Drowsiness Detection Automated System for Vehicles using Artificial Intelligence

Abstract:-

India is one of a country where accidents happen frequently. Mostly, in night on highways, Drivers feel drowsiness. This computer vision system detects driver's drowsiness and generates alarm. This can be very vital to avoid such accidents. I have developed a model which counts the no. of blinks and later based on a metric "EAR" generates the Alarm. This system is capable of detecting drowsiness, counting the blinks and later generating the alarm using facial landmarks and Open CV. To develop our detector, I am computing a metric called Eye Aspect Ratio. The System is developed using Python, Open CV, dlib and Scipy. A web cam will be taking the live real video stream of the driver and by the help of Open CV; it does the processing on further algorithms.

Introduction:-

Driving when you are sleepy & exhausted? The effects of drowsiness are similar to alcohol - it will make driving inputs (steering, acceleration, braking) poorer, destroy your reaction times & blur thought processes. he study, which examined the perceptions of sleepy driving and drunk driving of 114 young drivers (under 30) and 177 drivers over 30, found young drivers were more likely to drive sleepy than drunk and more accepting of enforcement practices for drink driving than they are for sleepy driving.

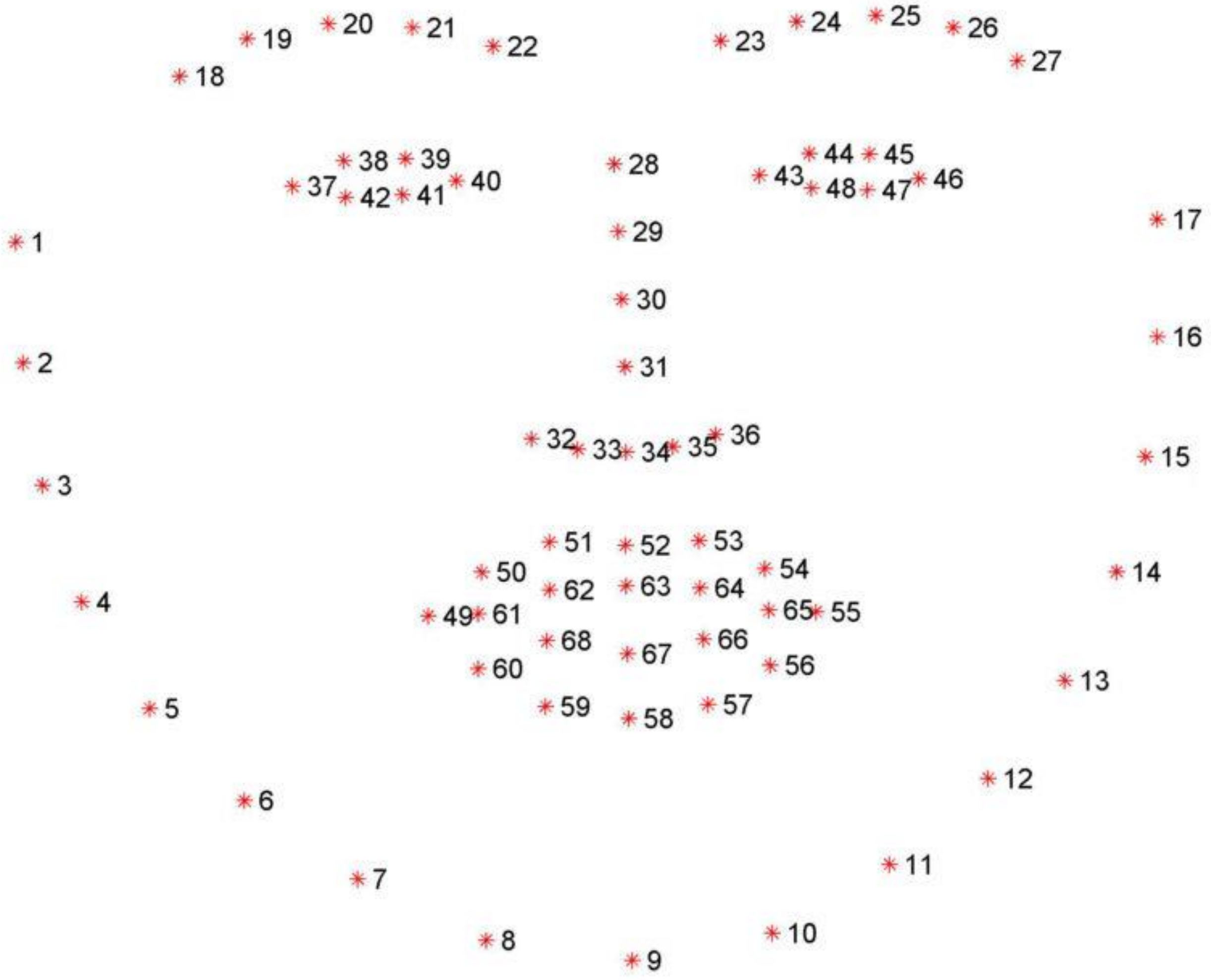
“Research shows a blood alcohol content (BAC) of 0.05 per cent has the same effect as being awake for 17 hours, and a BAC of 0.1 percent is roughly 20 hours, but drivers don’t consider the impairment to be the same,” Watling said.

Around 5,000 People Were Killed Last Year Due to Drowsy Driving. Now, by the help of technology advancement, it is possible to detect the drowsiness and avoid such accidents by generating alarms. Integrating a web cam in front of Driver is what can solve the half of the problem. This AI based model is very simple and fast to solve the problem by the help of Open CV which is a library used for developing Computer Vision’s applications.

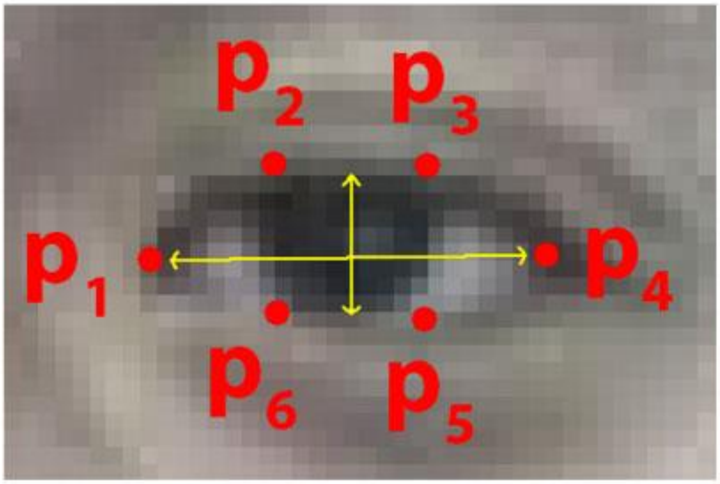
There are other tools like YOLO and COCO which is also used in Computer Vision for the same purpose but I am using Open CV for its speed and simplicity.

Learning Algorithms

I applied facial landmark detection to localize important regions of the face, including eyes, eyebrows, nose, ears, and mouth.



For detecting the drowsiness, there is only one thing which matters i.e. EYE. Each eye is represented by 6 *(x, y)*-coordinates, starting at the left-corner of the eye (as if you were looking at the person), and then working clockwise around the remainder of the region:



Based on this image, we can notice of the following things:

There is a relation between the *width* and the *height* of these coordinates.

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

Where *p1…..p6* are 2D facial landmark locations.

The numerator of this equation computes the distance between the vertical eye landmarks while the denominator computes the distance between horizontal eye landmarks, weighting the denominator appropriately since there is only *one* set of horizontal points but *two* sets of vertical points.

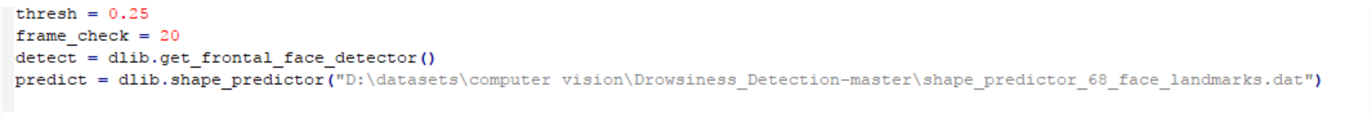
The EAR is constant while the eye is open, but will suddenly fall to zero when a blink is taking place.

Tools and Techniques

The following tools are used to develop this model as follows:-

1. **cv2**
2. **immutils**
3. **dlib**
4. **scipy**

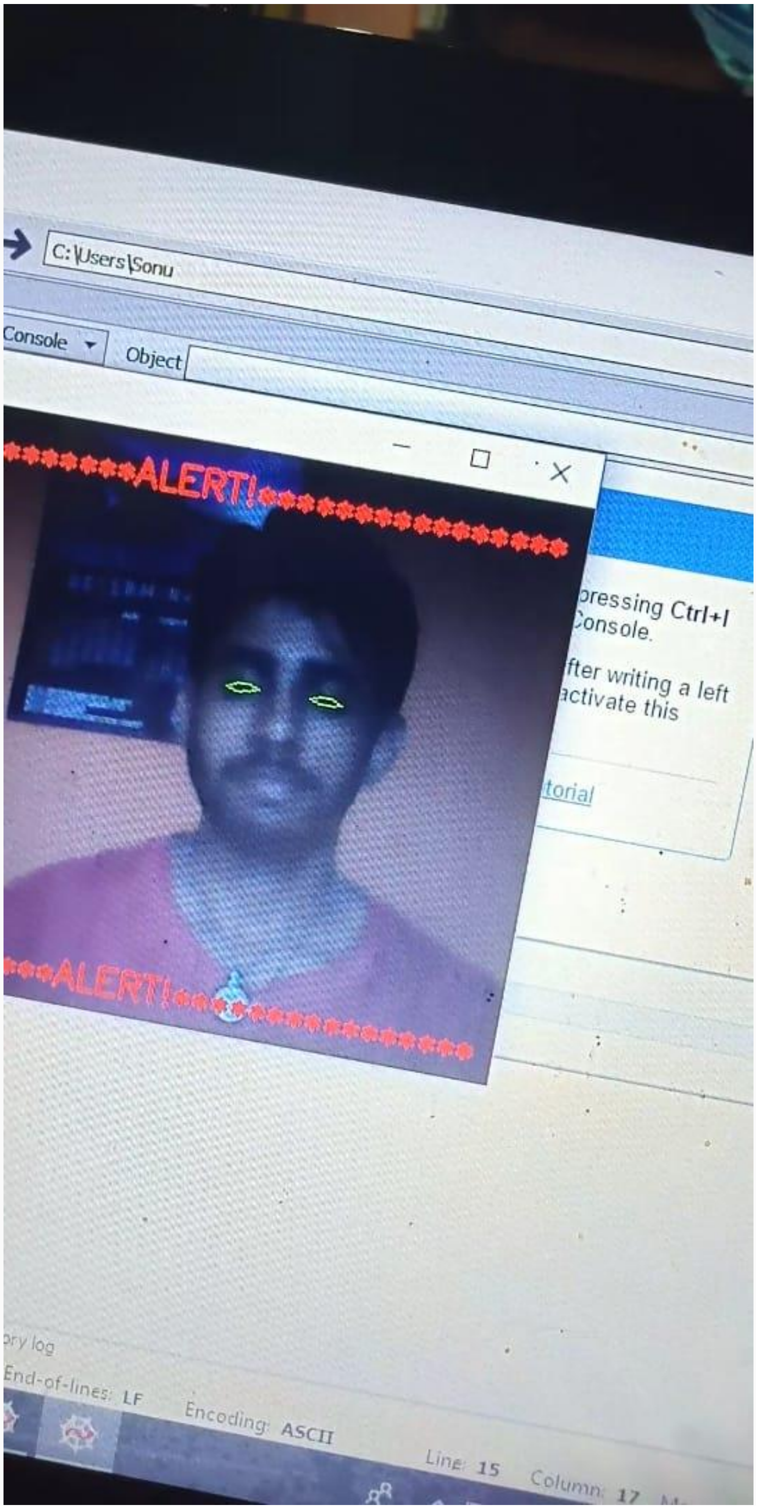
If the eye aspect ratio falls below a certain threshold and then rises above the threshold, then we'll register a "blink" — the thresh is this threshold value. We default it to a value of 0.25 as this is what has worked best for my applications, but it may need to tune it for other applications.



```
thresh = 0.25
frame_check = 20
detect = dlib.get_frontal_face_detector()
predict = dlib.shape_predictor("D:\datasets\computer vision\Drowsiness_Detection-master\shape_predictor_68_face_landmarks.dat")
```

We then have an important constant, frame_check— this value is set to 20 to indicate that 20 successive frames with an eye aspect ratio less than frame_check must happen in order for a blink to be registered.

Now that our imports and constants have been taken care of, we can initialize dlib’s face detector and facial landmark detector. The dlib library uses a pre-trained face detector. The model can work with Raspberry Pi, USB, and web cam or even with real time video processing.



Conclusion

Driver sleepiness is an important contributor to injury-involved four-wheel motor vehicle crashes in India, highlighting the need for evidence-based strategies to address this poorly characterized risk factor for vehicle crashes in less resourced settings. This model can also integrate with Alcohol detection system to make it all in one system for Accident avoiding scenarios in India. The further studies include “Object Detection” using tensorflow object detection API.