Drowsiness detection of a driver to prevent road accidents

***Abstract*—Proper attention of driver is must for safe driving and accident prevention but feeling sleepy while driving is the major reason for road accidents. The project will alert the driver as soon as he starts yawning or closes his eyes for more than a certain threshold value of time, so that he can focus on driving and will thus help greatly in preventing road accidents.**

I. Introduction

The problem trying to solve

- With this Python project, we have developed a drowsiness detecting system. A countless number of people drive on the highway day and night. Taxi drivers,bus drivers, truck drivers, and people traveling long-distance suffer from lack of sleep and overload , due to which they can pose dangers to other riders on the road.
- The majority of road accidents happen due to the drowsiness of the driver. So, to prevent these accidents, we have built a smart system which will detect whenever the driver will feel drowsy or sleepy based on monitoring the driver's eyes and mouth and will alert him to prevent mishap.

It is important and challenging problem because :

- This problem has recently grabbed the attention of many researchers around the world, leading to innumerable projects in this field. These projects aim at warning the driver when he feels drowsy to prevent mishappenings on the road. The alert is sounded as an alarm to grab the driver's attention so that he remains alert while on the road.
- The Problem is important and attention need to be paid because its estimated over 45 % - 50 % of the total road accidents occurs due to drowsiness and fatigue of driver. These accidents are most of the times are extremely dangerous and life threatening.
- In USA around 1.2 lac accidents took place due to fatigue and driver's drowsiness. So its important to have a solution to this problem.

Existing solutions are:

- There exits solution to this problem, having a solution similar to this project, but they have a downpoint.
- Most Of the current models use Har Cascade Classifier. Use of Har Cascade classifier increases the computation time and is way less accurate than HOG classifier used in this project.
- If driver's face orientation is improper Har Cascade may give less accurate results. Since Har features are to be determined manually there is always a limit on things it can detect.

Core idea of our project is

- Detecting the driver's face and then using its facial landmarks of eyes and mouth.
- Properly detecting whether driver is feeling drowsy or not.
- Generation of alert when he yawns or feel drowsy after a threshold value.

Improvement over existing models:

face detector we use is made using the classic Histogram of Oriented Gradients (HOG) feature combined with a linear classifier, an image pyramid, and a sliding window detection scheme.The pose estimator was created by using dlib's implementation of the paper: One Millisecond Face Alignment with an Ensemble of Regression Trees by VahidKazemi and Josephine Sullivan, CVPR 2014Most Of the current models use Har Cascade Classifier whereas our model uses HOG classifier Dlib is ahead of Haar cascade classifier over implementation, speed, and accuracy. There are several benefits of using the HOG classifier. First, the training is done using a sliding sub-window on the image, sono sub sampling and parameter manipulation is required like it is in the Haar classifier. This makes Dlib's HOG + SVMface detection easier to use and faster to train. HOG has higher accuracy for face detection than Haar cascade classifier.

II. Literature Survey

References we used for our project:

For facial landmark detection we have taken idea and learnt basics of facial recognition from Liu, J., Y. Wang and W. Han. 2019. Face recognition and tracking based on video image. Electronic Technology and Software Engineering

For learning more on dlib we referred to One Millisecond Face Alignment with an Ensemble of Regression Trees paper by Kazemi and Sullivan (2014).

To learn more on existing models we referred to Rainer Lienhart and Jochen Maydt. An extended set of haar-like

features for rapid object detection. In Image Processing. 2002. Proceedings. 2002 International Conference on, volume 1, pages I–900. IEEE, 2002
above mentioned paper uses har cascade classifier which is slower to dlib HOG based classifier

III. Problem Statement

"Drowsiness detection of a driver to prevent road accidents"

A. Objectives

- Get the live stream from a camera
- Analyse each frame from the live video
- Locate all the faces present in the image frame
- For each face
  1) Detection of facial landmarks and calculation of eye and mouth aspect ratio
  2) Detection of yawn and drowsiness based on calculated ratio
  3) Raising alarm when yawn or drowsiness is detected

B. Solved using:

- Live video stream provided using imutils and VideoStream package
- Dlib to detect the faces in the image
- Use 68x facial landmark detection model (dlib) for facial land mark detection, it uses Histogram of Oriented Gradients (HOG) feature combined with a linear classifier.
- face_utils package to convert the facial landmarks into a numpy array for easier use
- cv2 to grayscale the images and to also draw contours around eyes and mouth in live video
- Sounding the alarm on a new thread using mpg123 audio player

IV. Methodology

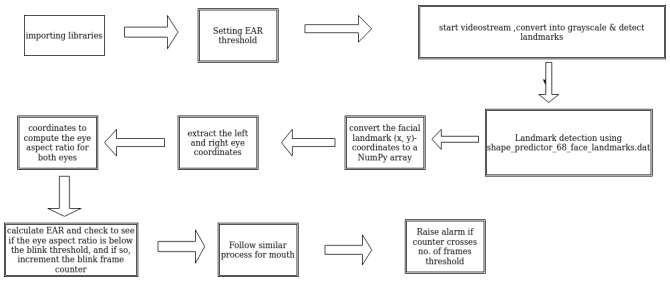

Fig. 1. Flowchart

The camera is placed in front of the driver so that the face of the driver lies in the range of the camera.The videostream is then started. The system then detects all the faces in the frame with the help of get_frontal_face_detector functon dlib library of python. The face is then grayscaled and the eye and mouth landmarks are then traced out using the shape_predictor_68_face_lanmarks.dat which is a freely available .dat file available for developer's help at the dlib's website. Most Of the current models uses Har Cascade Classifier where as this dlib's model uses HOG classifier.HOG works with something called a block which is similar to a sliding window. A block is considered as a pixel grid in which gradients are constituted from the magnitude and direction of change in the intensities of the pixel within the block.

Detecting a face from an image using HOG method requires

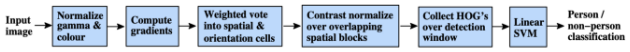

Fig. 2. Working of HOG

the following steps to be followed :-

- Calculate the Gradient Images
- Compute the HOG
- Block normalization
- Feeding the vector into SVM

This results in high accuracy and less time for detection of facial landmarks.

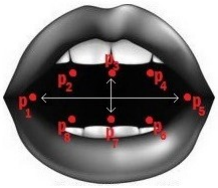

Fig. 3. Co-ordinates considered for calculating Mouth Aspect Ratio(MAR)

The facial landmarks are then converted into a numpy array using face_utils package and passed to the EAR and MAR functions where the respective EAR(Eye Aspect Ratio) and MAR(Mouth Aspect Ratio) values are computed. The Ear is calculated by taking the average of left EAR and right EAR.These values are then compared with the threshold values.If the value falls below the blink threshold or yawn threshold,the respective frame counter is incremented. If any of the above frame counter exceeds its normalised threshold (values from research paper) an alarm is sounded along with the display of a warning message on the screen.

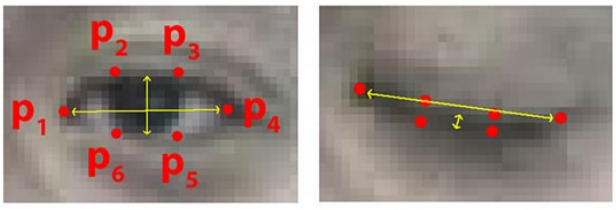

Fig. 4. Co-ordinates considered for calculating EYE Aspect Ratio(EAR)

We have stored driver's name along with count he/she feels drowsy and yawn. We have removed all the duplicates (like one driver may take multiple rides) , so we will replace his yawn and drowsiness count by average of data and other rows of his name are deleted(groupby function is used for

removal of duplicates). We then plot the graph of scores of all drivers currently present in our records. Best driver is picked on the basis of least score calculated by (1.5*average_yawn_count+3.5*average_drowsiness_count).His/Her name along with the score is displayed.

V. Results and Analysis

Fig. 5. Output in all possible scenarios

The output snapshots provided above depicts how the application detects the driver's drowsiness and yawning accurately and sounds the alarm bell to alert the driver to prevent any mishap. The data about the drowsiness and yawning are also updated in the record to provide proper feedback and rating.

A text file is auto-generated/updated file by the python code to provide a detailed analysis for that particular driver. The name of the file is the same as the driver's name. It offers excellent assistance to cab company managers to get an overview of their driver's alertness.

To get the risk score of all the drivers, the ratinganalysis python file is executed. It updates the record file with every driver's entry by taking the average of all the entries present in the file with that driver's name, displays the best driver's name, and shows a graphical analysis of all the drivers' performance.

	Name	Average Drowsiness Count	Average Yawn Count
2	Yash	0	1
3	Rishit	1	1
4	sam	1	2
5	sue	2	3
6	sam	31	4
7	sue	2	5
8	amar	3	1
9	rahul	21	3
10	amar	7	4
11	rahil	6	1
12	sahil	0	23
13	rahul	1	4
14	sam	1	12
15	rahil	6	2

Fig. 6. Record file before running ratinganalysis code(Consists of the entries generated by main code)

	Name	Average Drowsiness Count	Average Yawn Count
0	Rishit	1	1.0
1	Yash	0	1.0
2	amar	5	2.5
3	rahil	6	1.5
4	rahul	11	3.5
5	sahil	0	23.0
6	sam	11	6.0
7	sue	2	4.0

Fig. 7. Record file after running ratinganalysis code(Average of the entries for each driver)

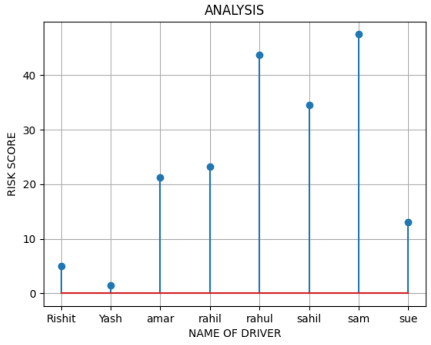

Fig. 8. Graphical analysis

In a nutshell, the project accurately detects the driver's alertness based on his eyes and mouth. It appropriately alerts the driver if he feels sleepy and will undoubtedly prove effective and efficient in most scenarios. It also provides the best possible feedback mechanism for the driver's self-improvement, for passenger's trust, and for the driver's employer to be sure how skilled and devoted the driver is.
(We have tested this project extensively. It always provides the correct and desired results in proper lighting. Still, it does not

prove much efficient in low-light scenarios as detecting the eyes and mouth is quite tricky in insufficient lighting).

VI. Conclusion

Our project aims at preventing road accidents by alarming the driver when he/she feels drowsy. The program first detects the face of the driver, which has been implemented by using the dlib module of python. The eye and the mouth landmarks are then detected from the face of the driver.

The eye and mouth coordinates are then converted into a NumPy array, with the help of which the Eye aspect ratio and Mouth aspect ratio are calculated. These values are then compiled and compared with a threshold value. On crossing this threshold, an alarm is raised, and a message is displayed on the system.

The respective driver's details are then stored and compiled in a database in the form of the count of yawns and sleepy tendencies (count of eye closures). These drivers are then rated according to the results obtained, which can be used in the future to help the passenger know the driver's experience of driving for a better and safe ride.

Acknowledgment

We would like to express our gratitude to Mr.Dinesh Naik for his assistance in making our project successful, along with the IT department, NITK, for providing us the opportunity to execute this project.

References

[1] https://docs.opencv.org/master/da/df6/tutorial_py_table_of_contents_setup.html

[2] http://www.willberger.org/cascade-haar-explained/

[3] Liu, J., Y. Wang and W. Han. 2019. Face recognition and tracking based on video image. Electronic Technology and Software Engineering

[4] Rainer Lienhart and Jochen Maydt. An extended set of haar-like features for rapid object detection. In Image Processing. 2002. Proceedings. 2002 International Conference on, volume 1, pages I–900. IEEE, 2002