***Methods and Techniques***

The basic idea of this project is to use 2 libraries that help us with face detection.

* OpenCV:- which uses Conventional Neural Network (CNN).
* Dlib:- which is used for real world machine learning.

What is CNN? A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm that can take an input picture, assign significance (learnable weights and biases) to different aspects or objects in the image, and distinguish between them.

We use this methodology to perform face and eye detection. With the help of this technology, we detect the main aspects of our face: The eye for checking the blink rate and for the yawn rate we use another method to see how often we yawn while driving. For both these aspects we have set a certain threshold level that has been set manually by us.

For Blink\_Rate, if we close our eyes the score increases by 1 and if we open our eyes the algorithm deducts the score by 1. So, if the score exceeds 10, that means the driver is sleeping and a warning will be sent along with an alarm that is played with the help of the PyGame library using the mixer.init() function. For this to work we first have to feed an input video frame into the system which is done using the cv2.VideoCapture(0), this function captures each frame of the video one by one. To make the frame more suitable for detection and analysis, we have to first pre-process the frame. From the frame we then get the right eye and left eye individually, on which we further perform few reconfigurations that make it suitable for quicker and simpler detection. After this we see whether the eye is open or closed, and we check this fact using an if condition. We mainly use the cv2 module, basically the OpenCV package, to perform all configurations. In the end we keep track of the Blink Rate to check whether it crosses the threshold or not.

For Yawn Rate, similar to Blink\_Rate, we have a Yawn count that will keep track of how much a person yawns while driving. For this we have to first detect certain features of the face, like its mouth where we have to find the location of the upper lip, the lower lip, the distance between both, etc. The shape predictor file (FP = ‘…\\Face\_Predict.dat’) is used to extract information from an image, such as the corners of eyes, regions surrounding the nose, and so on. The frame is transformed to a matrix, and the MarkFacem() method marks the characteristics that subsequent algorithms utilize to extract other attributes such as upper lips, lower lip, and so on. But now, one of the other main task is to determine what makes for a “yawn”. If the distance between the lower lip and upper lip is greater than 35, then we classify it as a yawn. When our Rate of Yawn or RoY becomes 10, then it displays a message saying, “Take a Break!” for 6 seconds. So, our threshold value here is 10.