Driver Drowsiness Detection and Driver Safety

- ## Objective

  On highways, many accidents occur due to the vehicle driver losing focus on driving because he is drowsy or distracted. We have written a code in Python which will enable us to detect whether the driver is drowsy or distracted, which is likely to be the cause of such accidents.

- ## Detecting facial features using facial landmarks

  We take frames from the video as input and pass each of them through a while loop. The frame rate of the camera is 30 fps.

  The 68 facial landmarks from Python's dlib library helps us detect the face and identify different facial features for each frame.

  We use the facial landmarks corresponding to the eyes of the person and calculate the aspect ratio of both eyes and find the average of it.

  We also calculate the aspect ratio of the mouth.

- ## Drowsiness Detection through eye aspect ratio

  If the person is feeling sleepy, he/she will close his/her eyes for some time, which is likely to make them lose focus and cause an accident. We need to ring an alarm to bring back the focus of the person.
  So, for that, we will check the aspect ratio of the eyes. If the average aspect ratio falls below a certain threshold for a certain amount of time (here, the time is calculated by counting the number of frames, i.e., 30 frames are received per second. The sleep_score variable increases by 1

for every frame having the eye aspect ratio below the threshold), then the alarm goes off and an alert message is shown on the screen.

- **Yawning**

  If a person is yawning frequently, it is a sign that he/she is feeling sleepy. To detect drowsiness through yawning, we use the same approach as we used for eyes, that is calculating the aspect ratio of the mouth.

  So, if the aspect ratio of the mouth remains more than a certain threshold for a certain number of frames (out of a total number of frames), then the alarm will go off and an alert message will be shown on the screen.

- **Head pose detection**

  If the driver is not looking on the road and is looking sideways, then there is a high chance of accident, because of him/her not focusing on the road.

  To overcome this and ensure that the driver regains his focus within a short amount of time, we use head pose detection.

  By analyzing the location of the facial landmark numbered 34, which is the landmark of the nose tip, relative to its initial position, we can accurately predict if the driver is facing sideways or not.

If he is looking sideways for a certain amount of time (i.e, for a certain number of frames), then the alarm will go off and an alert message will be displayed on the screen.

- **Mobile phone detection**

  Many times, using a mobile phone while driving makes the driver lose focus on the road, and it is likely to cause accidents.

  To prevent this, we used a state-of-the-art CNN model called YOLO for detecting and predicting names of various objects.

  It is pre-trained on a COCO (Common Objects in Context) dataset which includes a list of 80 common objects, ranging from knives to people to teddy bears.

  We use this model to detect mobile phones in the image (frame). If the mobile phone is detected for a certain time, then the alarm goes off and an alert message is shown on the screen.

All these features use a separate and independent variable which increases by 1 for every frame which satisfies their respective conditions, and are known as score variables. When any of these variables cross their respective thresholds, the alarm goes off.

And when their respective condition is not satisfied by the frame, they decrease by 1 for each frame, and the alarm keeps going on until all the variables are on the safe side of their respective thresholds.

Sometimes, it may happen that some of the score variables have gone so much beyond their thresholds that they do not immediately return to the safe

zone. This can be annoying to the driver if the alarm keeps going on even when he/she is no longer violating any of the conditions.

So, to overcome this, we added a new feature, such that if the driver presses a certain key, then all of the variable values are instantly initialized to 0. This will stop the annoyance suffered by the driver.

- **Detecting facial features in low light conditions**

  The code is unable to detect facial features when there is a low light condition. Given that most of the drivers feel drowsy in night time, when there are low light conditions, it becomes extremely important to make the code detect the facial features even in the low light conditions.

  To solve this problem, we have used a state-of-the-art CNN model, named Zero-DCE (i.e., Zero-Reference Deep Curve Estimation), which is used for low light enhancement.

  It is a pre-trained model, trained on a huge number of trial images. Its work is to illuminate the low light image given to it as an input, and so as to increase its visibility.

  The incoming video is divided into frames, and each frame is passed into the model, and the output frame, which is enhanced, is then given as the input to the original code. After that, the code is able to detect the facial features because of the illuminated image and detect the above mentioned features as well.