# Real-Time Driver Drowsiness Detection System Using Eye Aspect Ratio and Eye Closure Ratio

*Sukrit Mehta, Sharad Dadhich, Sahil Gumber, Arpita Jadhav Bhatt*

*Jaypee Institute Of Information Technology, A10, Block A, Industrial Area, Sector 62, Noida*

ARTICLE INFO

*Article history:*
Received 15 January 19
Received in revised form 23 January 19
Accepted 24 February 19

Keywords:
Mobile Computing
Drowsiness detection
Facial Landmark Detection
EAR (Eye Aspect Ratio)
ECR (Eye Closure Ratio)
Fatigue Detection
Non-Intrusive Methods
Driver monitoring system

ABSTRACT

Every year many people lose their lives due to fatal road accidents around the world and drowsy driving is one of the primary causes of road accidents and death. Fatigue and micro sleep at the driving controls are often the root cause of serious accidents. However, initial signs of fatigue can be detected before a critical situation arises and therefore, detection of driver's fatigue and its indication is ongoing research topic. Most of the traditional methods to detect drowsiness are based on behavioural aspects while some are intrusive and may distract drivers, while some require expensive sensors. Therefore, in this paper, a light-weight, real time driver's drowsiness detection system is developed and implemented on Android application. The system records the videos and detects driver's face in every frame by employing image processing techniques. The system is capable of detecting facial landmarks, computes Eye Aspect Ratio (EAR) and Eye Closure Ratio (ECR) to detect driver's drowsiness based on adaptive thresholding. Machine learning algorithms have been employed to test the efficacy of the proposed approach. Empirical results demonstrate that the proposed model is able to achieve accuracy of 84% using random forest classifier.

Peer review under responsibility of International Conference on Sustainable Computing in Science, Technology and Management.

## 1. Introduction

One of the major causes behind the casualties of people in road accidents is driver's drowsiness. After continuous driving for long time, drivers easily get tired resulting into driver fatigue and drowsiness. Research studies have stated that majority of accidents occur due to driver fatigue. Different countries have different statistics for accidents that occurred due to driver fatigue. Developing technology for detecting driver fatigue to reduce accident is the main challenge. According to the report by "Ministry of Road Transport & Highways" there were 4,552 accidents reported every year in India, that took lives of thousands of people because of sleepy drivers(Road Accidents in India 2016). For instance, many vehicles are driven mostly at night such as loaded trucks. The drivers of such vehicles who drive for such continuous long period become more susceptible to these kinds of situations. Detecting drowsiness of drivers is still an ongoing research in order to reduce the number of such miss-happenings and accidents. Typical methods used to identify drowsy drivers are physiological based, vehicle based, and behavioural based(S. Sangle, B. Rathore, R. Rathod, A. Yadav, and A. Yadav,2018)–( A. Kumar and R. Patra,2018). Physiological methods such as heartbeat, pulse rate, and Electrocardiogram(T. Hwang, M. Kim, S. Hong, and K. S. Park,2016), (S. Junawane, S. Jagtap, P. Deshpande, and L. Soni,2017) etc. are used to detect fatigue level. Vehicle based methods include accelerator pattern, acceleration and steering movements. Behavioural methods(S. Sangle, B. Rathore, R. Rathod, A. Yadav, and A. Yadav,2018)–( A. Kumar and R. Patra,2018) include yawn, Eye Closure, Eye Blinking, etc. To encounter this worldwide problem, a solution that captures images in a succession, transmits real-time driver's data to the server, and determines drowsiness using EAR (Eye Aspect Ratio) and ECR (Eye Closure Ratio) has been proposed and implemented using Android application. The computed value via the system prompts the driver to take a break or rest for some time. The methods used are non-intrusive in nature; hence, no additional costs would be incurred during the course of the drowsiness detection method. The rest of the paper is organised as follows. In section 2, the literature review is presented. Section 3 presents the proposed approach to detect driver's drowsiness. It also details the components which are developed as a part of application to compute EAR and ECR. Section 4 describes comparison of the proposed approach with existing approach. Section 5 describes the performance evaluation with discussion of experimental results. The paper is concluded in Section 6.

## 2. Related Work

In order to detect drowsiness of drivers, numerous approaches have been proposed. This section summarizes the existing approaches to detect drowsiness. Rateb et al. (R. Jabbar, K. Al-Khalifa, M. Kharbeche, W. Alhajyaseen, M. Jafari, and S. Jiang,2018)detected real-time driver drowsiness using deep neural networks. They developed an Android application. Tereza Soukupova et al. (T. Soukupova and J. Cech,2016)used EAR(Eye Aspect Ratio) as a standard measure to compute drowsiness of a person. They also detailed the types of systems used for detecting drowsiness of driver. For example, Active Systems (considered as reliable, but use special hardware that are expensive and intrusive like infrared cameras etc.) and Passive Systems (are inexpensive and rely on Standard cameras). Shailesh et al.( S. Sangle, B. Rathore, R. Rathod, A. Yadav, and A. Yadav,2018) used a camera fixed on the dashboard to capture and send images to Raspberry Pi server installed in the vehicle, to detect faces using Harr classifier and facial points using the Dlib Library. Vibin Varghese(V. Varghese, A. Shenoy, S. Ks, and K. P. Remya,2018)detected landmarks for every frame captured to compute the EAR(between height and width of eye) using the landmark points of face. After computing the EAR; (V. Varghese, A. Shenoy, S. Ks, and K. P. Remya,2018) determined the driver as drowsy if the EAR was less than the limit for 2 or 3 seconds (because the eye blink lasts approximately 100-400ms). Ashish Kumar( A. Kumar and R. Patra,2018) used Mouth Opening Ratio as a parameter to detect yawning during drowsiness. There are several other research works that have been conducted to determine vision based drowsiness detection(I. García, S. Bronte, L. M. Bergasa, J. Almazán, and J. Yebes,2012)–( K. Srijayathi and M. Vedachary,2013), fatigue detection(A. Chellappa, M. S. Reddy, R. Ezhilarasie, S. Kanimozhi Suguna, and A. Umamakeswari,2015), eye-tracking to detect driver fatigue(2011). Thus, with reference to the literature work we have a proposed a system that detects driver's drowsiness using EAR and ECR which are detailed in the following section.

## 3. Proposed Approach to detect Driver's Drowsiness

This section details the proposed approach to detect driver's drowsiness that works on two levels. The application is installed on driver's device running Android operating system (OS). The process starts with capturing of live images from camera and is subsequently sent at local server. At the server's side, Dlib library is employed to detect facial landmarks and a threshold value is used to detect whether driver is drowsy or not(T. Soukupova and J. Cech,2016). These facial landmarks are then used to compute the EAR (Eye Aspect Ratio) and are returned back to the driver. In our context, the EAR value received at the application's end would be compared with the threshold value taken as 0.25(T. Soukupova and J. Cech,2016). If the EAR value is less than the threshold value, then this would indicate a state of fatigue. In case of Drowsiness, the driver and the passengers would be alerted by an alarm. The subsequent section details the working of each module.

### *3.1. Data Procurement*

For using the application, the driver performs a registration if using the application for the first time. After performing a sign-up, the driver adds a ride by entering the source and destination of the ride. Likewise, an interface for the passengers is also provided where the passengers can connect with the ride, added by the driver. The driver then starts the ride. The proposed application then captures the real-time images of the driver. Images are captured every time the application receives a response from the server. The process goes on until the driver stops the ride. For testing the efficiency of the proposed approach, a data set of 50 volunteers was collected. Every participant was asked to blink their eyes intermittently while looking at camera for capturing EAR values. The logs of the results that were captured by the application were collected and analysed with the help of machine learning classifiers.

### *3.2. Facial Landmark Marking*

To extract the facial landmarks of drivers, Dlib library was imported and deployed in our application(T. Soukupova and J. Cech,2016), (J. D.

Fuletra,2013). The library uses a pre-trained face detector, which is based on a modification to the histogram of oriented gradients and uses linear SVM (support vector machine) method for object detection. Actual facial landmark predictor was then initialized and facial landmarks captured by the application were used to calculate distance between points. These distances were used to compute EAR value (K. C. Patel, S. A. Khan, and V. N. Patil,2018). EAR is defined as the ratio of height and width of the eye and was computed using equation 1. The numerator denotes the height of the eye and the denominator denotes the width of the eye and the details of the all the landmarks of eye are depicted by figure 1.

$$EAR = \frac{(\ |p2 - p6| + |p3 - p5|\ )}{2*|p1 - p4|} \qquad \text{-(1)}$$

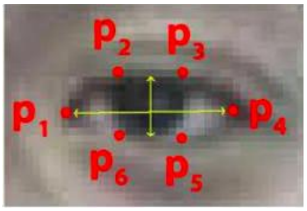


**Fig 1. Landmarks of Eye in EAR**

Referring equation 1, the numerator calculates the distance between the upper eyelid and the lower eyelid. The denominator represents the horizontal distance of the eye. When the eyes are open, the numerator value increases, thus increasing the EAR value, and when the eyes are closed the numerator value decreases, thus decreasing the EAR value. In this context, EAR values are used to detect driver's drowsiness. EAR value of left and right eyes is calculated and then average is taken. In our drowsiness detector case, the Eye Aspect Ratio(K. C. Patel, S. A. Khan, and V. N. Patil,2018) is monitored to check if the value falls below threshold value and also it does not increases again above the threshold value in the next frame. The above condition implies that the person has closed his/her eyes and is in a drowsy state. On the contrary, if the EAR value increases again, it implies that the person has just blinked the eye and there is no case of drowsiness. Figure 2 depicts the block diagram of our proposed approach to detect driver's drowsiness. Figure 3 represents a snapshot of facial landmark points using Dlib library, which are used to compute EAR. Table 1 details the facial landmark points for left and right eye which were used for computation.

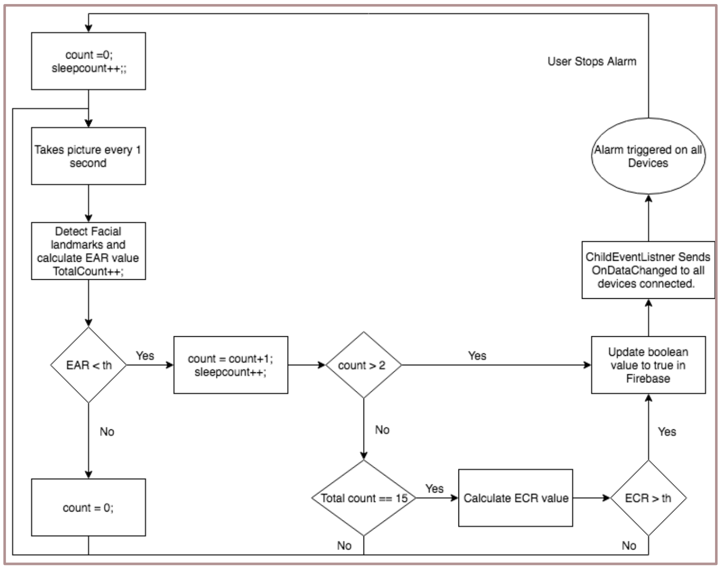

count =0;
sleepcount++;;
User Stops Alarm
Takes picture every 1 second
Alarm triggered on all Devices
Detect Facial landmarks and calculate EAR value TotalCount++;
ChildEventListner Sends OnDataChanged to all devices connected.
EAR < th
Yes
count = count+1; sleepcount++;
count > 2
Yes
Update boolean value to true in Firebase
No
No
Yes
count = 0;
Total count == 15
Yes
Calculate ECR value
ECR > th
No
No

**Fig. 2 - Block Diagram of Proposed Drowsiness Detection Algorithm**

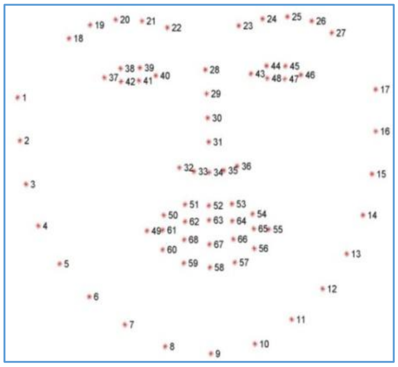


**Fig. 3 –Facial Landmark Points according to Dlib library**

**Table 1 – Facial Landmarks(A. Kumar and R. Patra,2018)**

| **Part** | **Landmark Points** |
|---|---|
| Left Eye | [37-42] |
| Right Eye | [43-48] |

### *3.3. Classification*

After capturing the facial landmark points, EAR value computed by the server is now received at the android device of the driver and compared with the threshold value which was earlier set to be 0.25(T. Soukupova and J. Cech,2016) . If the value is less than the threshold then the counter value is incremented, else the counter value is set back to zero. If the counter value reaches to three, an alarm is triggered in the android device. In addition, another variable (Sleep Counter) is maintained which counts the number of times the EAR value is less than threshold value. Variable (Total Counter) stores the total count of responses from the server side and is used to calculate the ECR (Eye Closure Ratio). It is defined as the ratio of Sleep Counter and Total Counter value and was computed using equation 2.

$$ECR = \frac{Sleep\ Counter}{Total\ Counter} \qquad \text{- (2)}$$

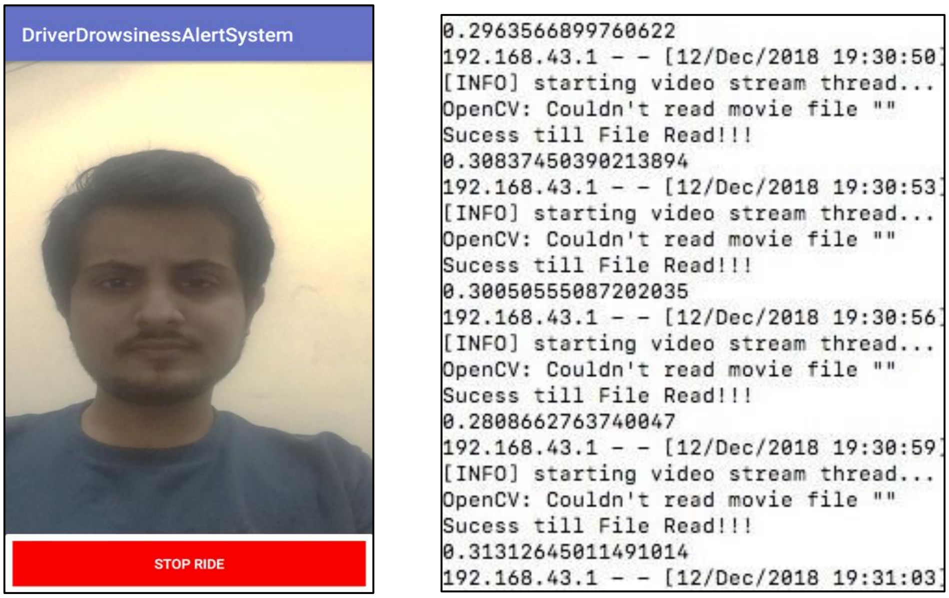
In our context, the value of ECR was calculated for every 15 consecutive frames (captured from camera). As soon as the frame number reaches to 16, the value of total counter becomes one and sleep counter becomes zero. Whenever the ECR value exceeds the threshold value which is set to 0.5, then a notification is generated in the android app to indicate drowsy state of the driver.

## 4. Comparison with state of art approach

We have used the EAR (eye aspect ratio) and proposed a way to compute ECR (Eye Closure Ratio). Compared to other methods from the literature, our proposed algorithm provides better accuracy and reduces response time of calculating the EAR at server as the server is locally setup and also the returned EAR value is locally checked in the android device of the driver thereby improving the results of alertness as soon as the driver feels drowsy. Moreover, in other intrusive methods(T. Hwang, M. Kim, S. Hong, and K. S. Park,2016), (S. Junawane, S. Jagtap, P. Deshpande, and L. Soni,2017), a number of machines and devices need to be attached to the driver's body were required, thereby making it uncomfortable for the driver to concentrate on his driving. Moreover, in previous approaches a setup needs to be performed every time, whenever the driver starts the ride. However, these intrusive methods involve a good amount of cost to measure pulse rate, heartbeats, etc. In our suggested measure, we have just used an android device and a local server to detect drowsiness that removes the factors of cost of machines and interruption in driver's concentration. In comparison to the usage of external camera in the intrusive methods, we have used the android device, which is often used by people for navigation and various purposes. The proposed algorithm has worked well in conditions having good lightning. It also works for people wearing spectacles. Following section describes the performance evaluation of the proposed approach.

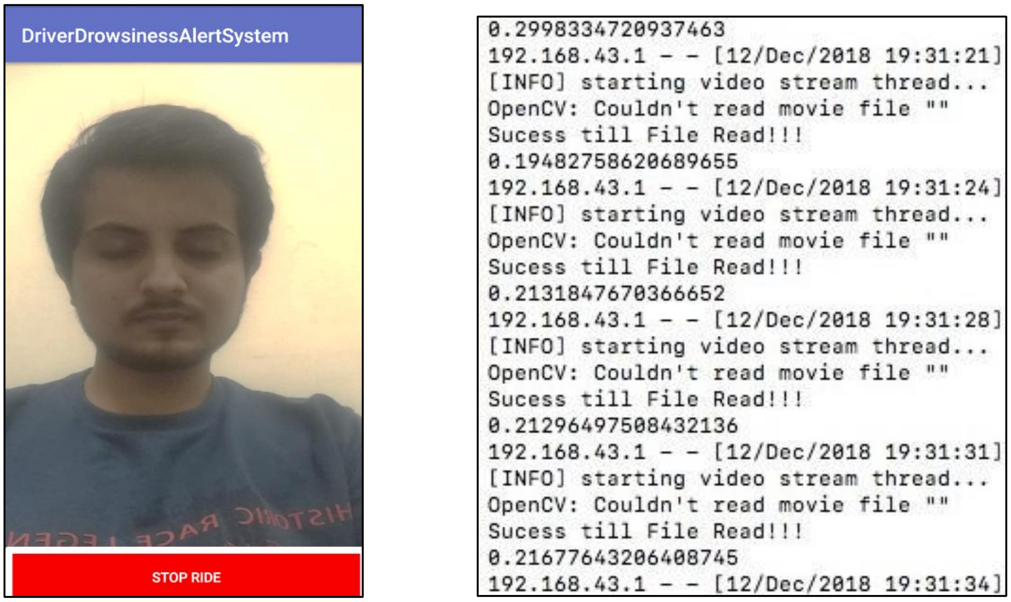
## 5. Performance evaluation with Experimental Results and Discussion

The section presents the performance evaluation of the proposed approach by performing an empirical analysis of obtained results. First, the system collects the real-time data of the drivers depicted by Figures 4-a, 4-b and 4-c. It then determines drowsiness of the drivers based on the EAR values that are computed based on the images captured of the user and its response from the server. It also detects the drowsiness using ECR values.



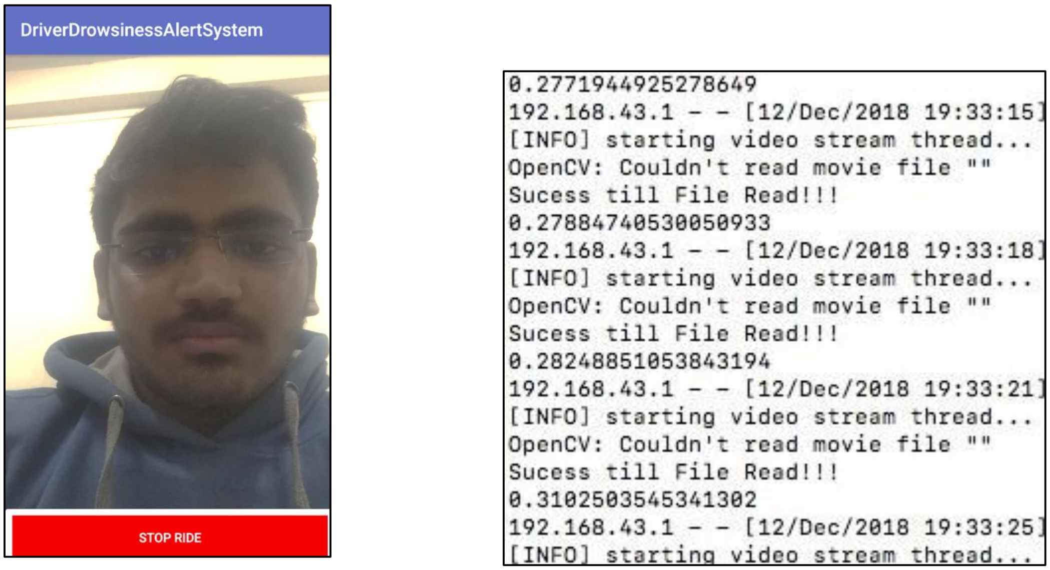
```
0.2963566899760622
192.168.43.1 - - [12/Dec/2018 19:30:50]
[INFO] starting video stream thread...
OpenCV: Couldn't read movie file ""
Sucess till File Read!!!
0.30837450390213894
192.168.43.1 - - [12/Dec/2018 19:30:53]
[INFO] starting video stream thread...
OpenCV: Couldn't read movie file ""
Sucess till File Read!!!
0.30050555087202035
192.168.43.1 - - [12/Dec/2018 19:30:56]
[INFO] starting video stream thread...
OpenCV: Couldn't read movie file ""
Sucess till File Read!!!
0.2808662763740047
192.168.43.1 - - [12/Dec/2018 19:30:59]
[INFO] starting video stream thread...
OpenCV: Couldn't read movie file ""
Sucess till File Read!!!
0.31312645011491014
192.168.43.1 - - [12/Dec/2018 19:31:03]
```

**Fig. 4 (a) - Results when eyes are open (Without Spectacles)**



```
0.2998334720937463
192.168.43.1 - - [12/Dec/2018 19:31:21]
[INFO] starting video stream thread...
OpenCV: Couldn't read movie file ""
Sucess till File Read!!!
0.19482758620689655
192.168.43.1 - - [12/Dec/2018 19:31:24]
[INFO] starting video stream thread...
OpenCV: Couldn't read movie file ""
Sucess till File Read!!!
0.2131847670366652
192.168.43.1 - - [12/Dec/2018 19:31:28]
[INFO] starting video stream thread...
OpenCV: Couldn't read movie file ""
Sucess till File Read!!!
0.21296497508432136
192.168.43.1 - - [12/Dec/2018 19:31:31]
[INFO] starting video stream thread...
OpenCV: Couldn't read movie file ""
Sucess till File Read!!!
0.21677643206408745
192.168.43.1 - - [12/Dec/2018 19:31:34]
```

**Fig. 4 (b) - Results when eyes are closed (Without Spectacles).**



```
0.2771944925278649
192.168.43.1 - - [12/Dec/2018 19:33:15]
[INFO] starting video stream thread...
OpenCV: Couldn't read movie file ""
Sucess till File Read!!!
0.27884740530050933
192.168.43.1 - - [12/Dec/2018 19:33:18]
[INFO] starting video stream thread...
OpenCV: Couldn't read movie file ""
Sucess till File Read!!!
0.28248851053843194
192.168.43.1 - - [12/Dec/2018 19:33:21]
[INFO] starting video stream thread...
OpenCV: Couldn't read movie file ""
Sucess till File Read!!!
0.3102503545341302
192.168.43.1 - - [12/Dec/2018 19:33:25]
[INFO] starting video stream thread...
```

**Fig. 4 (c) - Results when eyes are open (With Spectacles).**

Two-way analysis has been performed in our work. Our first phase includes the results obtained by the android application when the driver faces the camera. Data is collected from this phase, is further used in the second phase where detailed analysis of the results has been performed using machine learning classifiers to test the effectiveness of the proposed approach. Classifiers that were employed for empirical analysis were Naive Bayes, Support Vector Machine and Random Forest(K. Das and R. N. Behera,2017). To evaluate the performance of the classifiers, we compared the results obtained based on standard performance metrics. Naive Bayes Classifier is used to identify objects by applying Bayes Algorithm. Random Forest Classifier is an ensemble algorithm which generates a set of uncorrelated decision trees by randomly selecting the subset of training set and then aggregates them to arrive at a conclusion. SVM (Support Vector Machine) is a discriminative classifier that finds out a line that demarcates the classes. Table 2 enumerates the results obtained by employing different classifiers.

**Table-2: Results (in percentage) obtained after applying different classifiers**

**TPR: True Positive Rate, FPR: False Positive Rate, SVM: Support Vector Machine**

| S.No. | Classifier | TPR | FPR | Accuracy | Precision | Recall | F-Measure |
|---|---|---|---|---|---|---|---|
| 1. | Naive Bayes | 80 | 20.7 | 80 | 80.7 | 80 | 79.8 |
| 2. | SVM | 80 | 20.4 | 80 | 80.1 | 80 | 79.9 |
| 3. | Random Forest | 84 | 16.1 | 84 | 84 | 84 | 84 |

From table 2 we enumerate that Random Forest classifier give the best classification results with the accuracy of 84%.

## 6. Conclusion and Future Work

In this work, a real time system that monitors and detects the loss of attention of drivers of vehicles is proposed. The face of the driver has been detected by capturing facial landmarks and warning is given to the driver to avoid real time crashes. Non-intrusive methods have been preferred over intrusive methods to prevent the driver from being distracted due to the sensors attached on his body. The proposed approach uses Eye Aspect Ratio and Eye Closure Ratio with adaptive thresholding to detect driver’s drowsiness in real-time. This is useful in situations when the drivers are used to strenuous workload and drive continuously for long distances. The proposed system works with the collected data sets under different conditions. The facial landmarks captured by the system are stored and machine learning algorithms have been employed for classification. The system gives best case accuracy of 84% for random forest classifier.

The future work can include integration of the proposed system with globally used applications like Uber and Ola. The system, if integrated, can reduce the number of casualties and injuries that happen regularly due to these drowsy states of the drivers. This experiment can run as a part of pilot plan i.e. for a few days/months in different regions of the world where such incidents occur regularly. Thus, our proposed approach also gives the same accuracy for the people wearing spectacles. Accuracy of our proposed system improves with the increase in brightness of the surrounding environment. The work can be extended for different types users such as bike riders or in different domains like railways, airlines etc.

## References

"Road Accidents in India 2016," 2016.

S. Sangle, B. Rathore, R. Rathod, A. Yadav, and A. Yadav, "Real Time Drowsiness Detection System," pp. 87–92, 2018.

V. Varghese, A. Shenoy, S. Ks, and K. P. Remya, "Ear Based Driver Drowsiness Detection System," vol. 2018, pp. 93–96, 2018.

A. Kumar and R. Patra, "Driver drowsiness monitoring system using visual behaviour and machine learning," *ISCAIE 2018 - 2018 IEEE Symposium on Computer Applications and Industrial Electronics*, pp. 339–344, 2018.

*T. Hwang, M. Kim, S. Hong, and K. S. Park, "Driver drowsiness detection using the in-ear EEG," Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS, vol. 2016–Octob, pp. 4646–4649, 2016.*

S. Junawane, S. Jagtap, P. Deshpande, and L. Soni, "Driver Drowsiness Detection Techniques : A Survey," vol. 6, no. 11, pp. 2015–2017, 2017.

R. Jabbar, K. Al-Khalifa, M. Kharbeche, W. Alhajyaseen, M. Jafari, and S. Jiang, "Real-time Driver Drowsiness Detection for Android Application Using Deep Neural Networks Techniques," *Procedia Computer Science*, vol. 130, pp. 400–407, 2018.

T. Soukupova and J. Cech, "Real-time eye blink detection using facial landmarks," *Computer Vision Winter Workshop (CVWW)*, 2016.

I. García, S. Bronte, L. M. Bergasa, J. Almazán, and J. Yebes, "Vision-based drowsiness detector for real driving conditions," *IEEE Intelligent Vehicles Symposium, Proceedings*, pp. 618–623, 2012.

S. S. Nagargoje and D. S. Shilvant, "Drowsiness Detection System for Car Assisted Driver Using Image Processing," *International Journal of Electrical and Electronics Research ISSN*, vol. 3, no. 4, pp. 175–179, 2015.

S. Podder and S. Roy, "Driver's drowsiness detection using eye status to improve the road safety," *International Journal of Innovation Research in Computer and Communication Engineering*, vol. 1, no. 7, 2013.

F. Omidi and G. Nasleseraji, "Non-intrusive Methods used to Determine the Driver Drowsiness: Narrative Review Articles," *International Journal of Occupational Hygiene*, vol. 8, no. 3, pp. 186–191, 2016.

A. Gopal and V. . Vineeth, "Driver Drowsiness Detection System," *International Journal of Advances in Electronics and Computer Science*, vol. 4, no. 3, pp. 101–104, 2017.

K. Srijayathi and M. Vedachary, "Implementation of the Driver Drowsiness Detection System," *International Journal of Science, Engineering and Technology Research (IJSETR)*, vol. 2, no. 9, pp. 1751–1754, 2013.

A. Chellappa, M. S. Reddy, R. Ezhilarasie, S. Kanimozhi Suguna, and A. Umamakeswari, "Fatigue detection using Raspberry Pi 3," *International Journal of Engineering and Technology(UAE)*, vol. 7, no. 2, pp. 29–32, 2018.

H. Singh, J. S. Bhatia, and J. Kaur, "Eye tracking based driver fatigue monitoring and warning system," *India International Conference on Power Electronics, IICPE 2010*, 2011.

J. D. Fuletra, "A Survey on Driver's Drowsiness Detection Techniques," *International Journal on Recent and Innovation Trends in Computing and Communication*, no. November, pp. 816–819, 2013.

K. C. Patel, S. A. Khan, and V. N. Patil, "Real-Time Driver Drowsiness Detection System Based on Visual Information," vol. 8, no. 3, pp. 16200–16203, 2018.

K. Das and R. N. Behera, "A Survey on Machine Learning: Concept, Algorithms and Applications," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 5, no. 2, pp. 1301–1309, 2017.