SCHOOL OF COMPUTER SCIENCE

DROWSINESS DETECTION AND ALERT SYSTEM

TEAM MEMBERS :

MALLENI SAI PRANEETH(18BCE0242)

LAGUMSANI CHARAN (18BCE0365)

D HEMANTH (18BCI0106)

Project report

of

IMAGE PROCESSING

WINTER SEM 2020-21

Submitted to

Faculty: Prof. SWATHI JN Signature:

Slot: E1+TE1

Introduction:

Drowsiness is a condition which diminishes the awareness caused by the absence of rest or weariness. Because of weariness, driver lose their control which may divert them from the street and prompts serious mishaps. Basically, these mishaps may because of the driver tired condition. Driving continuosly for quite a while prompts tiredness and make them to lost their awareness. Due to substantial increment in the number of mischances day by day which makes a major issues.Therefore, a model is required to keep the driver in the engaged state. The goal of our model is to make a driver weariness location which demonstrates driver's lethargic condition through their face.The thought is to build up the model that will persistently recognize the driver's exhaustion

progressively. One of the identification procedure is Percentage of squinting of eyes. Here we utilize a system totally in view of the development of eyes. The perception of eye development is essentially in view of the driver's condition. On the off chance that the individual is expending liquor it can be effortlessly identified by alcoholic sensors and if any temperature misused it can be recognized by temperature sensors. The spillage of gas can likewise be distinguished by the gas sensors. In request to identify the driver's laziness different methods has been implemented. But no strategy has been actualized to stop the auto naturally if the weakness or sluggish mode is identified. The most widely recognized strategy to distinguish the driver's state is of utilizing face detection. primarily focuses on the edges of the face through which the weakness condition is obtained. tiredness, stress, consumption of alcohol, mobile utilization.

Problem Identification:

A majority of road related accidents are due to driver’s fatigue, distraction and drowsiness. Recently accidents are increased at large amount. Various new creative technologies are introduced to avoid and reduce these accidents. In these accidents, driver can meet severe injuries and even death. These may lead to significant economic loss. This system can reduce the road related accidents due to driver’s fatigue. The most widely recognized strategy to distinguish the driver's state is of utilizing face detection primarily focuses on the edges of the face through

which the weakness condition is obtained tiredness, stress, consumption of alcohol, mobile

utilization etc. Our main objective is to develop a model which can detect the driver's drowsiness and indicate the driver's condition by alarm. Here we provide a prevention technique using eye blink and where we can stop the vehicle to avoid collision of vehicle. In the existing system, the drowsiness is detected by the SVM (Support Vector Machine). This SVM classifies a sequence of video segments into alert or non-alert driving event. To overcome the drawbacks of the existing system, the proposed system implements, Eye aspect ratio (EAR) algorithm for drowsiness detection is used.

Objective:

Most of the road accidents are due to driver's fatigue, distraction and drowsiness. Various new creative technologies are introduced to avoid and reduce these accidents. This system can reduce the road related accidents due to drowsiness. The drivers face will be detected, focusing on the edges of the face through which the weakness condition is obtained tiredness, stress, consumption of alcohol, etc. The eye is monitored by the webcam and the data is sent and processed and the signal is sent to the alarm system. On being detected the alarm starts to go off alerting the driver who might be drowsy or distracted and alongside a message is sent to the drivers phone number and also to two of his or her kins.

Literature Review:

The first approach was to study the physiological features such as brain wave (EEG wave), heart rate. Later they used ORD model which accounts the physical appearance, behavior, mannerisms. Recently came the HMM and SVM. HMM uses the observations to get the states. It can be used for time series classification if the given time series data is an output of a state-machine system. HMM is a powerful modeling technique when the system states are partially observable and the behavior of the system is considered as autonomous.

	Title of the journal paper	Name of the journal paper	Author	Year	Abstract
1.	Support Vector Machine Based Detection of Drowsiness Using Minimum EEG Features	IEEE	Shaoda Yu ; Peng Li ; Honghuang Lin ; Ehsan Rohani ; Gwan Choi ; Botang Shao ; Qian Wang	2013	While there is a body of work on drowsiness detection using EEG signals in neuroscience and engineering, there exist unanswered questions pertaining to the best mechanisms to use for detecting drowsiness. Targeting a range of practical safety-awareness applications, this study adopts a machine learning based approach to build support vector machine (SVM) classifiers to distinguish between awake and drowsy states. While broadband alpha, beta, delta, and theta waves are often used as features in the existing work, lack of widely agreed precise definitions of such broadband signals and difficulty in accounting for interpersonal variability has led to poor classification performance as demonstrated in this study.
2.	Facial Features Monitoring for Real Time Drowsiness Detection	2016 12th International Conference on Innovations in Information Technology (IIT) Al Ain, UAE, At Al Ain, UAE	Manu Bn	2016	This paper describes an efficient method for drowsiness detection by three well defined phases. These three phases are facial features detection using Viola Jones, the eye tracking and yawning detection. Once the face is detected, the system is made illumination invariant by segmenting the skin part alone and considering only the chromatic components to reject most of the non face image backgrounds based on skin color. The tracking of eyes and yawning detection are done by correlation coefficient template matching. The feature vectors from each of the above phases are concatenated and a binary linear support vector machine classifier is used to classify the consecutive frames into fatigue and nonfatigue states and sound an alarm for the former, if it is above the threshold time. Extensive real time experiments prove that the proposed method is highly efficient in finding the drowsiness and alerting the driver.

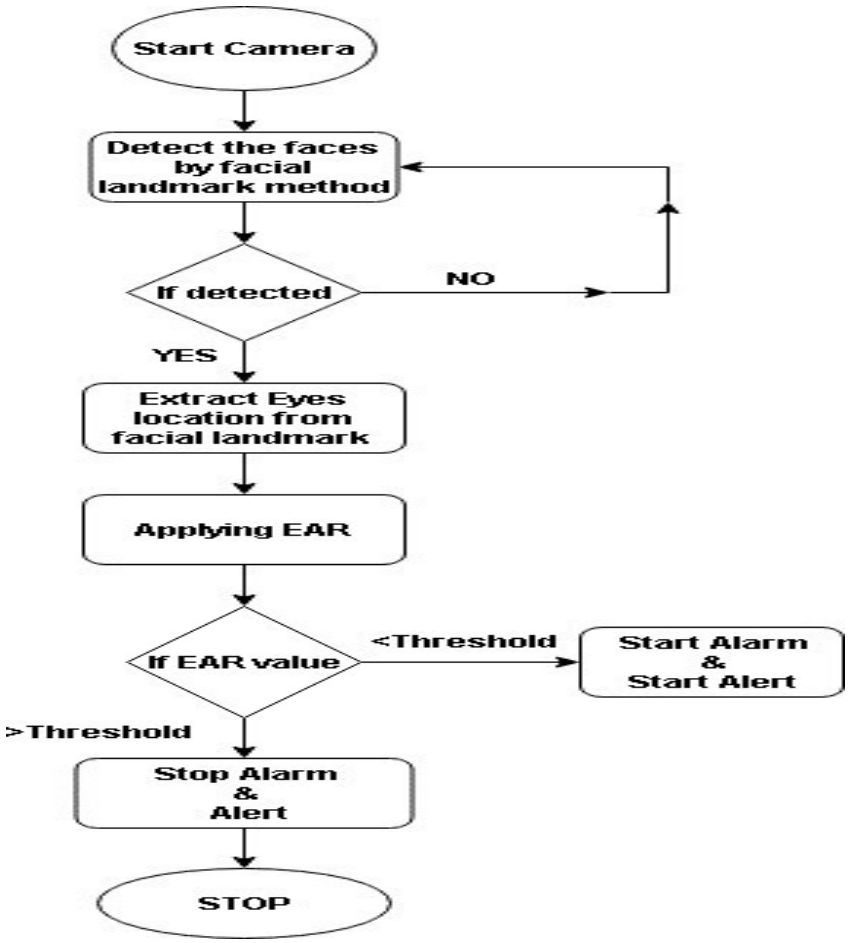
3.	Driver fatigue monitoring system using Support Vector Machines	Communications Control and Signal Processing (ISCCSP), 2012 5th International Symposium	Matthew Sacco, Reuben A Farrugia	2012	Driver fatigue is one of the leading causes of traffic accidents. This paper presents a real-time nonintrusive fatigue monitoring system which exploits the driver's facial expression to detect and alert fatigued drivers. The presented approach adopts the Viola-Jones classifier to detect the driver's facial features. The correlation coefficient template matching method is then applied to derive the state of each feature on a frame by frame basis. A Support Vector Machine (SVM) is finally integrated within the system to classify the facial appearance as either fatigued or otherwise. Using this simple and cheap implementation, the overall system achieved an accuracy of 95.2%, outperforming other developed systems employing expensive hardware to reach the same objective.using MQTT protocol.
4.	Detection and prediction of driver drowsiness using artificial neural network models	Science Direct	Charlotte Jacobe de Naurois , Christophe Bourdin , Anca Stratulat, Emmanuelle Diaz , Jean Louis Vercher	2017	Not just detecting but also predicting impairment of a car driver’s operational state is a challenge. This study aims to determine whether the standard sources of information used to detect drowsiness can also be used to predict when a given drowsiness level will be reached. Moreover, we explore whether adding data such as driving time and participant information improves the accuracy of detection and prediction of drowsiness. Twenty-one participants drove a car simulator for 110 min under conditions optimized to induce drowsiness. We measured physiological and behavioral indicators such as heart rate and variability, respiration rate, head and eyelid movements (blink duration, frequency and PERCLOS) and recorded

					driving behavior such as time-to-lane-crossing, speed, steering wheel angle, position on the lane. Different combinations of this information were tested against the real state of the driver, namely the ground truth, as defined from video recordings via the Trained Observer Rating. Two models using artificial neural networks were developed, one to detect the degree of drowsiness every minute, and the other to predict every minute the timerequired to reach a particular drowsiness level (moderately drowsy). The best performance in both detection and prediction is obtained with behavioral indicators and additional information. The model can detect the drowsiness level with a mean square error of 0.22 and can predict when a given drowsiness level will be reached with a mean square error of 4.18 min.
5.	The Steps of proposed drowsiness detection system design based on image processing in simulator driving	Internationa l Campus (TUMS-IC) of Tehran University of Medical Sciences	Mohsen Karchani , Adel Mazloumi , Gebraeil Nasl Saraji , Ali Nahvi	2015	Drowsiness detection has many implications including reducing roads traffic accidents importance. Using image processing techniques is amongst the new and reliable methods in sleepy face. The present pilot study was done to investigate sleepiness and providing images of drivers' face, employing virtual-reality driving simulator.In order to detecting level of sleepiness according to the signal, information related to 25 drivers was recorded with imaging rate of 10 fps. Moreover, on average 3000 frames was analysed for each driver. The frames were investigated by transforming in grey scale space and based on the Cascade and Viola & Jones techniques and the images characteristics were extracted using Binary

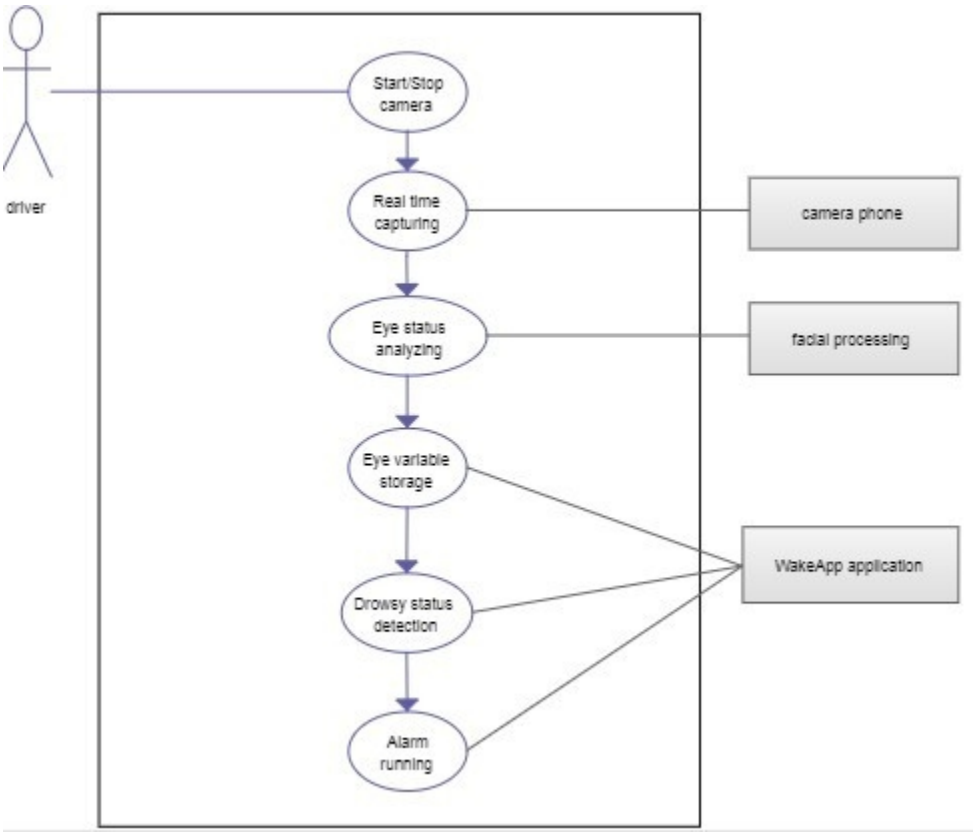
					and Histogram methods. The MPL neural network was applied for analysing data.70% of information related to each driver were inserted to the network of which 15% for test and 15% for validation. In the last stage the accuracy of 93% of the outputs were evaluated. The intelligent detection and usage of various criteria in longterm time frame are of the advantages of the present study, comparing to other researches. This is helpful in early detection of sleepiness and prevents the irrecoverable losses by alarming.

Detailed Methodology:

Flow Chart:



Experimental Design:



Here the wakeApp application is cloud application controlled byThingSpeak.

Alarm will ring to wake up the driver. Simultaneously an alert message will be sent to whom so ever concerned number.

Algorithm Development:

In the proposed system, Eye aspect ratio (EAR) algorithm for drowsiness detection is used. The EAR is mostly constant when an eye is open and is getting close to zero while closing an eye. It is partially person and head pose insensitive. Aspect ratio of the open eye has a small variance among individuals and it is fully invariant to a uniform scaling of the image and inplane rotation of the face. Since eye blinking is performed by both eyes synchronously, the EAR of both eyes is averaged.

It generally does not hold that low value of the EAR means that a person is blinking. A low value of the EAR may occur when a subject closes his/her eyes intentionally for a longer time or performs a facial expression, yawning, etc., or the EAR captures a short random fluctuation of the landmarks. Therefore, we propose a classifier that takes a larger temporal window of a frame as an input. For the 30fps videos, we experimentally found that ±6 frames can have a significant impact on a blink detection for a frame where an eye is the most closed when blinking. Thus, for each frame, a 13- dimensional feature is gathered by concatenating the EARs of its ±6 neighboring frames.

Standards Adopted:

"IEEE Standard Glossary of Image Processing and Pattern Recognition Terminology," in IEEE Std 610.4-1990 , vol., no., pp.0_1-, 1990, doi: 10.1109/IEEESTD.1990.94600.

Trade off Identified:

While developing the system, we found that due to the light intensity of the video captured there is possibility of glare for the users with spectacles. Though the eye is detected and the contour is formed around the eye, it is difficult for the camera to identify whether the person's eyes are being shut or not as the shadow falling on the glass ie the glare causes a hindrance. It is not possible for us to suggest the customer to use a particular type of spectacles for our system to work, so this is one major trade-off we have made.

Algorithm Implementation:

Eye blinking is a fast closing and reopening of a human eye. Each individual has a little bit different pattern of blinking. The pattern differs in the speed of closing and opening, a degree of squeezing the eye and in a blink duration. The eye blink lasts approximately 100-400 ms. We propose to exploit state-of-the-art facial landmark detectors to localize the eyes and eyelid contours. From the landmarks detected in the image, we derive the eye aspect ratio (EAR) that is used as an estimate of the eye opening state. Since the perframe EAR may not necessarily recognize the eye blinks correctly, a classifier that takes a larger temporal window of a frame into account is trained.

For every video frame, the eye landmarks are detected. The eye aspect ratio (EAR) between height and width of the eye is computed.

$$\mathrm{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|},$$

,

where p1, . . . , p6 are the 2D landmark locations of eye.

The EAR is mostly constant when an eye is open and is getting close to zero while closing an eye.

It is partially person and head pose insensitive. Aspect ratio of the open eye has a small variance among individuals and it is fully invariant to a uniform scaling of the image and inplane rotation of the face. Since eye blinking is performed by both eyes synchronously, the EAR of both eyes is averaged.

It generally does not hold that low value of the EAR means that a person is blinking. A low value of the EAR may occur when a subject closes his/her eyes intentionally for a longer time or performs a facial expression, yawning, etc., or the EAR captures a short random fluctuation of the landmarks. Therefore, we propose a classifier that takes a larger temporal window of a frame as an input. For the 30fps videos, we experimentally found that ±6 frames can have a significant impact on a blink detection for a frame where an eye is the most closed when blinking. Thus, for each frame, a 13-dimensional feature is gathered by concatenating the EARs of its ±6 neighboring frames

We are going to use dlib and OpenCV to detect *facial landmarks* in an image. OpenCV is a library of programming functions mainly aimed at real-time computer vision.

Facial landmarks are used to localize and represent salient regions of the face, such as:

Eyes, Eyebrows, Nose, Mouth, Jawline

Facial landmarks have been successfully applied to face alignment, head pose estimation, face swapping, blink detection and much more. The pre-trained facial landmark detector inside the dlib library is used to estimate the location of *68 (x, y)-coordinates* that map to facial structures on the face.

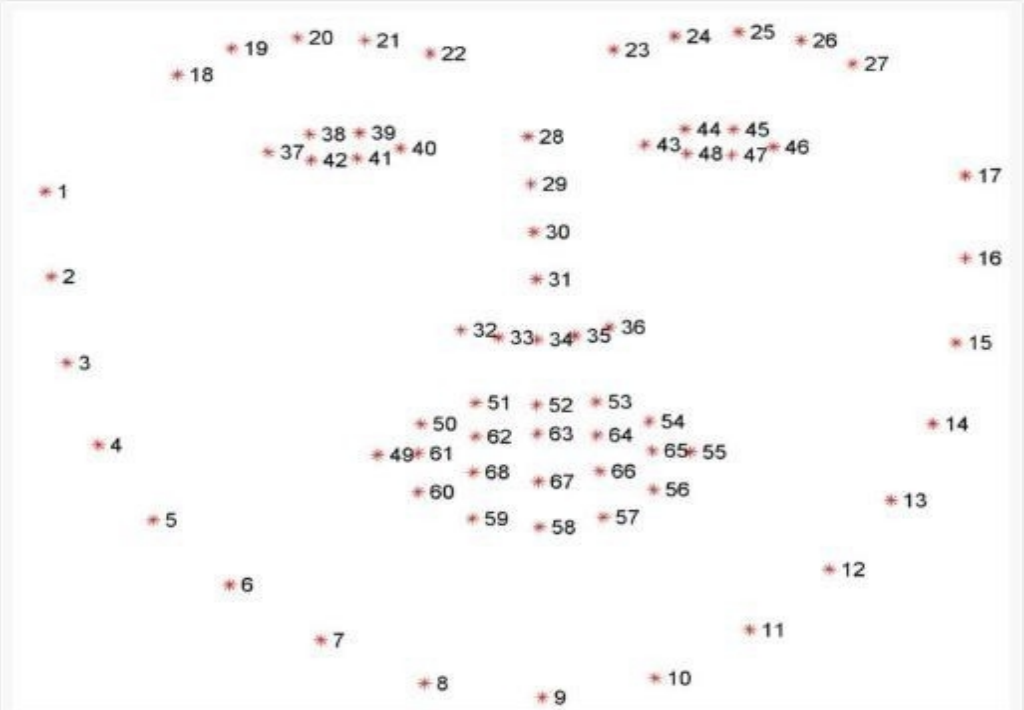
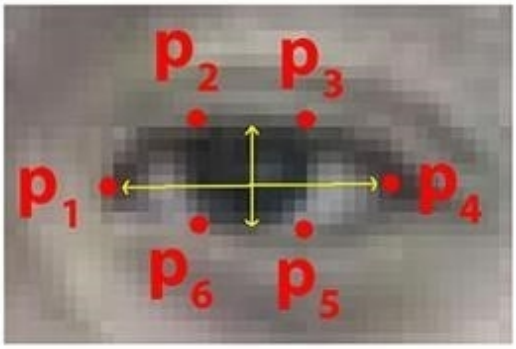


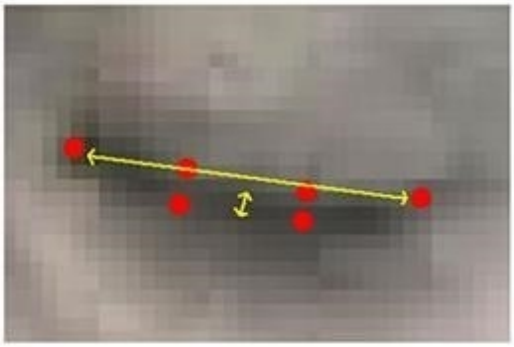
Figure 2: Visualizing the 68 facial landmark coordinates from the iBUG 300-W dataset (higher resolution).

Firstly, setting up the camera that monitors a stream of faces.

If a face is found, we apply facial landmark detection and extract the eye regions. Now that we have the eye regions, we can compute the eye aspect ratio to determine if the eyes are closed.

If the eye aspect ratio indicates that the eyes have been closed for a sufficiently long enough amount of time, we'll sound an alarm to wake up the driver.





Continuosly the EAR will be compared with threshold ,and raise an alarm if EAR is less than threshold for considerable time.

$$\mathrm{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|},$$

b) Designing of Web page:

How to setup ThingSpeak:

ThingSpeak requires a user account and a channel. A channel is where you send data and where ThingSpeak stores data. Each channel has up to 8 data fields, location fields, and a status field. You can send data every 15 seconds to ThingSpeak, but most applications work well every minute.

- Sign up for new User Account – https://thingspeak.com/users/sign_up
- Create a new Channel by selecting Channels, My Channels, and then New Channel
- Note the Write API Key and Channel ID

Demonstration of Result:

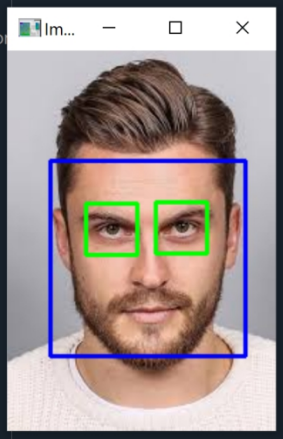
The drowsiness has been detected using the eye aspect ratio and an alert system has been generated.
Data is continuously sent and retrieved from cloud
Alert message is sent using Message 91 API.

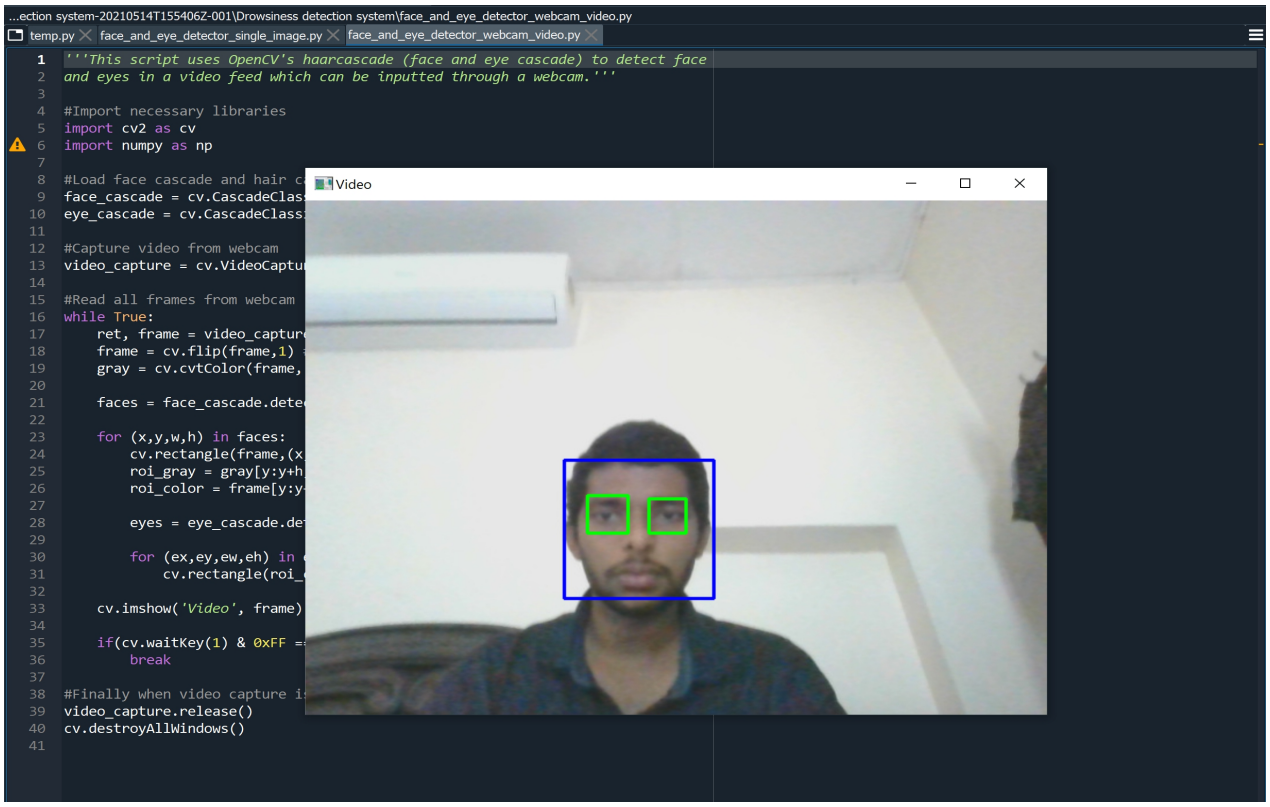
Code /results :

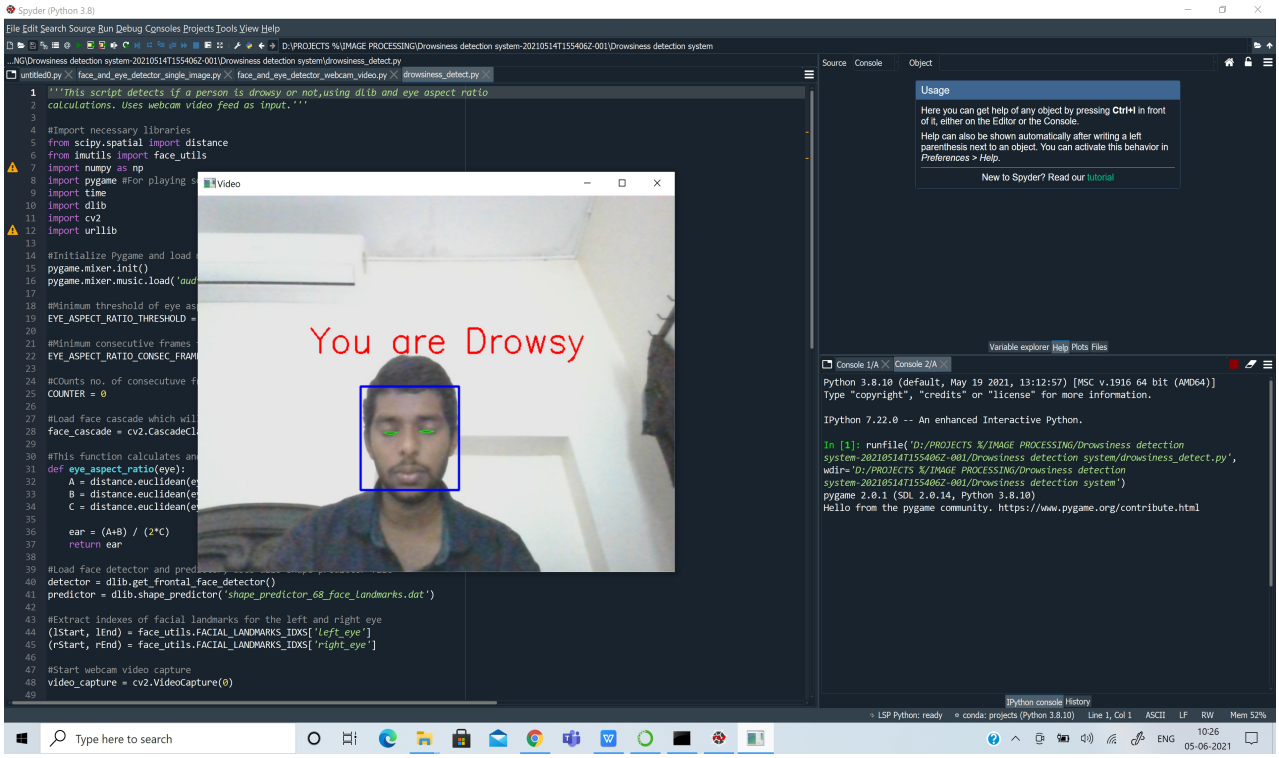
...ection system-20210514T155406Z-001\Drowsiness detection system\face_and_eye_detector_single_image.py

temp.py | face_and_eye_detector_single_image.py

```python
'''This script uses OpenCV's haarcascade (face and eye cascade) to detect face
and eyes in a given input image.'''

#Import necessary libraries
import cv2 as cv
import numpy as np

#Load face cascade and hair cascade from haarcascades folder
face_cascade = cv.CascadeClassifier("haarcascades/haarcascade_frontalface_default.xml")
eye_cascade = cv.CascadeClassifier("haarcascades/haarcascade_eye.xml")

#Read image in img and convert it to grayscale and store in gray.
#Image is converted to grayscale, as face cascade doesn't require to operate o
img = cv.imread('images/test.jpeg')
gray = cv.cvtColor(img, cv.COLOR_BGR2GRAY)

#Detect all faces in image.
faces = face_cascade.detectMultiScale(gray, 1.3, 5)

#Draw a rectangle over the face, and detect eyes in faces
for (x,y,w,h) in faces:
    cv.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)

    #ROI is region of interest with area having face inside it.
    roi_gray = gray[y:y+h, x:x+w]
    roi_color = img[y:y+h, x:x+w]

    #Detect eyes in face
    eyes = eye_cascade.detectMultiScale(roi_gray)

    for (ex,ey,ew,eh) in eyes:
        cv.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)

cv.imshow('Image', img)
cv.waitKey(0)
cv.destroyAllWindows()
```




...ection system-20210514T155406Z-001\Drowsiness detection system\face_and_eye_detector_webcam_video.py
temp.py
face_and_eye_detector_single_image.py
face_and_eye_detector_webcam_video.py
'''This script uses OpenCV's haarcascade (face and eye cascade) to detect face and eyes in a video feed which can be inputted through a webcam.'''
#Import necessary libraries
import cv2 as cv
import numpy as np
#Load face cascade and hair c
face_cascade = cv.CascadeClas
eye_cascade = cv.CascadeClass
#Capture video from webcam
video_capture = cv.VideoCaptu
#Read all frames from webcam
while True:
ret, frame = video_captur
frame = cv.flip(frame,1)
gray = cv.cvtColor(frame,
faces = face_cascade.dete
for (x,y,w,h) in faces:
cv.rectangle(frame,(x
roi_gray = gray[y:y+h
roi_color = frame[y:y
eyes = eye_cascade.de
for (ex,ey,ew,eh) in
cv.rectangle(roi_
cv.imshow('Video', frame)
if(cv.waitKey(1) & 0xFF =
break
#Finally when video capture i
video_capture.release()
cv.destroyAllWindows()
Video


Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
'''This script detects if a person is drowsy or not,using dlib and eye aspect ratio calculations. Uses webcam video feed as input.'''
#Import necessary libraries
from scipy.spatial import distance
from imutils import face_utils
import numpy as np
import pygame #For playing s
import time
import dlib
import cv2
import urllib
#Initialize Pygame and load
pygame.mixer.init()
pygame.mixer.music.load('aud
#Minimum threshold of eye as
EYE_ASPECT_RATIO_THRESHOLD =
#Minimum consecutive frames
EYE_ASPECT_RATIO_CONSEC_FRAM
#COunts no. of consecutuve f
COUNTER = 0
#Load face cascade which wil
face_cascade = cv2.CascadeCl
#This function calculates an
def eye_aspect_ratio(eye):
A = distance.euclidean(e
B = distance.euclidean(e
C = distance.euclidean(e
ear = (A+B) / (2*C)
return ear
#Load face detector and pred
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor('shape_predictor_68_face_landmarks.dat')
#Extract indexes of facial landmarks for the left and right eye
(lStart, lEnd) = face_utils.FACIAL_LANDMARKS_IDXS['left_eye']
(rStart, rEnd) = face_utils.FACIAL_LANDMARKS_IDXS['right_eye']
#Start webcam video capture
video_capture = cv2.VideoCapture(0)
Video
You are Drowsy
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
New to Spyder? Read our tutorial
Python 3.8.10 (default, May 19 2021, 13:12:57) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.
IPython 7.22.0 -- An enhanced Interactive Python.
In [1]: runfile('D:/PROJECTS %/IMAGE PROCESSING/Drowsiness detection system-20210514T155406Z-001/Drowsiness detection system/drowsiness_detect.py', wdir='D:/PROJECTS %/IMAGE PROCESSING/Drowsiness detection system-20210514T155406Z-001/Drowsiness detection system')
pygame 2.0.1 (SDL 2.0.14, Python 3.8.10)
Hello from the pygame community. https://www.pygame.org/contribute.html
Type here to search
ENG
10:26
05-06-2021

Conclusion :

This task speaks to a case of methodical way to deal with the appraisal of wearable sensors for physiological parameter estimation. By this undertaking driver's laziness is checked persistently. In a bad position, they will be frightened and display effortlessly. In order to keep away from street mishaps outlining driver sluggishness location framework will be absolutely productive. In this way it is presumed that eye checking framework is one of the quick technique for influencing driver to alarm if sluggishness orehaustion is experienced while driving.. The framework utilizes a blend of layout based coordinating so as to restrict the eyes. Amid following, framework will have the capacity to choose if the eyes are open or shut and whether the driver is looking in front. At the point when the eye will be shut for a really long time, a notice flag will be given as bell or caution pack message.

REFERENCES:

[1] Rosebrock. (2017, Apr. 3*). Facial landmarks with dlib, OpenCV, and Python* [Online]. Available:

[2] A T. Soukupova and J. Cech. (2016, Feb. 3) *Real-Time Eye Blink Detection using Facial Landmarks*

[3]*V.* Kazemi and J. Sullivan. (2014) *One Millisecond Face Alignment*.