, just use the first
            # if True in matches:
            #     first_match_index = matches.index(True)
            #     name = known_face_names[first_match_index]

            # Or instead, use the known face with the smallest distance to the n
            face_distances = face_recognition.face_distance(known_face_encodings
            best_match_index = np.argmin(face_distances)
            if matches[best_match_index]:
                name = known_face_names[best_match_index]

                direction= eye_game.get_eyeball_direction(file)
                print(direction)
                #eye_game.api.get_eyeball_direction(cv_image_array)
                if previous != direction:
                    previous=direction


                else:
                    print("old same")
                    count=1+count
                    print(count)
                    if (count>=30):
                        espeak.synth("Are You sleeping Start Driving")


            face_names.append(name)


    process_this_frame = not process_this_frame


    # Display the results
    for (top, right, bottom, left), name in zip(face_locations, face_names):
        # Scale back up face locations since the frame we detected in was scaled
        top *= 4
```

Thank You