NAME : HARSH SHAH

------------------------------ COMPUTER VISION (CV)----------------------------------

The term Computer Vision (CV) is used and heard very often in artificial intelligence (AI) and deep learning (DL) applications. The term essentially means giving a computer the ability to see the world as we humans do.

Computer Vision is a field of study which enables computers to replicate the human visual system. As already mentioned above, It’s a subset of artificial intelligence which collects information from digital images or videos and processes them to define the attributes. The entire process involves image acquiring, screening, analysing, identifying and extracting information. This extensive processing helps computers to understand any visual content and act on it accordingly.

TASK-2 :

--problem statement-- :

COMPUTER VISION

- Given a dataset of human face images taken from a dashboard of a car, detect if the driver is drowsy or not. Note that a driver can be classified as drowsy if their eyes are closed more often than open.
    - Describe the preprocessing steps required on the image (if any).
    - Describe techniques you would use to detect the eyes in the entire image.
    - Describe techniques you would use to detect drowsiness in the eyes.
    - End the synopsis with a conclusion which should contain information about why you selected a particular algorithm over other existing solutions.

SYNOPSIS :

PREPROCCESING ON IMAGE :

The detection works only on grayscale images. So it is important to convert the color image to grayscale.The following images taken from the dataset are on grayscale . therefore while scanning the image we will use “cv2.imread(“img-name”,0). The zero represents that the the image has been scanned in grayscale form which takes image in shades of grey , black and white . if the images present in the dataset are coloured , then it can be read normally as “cv2.imread(“file-name”). and then be converted to grayscale by function “cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)”, which convertes the image to greyscale form .

PROCESS OF FACE/EYE DETECTION :

At this stage we are going to use what is known as " haar cascades". So what are haar cascades ?. haar cascade classifiers are an effective way for object detection . it is a machine learning based approach where the classifier is pretrained with lot of positive and negative images .

Positive images – the set of images which we want our classifier to identify

Negative images-the set of images which are vague and we don't want them to be recognised by our classifier

The haar cascade files can be imported from the opencv github repository

Let suppose we take a variable face_cascade , then using the frontal face detection xml file of haar cascade we are highlighting the face as our desired region of interest ("roi"). This is because we are certain that eyes are present in face and not somewhere else . (Atleast for beings on this planet)

Then using function cv2.rectangle , we draw a rectangle using the coordinate array which we obtained from the cascade function , the variable faces stores the with the model imported. It a list of coordinates for all the faces found in the photo. We iterate on each of the entries in the coordinate list, it contains the x and the y coordinates and the width and the height. Then we create a rectangle with these values and fill in with a random color. We then display the image with the face coordinates in the colored rectangles.

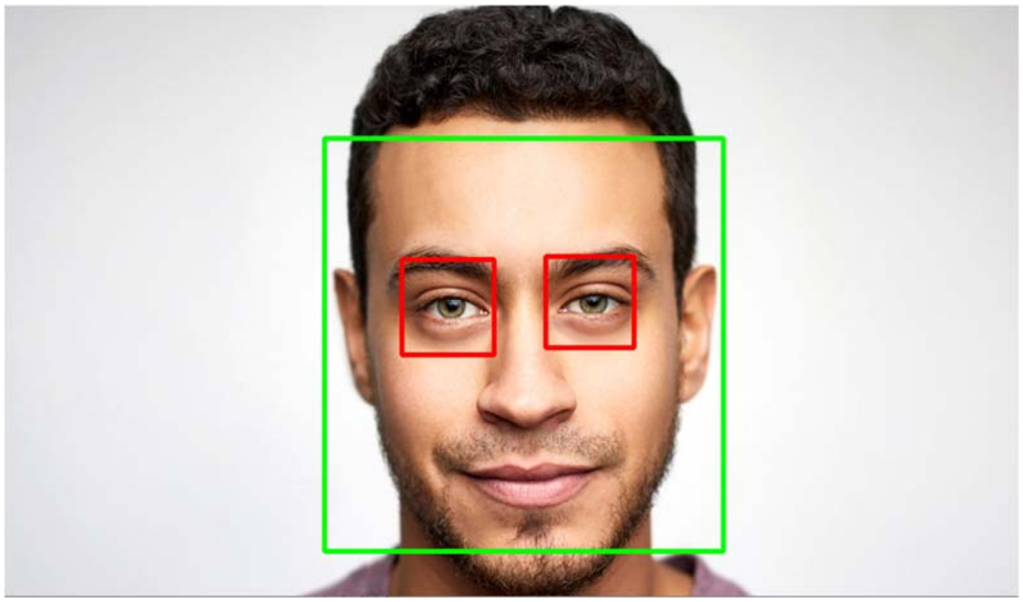
Now that we have determined our region of interest-1. Now we can simply find the coordinates of eyes in the second frame. As the model will be detecting the eye coordinates within a face. This also produces a list of coordinates for the eyes found. The coordinates again contain the x and y coordinates, and the width and the height.

As we iterate over the eyes list , we plot each point of the eye coordinate as a rectangle around both the eyes . it looks like :

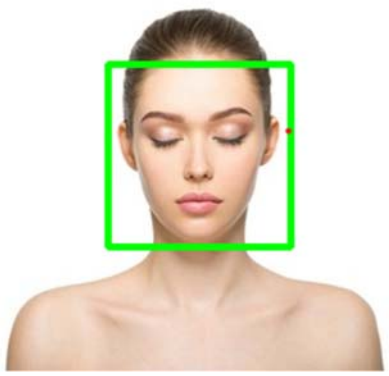
DETECTION OF DROWSINESS :

We don't have to do anything else particularly to detect the drowsiness in the eyes, this is because eye detection using cascade files is only applicable to open eyes. Therefore the closed eyes aren't detected. That is the list containing coordinates of the eye is empty . since we know it is empty we can tell that if the list is empty , we can increment the variable storing the count of closed eye photos. Therefore in the given dataset if the number of images which have closed eyes more than open eyes then we can say that the driver is drowsy otherwise he is just blinking normally !

When eyes are open the output is like :



WHEN EYES ARE CLOSED , OUTPUT IS LIKE :



--- **CONCLUSION** ---

Ive chosen the cascade files method , this is mainly because of the reason that the code becomes pretty precise also , we don’t need any separate function to detect closed eyes as the cascade files automatically rejects closed eyes. I am not so familiar neither did I understand other methods so well .they use shape “eye_aspect_ratio()” function to Loop over each of the faces in the frame and then apply facial landmark detection to each of them

Then we use convexhull to draw the shape of the eyes. This method is more precise as we draw the outline around the eyes. In my method we plot a rectangle to highlight the region . Also cascade file method was the first method I encountered , so I used it .