

# Drowsiness Detection System with Speed Limit Recommendation using Sentiment Analysis

Kavyashree Subramonian, G. Sumathi

**Abstract:** Driving while drowsy is a ubiquitous and extremely grave public health hazard that requires immediate consideration. Through studies in recent years, it has been proved that about 20 percent of all car accidents have occurred as a result of dozy driving. The main objective of new drowsiness detection systems is accurate doziness recognition. In this regard, the face is the most important part of the body as it sends a lot of essential information. The facial expressions of a drowsy driver include frequency of blinking and yawning. This paper proposes a model which detects the drivers' awareness using video stills of the driver's face and improves the tracking accuracy. Further, we introduce the auxiliary functionality of speed limit recommendations based on the driver's present state of mind. The various facial features are evaluated to determine the drivers' current state. By combining the features of the eyes and mouth, the driver is alerted with a fatigue warning and also suggested a safe speed limit. This system is very essential so as to prevent and hence reduce the number of fatal accidents that occur as a result of dozy driving saving a lot of lives and damage to property.

**Keywords:** Drowsiness Detection, Speed Limit Recommendation, Sentiment Analysis

## I. INTRODUCTION

Drowsiness can be described as feeling unusually sleepy during the day. A person who is drowsy may tend to fall asleep at inappropriate times leading to embarrassing or unpleasant situations. Sleep deprivation, driving for long periods of time, and when circadian rhythms are low (early morning hours or mid-afternoon) are some of the factors that may cause a driver to be dozy [1]. Various aspects of the functioning of the human body that are hypercritical to driving safely such as reflexes, attentiveness and the processing of information are severely affected by drowsiness. In recent years, drowsiness and fatigue have become the main reasons for the increasing number of road accidents in India and the rest of the world [2]. This is backed by the fact that drivers' performance deteriorates with escalated drowsiness.

The National Highways Authority of India conducted studies which show that 90% of the road accidents that occur are due to drowsiness especially in the late hours of the day [4]. The AAA Foundation for Traffic Safety in the United States, in its

2015 Drowsy Driving Fact Sheet found that from 2009 to 2015, the proportion of registered drivers that admitted to driving dozy (in the last 30 days) had stayed substantially constant, at about 30%. This study states that around 97% of American drivers believe it is slightly or absolutely offensive for anybody to drive when they are too sleepy to even keep their eyes open [5] - [6]. On the other hand, nearly 31.5 percent of all legal drivers admitted to having driven their vehicles when they were so exhausted that it was a struggle to keep their eyes open in the past month. A little over 22.3% of licensed drivers acknowledged the fact that they did this more than once, and about 3.5 percent of registered drivers admitted to doing this fairly regularly [3].

The eye-state information as processed by the existing systems is done by computing the percentages of regularity of eye closure, the eye closure itself and the duration of eye closure. The other systems join this data with facial expressions, yawning, and head-movement [8]. Alternatively, focusing on sleep monitoring, is Polysomnography (PSG) which is a system used for diagnosis of sleep disorders. PSG is built on biomedical multi-signal achievement and processing, for example: Electrocardiogram (ECG), Electroencephalogram (EEG), electrodermal response and breathing [12]. This is one of the most prevalent devices used for medical purposes since these biomedical indications offer critical data regarding the physical reaction during the various stages of slumber. Hence, quite a few of these biomedical signals are being projected as methods of gauging drowsiness recognition, the most commonly employed being respiration, EEGs [13] - [14] as well as ECGs. An Electrocardiogram is quite a weak signal and a car atmosphere is electromagnetically very noisy which makes detection of signals to be quite challenging. Heart rate variability (HRV) is used as an exhaustion gauge for sportspersons. The HRV obtained from the ECG is being used for determining sleep stages in Polysomnography [15]. Tracking accuracy has been a major challenge in most of the existing Drowsiness Detection Systems. Systems with auxiliary functionality are very few. Therefore, we aim to develop a system of high accuracy with an additional speed limit suggestion service based on the driver's sentiment.

## II. PROPOSED METHODOLOGY

Our proposed system for drowsiness recognition is based on variations in expressions seen on the faces of the drivers.

The system monitors the driver's face using a camera that will clearly cover his/her entire face and processes the video images obtained in real-time so as to categorize the driver's state of awareness as sleepy or awake [9].

Manuscript received on May 12, 2021.

Revised Manuscript received on May 19, 2021.

Manuscript published on May 30, 2021.

\* Correspondence Author

**Kavyashree Subramonian\***, Department of Information Technology, Sri Venkateswara College of Engineering, Sriperumbudur (Tamil Nadu), India. Email: [2017it0872@svce.ac.in](mailto:2017it0872@svce.ac.in)

**Dr. G. Sumathi**, Department of Information Technology, Sri Venkateswara College of Engineering, Sriperumbudur (Tamil Nadu), India. Email: [gsumathi@svce.ac.in](mailto:gsumathi@svce.ac.in)

It further recommends a safe speed limit based on the driver's current emotional state [7]. Our paper evaluates the capability of this upcoming system to alert the driver of the vehicle when the first symptoms of fatigue appear. Hence in a car environment, this is an extremely valuable system without any vulnerabilities as it provides timely warnings that ensure no loss of life.

### III. IMPLEMENTATION

To facilitate high efficiency and accurate driver drowsiness system functionalities, it is divided into:

1. Dataset Collection
2. CNN Architecture
3. Face Detection in the image
4. Classification

#### A. Dataset Collection

The dataset used for this model is self-made. We came up with a script that captures the eye-states through the use of a camera and then proceeds to store it in the hard drive. This comprises the dataset. They are classified into labels namely 'Open' and 'Closed'. The collected information was cleaned manually by filtering out the redundant images. The dataset consists of over 7000 photographs of people's eyes as observed under various states of illumination. The final weights and model architecture file "models/cnnCat2.h5" have been attached after training the model on our dataset.

#### B. CNN Architecture

Convolutional Neural Networks (CNN) have been used to build this system. This is a special type of deep neural network which works amazingly well for the classification of the images taken. A CNN consists of 3 layers namely, an output layer, an input layer, and a hidden layer (having multiple layers). A filter that performs 2D matrix multiplication is used to execute a convolution operation on the layers and also the filter.

Following are the layers in the CNN model architecture:

1. Convolutional layer; 32 nodes, kernel size 3
2. Convolutional layer; 64 nodes, kernel size 3
3. Fully connected layer; 128 nodes

The last layer is a completely connected layer having 2 nodes. A rectified linear activation function (ReLU) is used in all the layers barring the output layer in which Softmax is used. CNN is mainly used for the process of Sentiment Analysis.

#### C. Face Detection in the image

The image has to be converted to grayscale format because the OpenCV algorithm for object recognition only accepts grey pictures as its input. Colour information is not required for detection of objects. In this method we are using the Haar Cascade Classifier to detect faces.

#### D. Classification

We are making use of the CNN classifier for calculating eye-states. Certain operations need to be performed to input the image into the model because it requires precise proportions to begin with. Firstly, the colour picture is converted into grayscale format by the use of `r_eye=cv2.cvtColor(r_eye,cv2.COLOR_BGR2GRAY)`.

The picture is then resized to a size of 24\*24 pixels using the command `cv2.resize(r_eye, (24,24))`, since the model was trained on size 24\*24-pixel pictures. Next, our data is normalized for better convergence using `r_eye = r_eye/255` (All values are between 0-1). The proportions are expanded to feed into the classifier. Our model was then loaded using `model= load_model('models/cnnCat2.h5')`. Finally, the states of each of the eyes with our model are predicted with the command `lpred = model.predict_classes(l_eye)`. If value of `lpred[0] = 1`, it indicates that the eyes are open. If value of `lpred[0] = 0` then, it indicates that the eyes are shut.

### IV. FACE DETECTION AND RECOGNITION ALGORITHMS

- 1: Divide dataset in N folds
- 2: **for** I = 1 to N
- 3: Use N-1 as validation set
- 4: Use the remaining as test set
- 5: Divide the validation set in N-1
- 6: **for** j = I to N - 1 **do**
- 7: Use N-2 to train the classifier
- 8: Use the remaining to evaluate the classifier
- 9: **end for**
- 10: Select the classifier with best AUC
- 11: Build the classifier selected with the validation set
- 12: Store the predictions in the test set
- 13: **end for**
- 14: Estimate the mean performance metrics in the test set

#### A. Haar Cascade Classifier

The Haar Cascade is an ML entity recognition algorithm used to identify objects in picture or film.

The 4 stages of this algorithm are:

1. Haar Feature Selection
2. Creation of Integral Images
3. Adaboost Training
4. Cascading Classifiers

Multiple positive pictures of faces and negative pictures without faces are needed by the algorithm to train the classifier for face detection. The features are later extracted from it [10].

A *Haar feature* studies adjoining rectangular regions at a particular position in a detection window. It then adds up the concentration of the individual pixels in each of these regions and computes the difference between these sums.

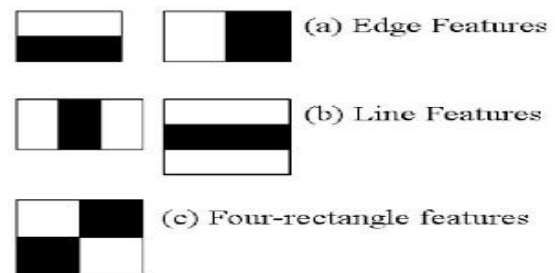


Fig.2 Different features in a detection window



Computations are accelerated using *Integral Images*. But among all the deliberate features, majority of them are inapt. For instance, take the images shown below. The upper row displays 2 suitable features. The first feature that is selected focuses on the property that areas around the eyes are mostly darker than the regions surrounding the nose and cheeks. The next selected character relies on the property that the bridge of the nose is lighter than the eyes. But the same windows applying on cheeks or any other area is irrelevant.

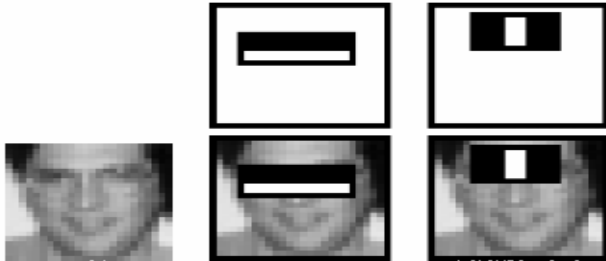


Fig.3 Appropriate feature selection

*Adaboost* helps in the selection of prime features. These are used to choose the best characters and train the classifiers that make use of them [11]. The algorithm builds a “strong” classifier as a direct blend of simple weighted “weak” classifiers.

A window of the target size is moved over the input image, during the detection phase. The Haar features are computed for each subdivision of the image. The variance is then equated to a trained threshold that isolates objects from non-objects since every Haar feature is just a “weak classifier” (its quality of detection being somewhat more advanced than random guesswork). To describe an object with sufficient accuracy a large number of Haar features are needed and are consequently ordered into *cascade classifiers* so as to create a robust classifier.

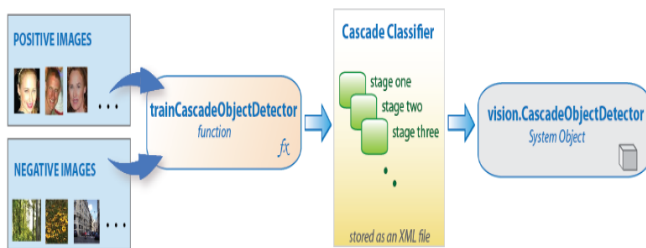


Fig.4 Cascade Classifier

A collection of stages makes up a cascade classifier. In this, every stage is a collection of learners. Simple classifiers are poor learners also known as *decision stumps*. Boosting is the technique that is employed to train all of the stages. It offers the capability to train a specific classifier by calculating the weighted mean of the choices made by the weak learners. The regions defined by the current location of the sliding window are labelled by each stage of the classifier as being either negative or positive. A *Positive* label shows that an entity was found while a *Negative* label shows that no entities were found. On the other hand, the classifier passes the section to the next stage if the label is positive. The sensor alerts us that an object has been located at the present window position once the ultimate stage classifies the region to be positive. These stages are intended to oppose negative samples as quickly as conceivable. It is presumed that the object of interest is not contained in the vast majority of

windows. On the contrary, true positives are occasional and are worth taking the time to validate.

When a positive sample is rightly classified, a *true positive* occurs.  
When a negative sample is wrongly classified as positive, a *false positive* occurs.  
When a positive sample is wrongly classified as negative, a *false negative* occurs.

Each of the stages in the cascade must have a low false negative rate for it to perform well. If an object is wrongly labelled as negative, the classification is terminated. This error cannot be rectified. Each stage might have a high false positive rate nevertheless. The overall false positive rate can be reduced by appending more stages but this also reduces the overall true positive rate. A collection of positive and negative images is essential for cascade classifier training. A collection of positive pictures with the areas of interest stated to be used as positive samples has to be provided. Objects of interest can be labelled with bounding boxes using the Image Labeller. The Image Labeller produces a table that can be used for positive samples. One must in addition provide a collection of negative pictures by means of which the function produces negative samples routinely. The total number of stages, character type, and other function arguments has to be declared in order to achieve sustainable detector accuracy.

### B. LBPH Face Reorganization

The Local Binary Patterns Histogram algorithm is extensively employed in facial recognition systems because of its computational straightforwardness and discriminatory control.

Following are the steps that must be carried out to accomplish this:

1. Creation of the Dataset
2. Face Acquisition
3. Feature Extraction
4. Classification

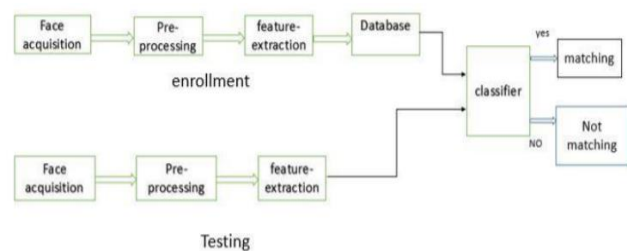


Fig.5 LBPH Algorithm

Consider an image having dimensions  $N \times M$ . We then go on to divide it into squares of  $m \times m$  dimension for each region.



Fig.6 Depicts division of an image with  $N \times M$  dimensions



The local binary operator is employed for each one of the regions. The LBP operator is demarcated in a window of dimensions 3x3.

$$LBP(x_c, y_c) = \sum_{p=0}^{P-1} 2^p s(i_p - i_c)$$

In the above formula, '(Xc, Yc)' is the central pixel which has an intensity of 'Ic'. The intensity of the adjacent pixel is 'In'.

Any pixel is equated to its 8 nearest pixels by making use of the median pixel value as the threshold. Shown below is the function being used:

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

The value of an adjoining pixel is set as 1 if it is more than or the same as the central value. Else it is set to 0. Consequently, a whole of 8 binary values are obtained from the 8 adjacent pixels. We get an 8-bit binary number on combining these values. This is converted to a decimal number for our ease of use. The obtained decimal number is called the pixel LBP value. It ranges from 0-255.

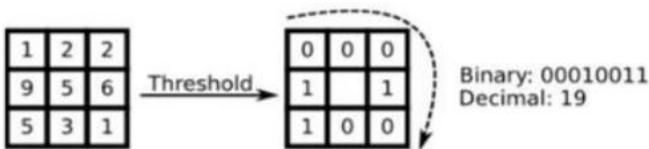


Fig.7 Pixel LBP value

It was later observed that a fixed neighborhood failed in encoding details that are changing in scale. The algorithm was upgraded to enable it to use varying values of radius and neighbors. It is then called a circular LBP.

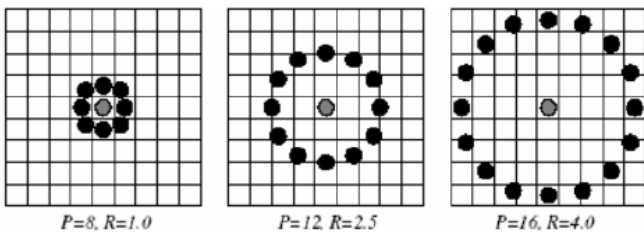


Fig.8 Circular LBP

Aligning a random number of neighboring pixels on a circle with a variable radius is the predominant idea. This enables the following neighborhoods to be captured.

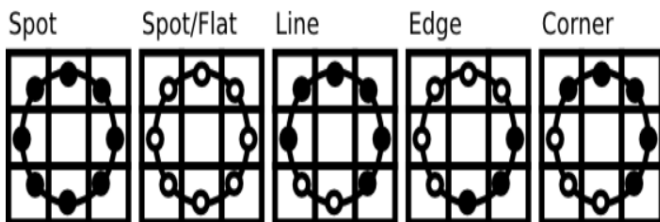


Fig.9 The neighbors aligned in a circle

The location of the neighboring pixel (Xp, Yp), p belonging to P, for a given point (Xc, Yc) can be calculated using the formula:

$$x_p = x_c + R \cos\left(\frac{2\pi p}{P}\right)$$

$$y_p = y_c - R \sin\left(\frac{2\pi p}{P}\right)$$

Here, P denotes the total number of sample points and R denotes the radius of the circle.

If and when a point on the circle doesn't parallel the image coordinates, it gets interpolated by bilinear interpolation

$$f(x, y) \approx [1 - x \ x] \begin{bmatrix} f(0,0) & f(0,1) \\ f(1,0) & f(1,1) \end{bmatrix} \begin{bmatrix} 1 - y \\ y \end{bmatrix}$$

The LBP operator is resistant to monotonic gray scale transformations.

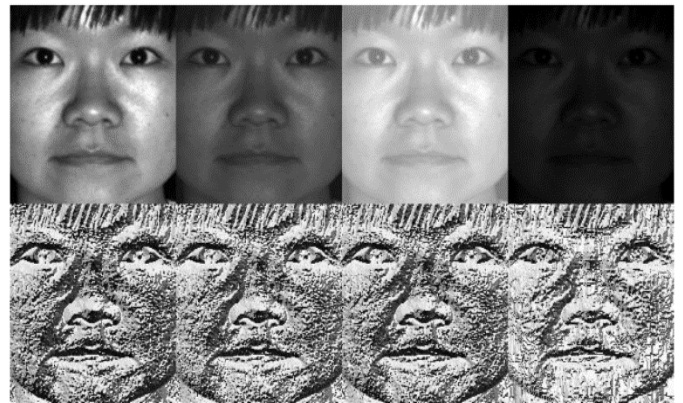


Fig.10 Gray scale transformations

Once the LBP value is produced, the number of comparable LBP values in the area are counted and a histogram of the region is created. After a histogram is created for each area, a final histogram is created by combining each of the histograms. This is called the feature vector of the image.

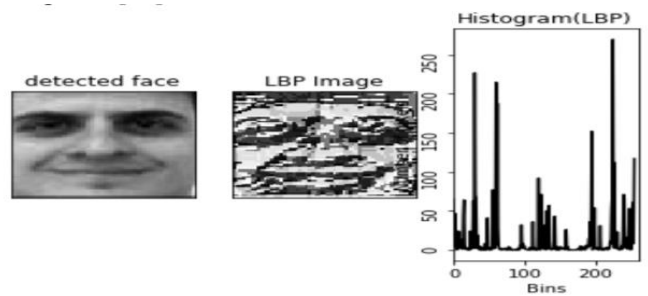


Fig.11 Generation of histogram

Now, the histograms of the test picture and the pictures in the database are compared and the picture with the most comparable histogram is returned. The Euclidean distance is computed by comparing the features in the dataset with the test picture features. The smallest distance between the test picture and original picture gives us the matching rate.

$$d(a, b) = \sqrt{\sum_{i=1}^n |a_i - b_i|^2}$$



The ID of a picture in the database is displayed if the test picture has been identified.



Fig.12 Output of the LBPH algorithm

### V. EXPERIMENTAL RESULTS

The developed model was trained with a predefined dataset. An accuracy of around 80 to 90 percent was achieved. Video stills were taken for training purposes.

Accuracy, %
72.45
80.12
75.26
81.39
88.01

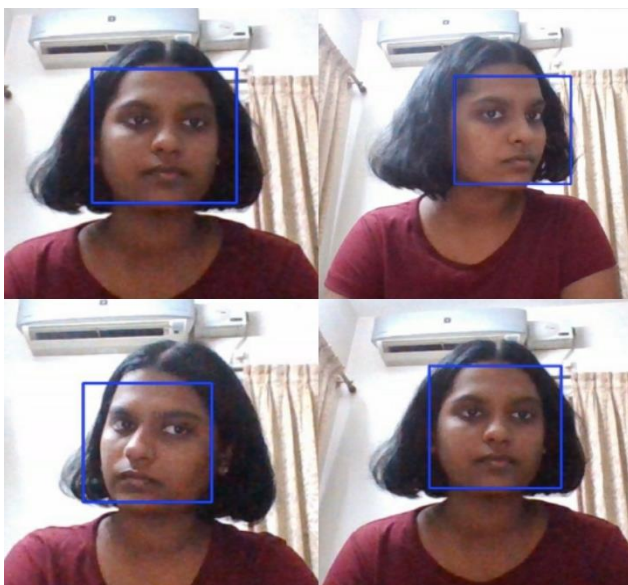


Fig.13 The video stills captured for training

Once the images were used for the training, real-time video stills of the driver are captured by the camera to observe the mental state of the driver. If the driver is drowsy an alarm is triggered. A safe speed limit is also recommended periodically depending on the driver's mood.

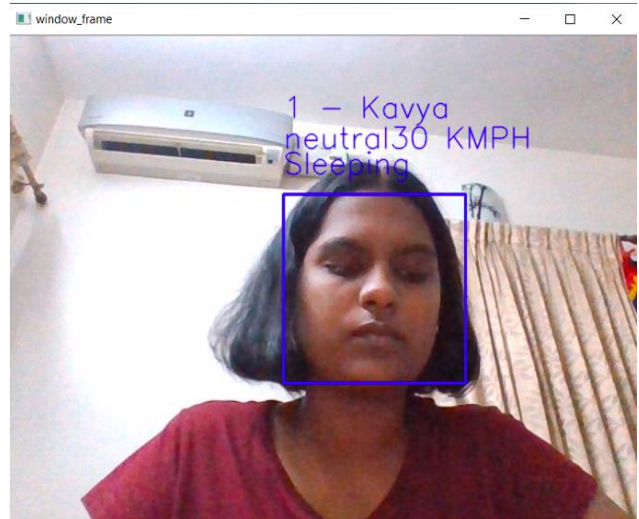


Fig.14 Detection of drowsiness followed by activation of the alarm



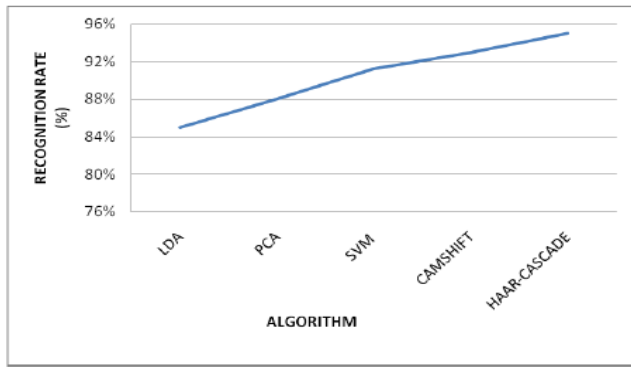
Fig.15 Speed limit recommendations based on driver's sentiment

The table below displays the performance analysis of various algorithms. The Haar Cascade algorithm provides the highest recognition rate of 95%.

ALGORITHM	LDA	PCA	SVM with binary	CAMSHIFT	HAAR-CASCADE
RECOGNITION RATE	85%	88%	91.2%	93%	95%

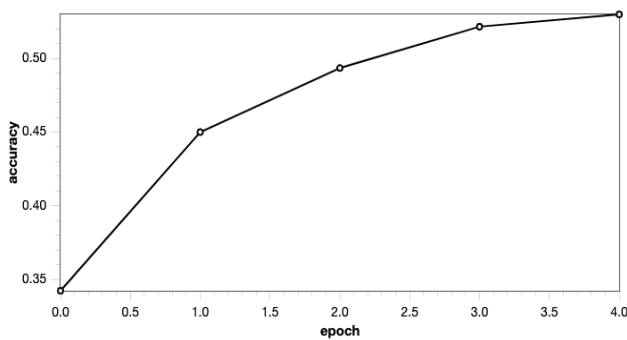
Fig.16 shows the graphical representation of performance analysis of various algorithms.[16]





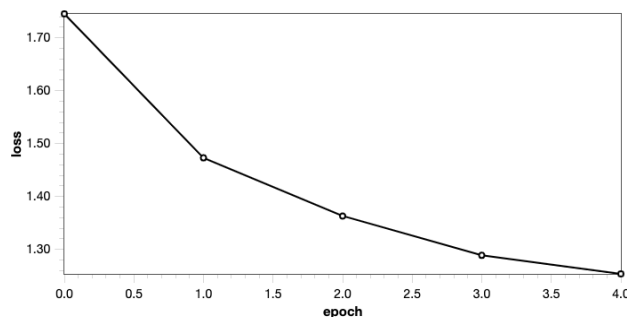
**Fig.16 Performance analysis of various algorithms**

Fig. 17 shows the analogy between the epoch and accuracy in the emotion training period. The accuracy values for epoch 0, 1, 2, 3 and 4 are 0.3423665052770908, 0.44992859382075306, 0.49343411473753873, 0.5215089344809956 and 0.5298338500132294.



**Fig.17 Epoch vs Accuracy in the emotion training period**

Fig. 18 shows the analogy between the epoch and loss in the emotion training period. The loss values for epoch 0, 1, 2, 3 and 4 are 1.745283404215114, 1.4730508763146768, 1.3632273462823643, 1.2887520346631076 and 1.2537492939851937.



**Fig.18 Epoch vs Loss in the emotion training period**

**VI. CONCLUSION**

An algorithm for monitoring the driver’s state of alertness has been put forth grounded on the examination of changing face features. The classification results are improved by the drowsiness detection algorithm through means of reducing the number of false positives owing to variations in measured eye blink rate which is related not to the drowsiness but to the body language or speech. In addition to this, the model also recommends a safe speed limit based on the driver’s present emotional state. All these features together make it an incredibly versatile system. This algorithm is therefore extremely well equipped and invaluable in preventing any

accidents and loss of life that may occur when people are driving dozy

**REFERENCES**

1. J. D. Slater, “A definition of drowsiness: One purpose for sleep?” *Med. Hypotheses*, vol. 71, no. 5, pp. 641–644, Nov. 2008. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0306987708002946>
2. “Drowsy driving and automobile crashes,” *Nat. Center Sleep Disorders Res./Nat.HighwayTrafficSaf.Admin.ExpertPanelDriverFatigueSleepiness*, Tech. Rep. DOT-HS-808-707, 1998, vol. 808, p. 707.
3. L. S. Arnold and B. C. Tefft, “Prevalence of self-reported drowsy driving, United States: 2015,” *AAA Found. Traffic Saf.*, Washington, DC, USA, Tech. Rep., 2015. [Online]. Available: <https://www.aaafoundation.org/sites/default/files/PrevalenceOfSelf-ReportedDrowsyDrivingReport.pdf>
4. *Violence and Injury Prevention and World Health Organization, Global Status Report on Road Safety 2015*, World Health Organization, Geneva, Switzerland, 2015.
5. P. Jackson, C. Hilditch, A. Holmes, N. Reed, N. Merat, and L. Smith, *Fatigue and Road Safety: A Critical Analysis of Recent Evidence*. London, U.K.: Department of Transport, 2011, no. 21.
6. B. C. Teff, “Acute sleep deprivation and risk of motor vehicle crash involvement,” *AAA Found. Traffic Saf.*, Tech. Rep. Dec. 2016. [Online]. Available: <https://trid.trb.org/view.aspx?id=1436965>
7. A. D. McDonald, J. D. Lee, C. Schwarz, and T. L. Brown, “Steering in a random forest: Ensemble learning for detecting drowsiness-related lane departures,” *Hum. Factors*, vol. 56, no. 5, pp. 986–998, Aug. 2014.
8. W. W. Wierwille, S. S. Wreggit, C. L. Kirm, L. A. Ellsworth, and R. J. Fairbanks, “Research on vehicle-based driver status/performance monitoring: development, validation, and refinement of algorithms for detection of driver drowsiness. Final report,” *U.S. Dept. Transp., Nat. Highway Traffic Saf. Admin.*, Washington, DC, USA, Tech. Rep. DOTHS-808-247, 1994.
9. T. D’Orazio, M. Leo, C. Guaragnella, and A. Distanto, “A visual approachfordriverinattentiondetection,” *PatternRecognit*, vol.40,no.8, pp.2341–2355,Aug.2007.[Online].Available:<http://linkinghub.elsevier.com/retrieve/pii/S0031320307000544>
10. J. Jo, S. J. Lee, K. R. Park, I.-J. Kim, and J. Kim, “Detecting driver drowsiness using feature-level fusion and user-specific classification,” *Expert Syst. Appl.*, vol. 41, no. 4, pp. 1139–1152, Mar. 2014.
11. S. J. Lee, J. Jo, H. G. Jung, K. R. Park, and J. Kim, “Real-time gaze estimator based on driver’s head orientation for forward collision warning system,” *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 1, pp. 254–267, Mar. 2011. [Online]. Available: <http://ieeexplore.ieee.org/document/5688323/>
12. A.G.Corraea,L.Orosco,andE.Laciar,“Automaticdetectionofdrowsiness in EEG records based on multimodal analysis,” *Med. Eng. Phys.*, vol. 36, no. 2, pp. 244–249, Feb. 2014.
13. J.W.Fu, M.Li, and B.L.Lu,“ Detecting drowsiness in driving simulation based on EEG,” in *Proc. Auton. Syst.-Self-Org., Manage., Control*, 8th Int. Workshop Shanghai Jiao Tong Univ. Dordrecht, The Netherlands: Springer, 2008, pp. 21–28. doi: 10.1007/978-1-4020-8889-6\_3.
14. M. Hajinorozi, Z. Mao, T.-P. Jung, C.-T. Lin, and Y. Huang, “EEGbased prediction of driver’s cognitive performance by deep convolutional neural network,” *Signal Process., Image Commun.*, vol. 47, pp. 549–555, May 2016. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0923596516300832>
15. L. Schmitt, J. Regnard, M. Desmarests, F. Mauny, L. Mouroit, J.-P. Fouillot, N. Coulmy, and G. Millet, “Fatigue shifts and scatters heart rate variability in elite endurance athletes,” *PLoSOne*, vol.8,no.8,2013,Art.no.e71588.
16. Senthamizh Selvi.R, D.Sivakumar, Sandhya.J.S , Siva Sowmiya.S, Ramya.S , Kanaga Suba Raja.S “Face Recognition Using Haar - Cascade
17. Classifier for Criminal Identification” *International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878 Volume-7, Issue-6S5 April 2019.* [Online].



## AUTHORS PROFILE



**Kavyashree Subramonian**, is a final year student pursuing B.Tech in Information Technology in Sri Venkateswara College of Engineering, Sriperumbudur. She is currently doing an internship in the field of software engineering. Areas of interest – Web Development and Machine Learning.



**Dr. G. Sumathi**, obtained her B.E. degree in Electronics and Communication from Bharathidasan University, M.E. degree in Computer Science and Engineering from Regional Engineering College, Tiruchirappalli, and PhD in Computer Science and Engineering from National Institute of Technology, Tiruchirappalli. She is currently working as a Professor in the Department of Information Technology, Sri Venkateswara College of Engineering, Sriperumbudur. Her research interest includes Distributed Computing and Computer Networks.