
	<b>Pimpri Chinchwad Education Trust's Pimpri Chinchwad College of Engineering</b>	
<b>Project Report</b>		

**Department - First Year  
Semester II**

**Academic Year 2019-20**

**Date-24 May 2020**

**Project Guide:** Mrs.Meera Thorat

**Submitted By:** Shivam Tikhe, Om Sarulkar, Mitali Gadiya, Shivam Dhavale

**Project Statement:** Driver's Drowsiness Detection

**Prerequisite Study:**

- Familiar with Python Programming Language
- Strong knowledge about opencv and computer vision
- Basic Knowledge to use any python interpreter

**Introduction:**

In today's world there is a tremendous increase in the number of accidents on roads. Most of the accidents occur due to carelessness or due to drowsiness of the driver. So it is necessary to develop a drowsiness detector. So we can achieve it using the computer's vision. Using this drowsiness detector we can further work on how to take drivers out of drowsiness.

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database,

remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc.

It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interface is being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers.

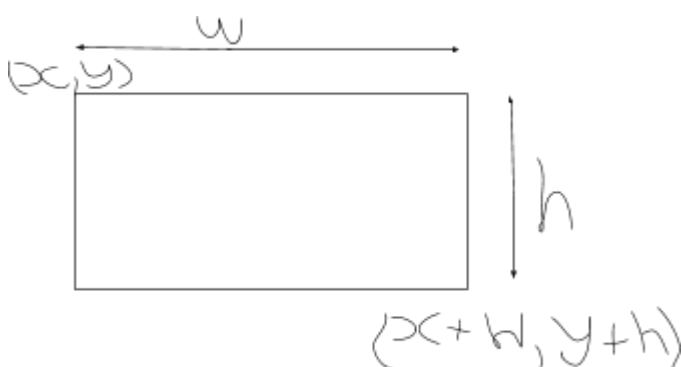
### Implementation:

We gained the knowledge of the opencv module of computer vision, also the numpy module. Picked out the best functions for the detector.

First of all we import the opencv module and numpy module. After that we used `cv2.VideoCapture` function to stream the video. The opencv has a haar cascade file named `haarcascade_frontalface_alt.xml` using this file we can detect the face, `haarcascade_eye.xml` we can detect the eyes of the grayscale frame. We have to read the frames from the video streaming using `read()` function. Now we have converted our RGB color frame to grayscale for the detection of face and the eyes using function `cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)`. Using `face.detectMultiScale()` function we get the coordinates of face and eyes use them separately for eyes and face. To stream the video over screen use function `imshow()`. To make the rectangular box over the face and eyes we have used `cv2.rectangle()` function. For the continuous flow of frames we have used the `waitkey()` function.

We have extracted the coordinates of face and calculate the area of face also we have extracted the coordinates of eyes and calculate the area. If the area of face minus eyes is equal to the area of face the eyes are closed and when they are not equal eyes are closed. Using this area details we had made the threshold value of drowsiness and if it goes above the drowsiness is detected.

### Area calculation part



Area of Face and Eyes

Left top coordinates:  $(x, y)$

Right down coordinates:  $(x+w, y+h)$

**Formula =  $w \times h$**

Code:

[https://github.com/shivamtikhe/PBL-project/blob/master/drowsiness\\_detection.py](https://github.com/shivamtikhe/PBL-project/blob/master/drowsiness_detection.py)

Screenshots:



### Conclusion:

Driver's drowsiness detection system is built to identify a driver's state while driving to reduce accidents caused due to drowsiness.

### Future Scope:

Can be used in wakeup systems of drivers in automobiles.

### References:

- Opencv Documentation
- Youtube videos
- Numpy documentations

### List of Material Submitted:

- Python Code of detection
- Video of detection
- Screenshots of detection

### Students info:

- Shivam Tikhe | FEM1352 | [shivamtikhe9@gmail.com](mailto:shivamtikhe9@gmail.com)
- Om Sarulkar | FEM1345 | [osarulkar@gmail.com](mailto:osarulkar@gmail.com)
- Mitali Gadiya | FEM1315 | [mitaligadiya07@gmail.com](mailto:mitaligadiya07@gmail.com)
- Shivam Dhavale | FEM1312 | [shivamdhavale07@gmail.com](mailto:shivamdhavale07@gmail.com)