

Driver Fatigue Detection using Mean Intensity, SVM, and SIFT

Saima Naz, Sheikh Ziauddin*, Ahmad R. Shahid

Department of Computer Science, COMSATS Institute of Information Technology, Islamabad (Pakistan)

Received 13 July 2017 | Accepted 23 September 2017 | Published 6 October 2017

ABSTRACT

Driver fatigue is one of the major causes of accidents. This has increased the need for driver fatigue detection mechanism in the vehicles to reduce human and vehicle loss during accidents. In the proposed scheme, we capture videos from a camera mounted inside the vehicle. From the captured video, we localize the eyes using Viola-Jones algorithm. Once the eyes have been localized, they are classified as open or closed using three different techniques namely mean intensity, SVM, and SIFT. If eyes are found closed for a considerable amount of time, it indicates fatigue and consequently an alarm is generated to alert the driver. Our experiments show that SIFT outperforms both mean intensity and SVM, achieving an average accuracy of 97.45% on a dataset of five videos, each having a length of two minutes.

KEYWORDS

Driver Fatigue Detection, Eye Detection, Scale Invariant Feature Transform, Support Vector Machine, Traffic Accidents.

DOI: 10.9781/ijimai.2017.10.002

I. INTRODUCTION

DRIVER fatigue is one of the major causes of vehicle accidents [1],[2]. According to an estimate, up to 35-45% of all accidents are caused by driver fatigue [3]. It is reported that 57% of the fatal truck accidents are caused due to driver fatigue and 50% of the truck drivers report that driving fatigue is the major cause for heavy truck crashes [4]. It makes it imperative to devise intelligent systems that can detect driver fatigue and warn the driver accordingly.

Fatigue can be caused by many reasons such as late-night driving, sleep deprivation, alcohol usage, driving on monotonous roads, medicine intake that causes drowsiness and tiredness or sleep disorders [3], [5]. It slows down reaction time, decreases awareness and impairs judgement while driving. The motivation behind studying the driver fatigue detection problem and alerting the driver when he/she is fatigued, is to decrease the human and financial cost.

Our work has the objective of implementing different techniques and evaluating the performance of each to know which technique is the most suitable for detecting fatigue. Detection can be done based on certain indicators of fatigue. From the literature, we see that there are a few basic symptoms of drowsiness that are feasible to detect using camera and image processing techniques. These symptoms include micro-sleep, bouncing movement of the head and yawning [1], [33]. The bouncing movement of the head can be due to other reasons as well, for example if the driver is listening to music or the road is bumpy. In addition, nodding or swinging of head is apparent after the driver is almost asleep and it might be too late to prevent an accident. Detecting yawning can be a bit misleading as one's mouth may be open for more than one reasons, such as the driver might be singing or talking to his co-passenger. Therefore, bouncing movement

of head and yawning are not reliable ways to detect driver fatigue. In this paper, we focus on micro-sleeps to detect fatigue and to prevent accidents. Micro-sleep is the drowsy state in which the driver closes his/her eyes for short intervals of time. He/she may wake up and then fall asleep again. Even this short duration of sleep episode may cause an accident, especially in congested areas. The most intuitive solution is to generate an alarm which will wake up the driver and hence help avoid the potential accident.

In the proposed scheme, the input is in the form of driver's videos. Frames are extracted from each video and eyes of the driver are localized using Viola-Jones algorithm. After eyes have been localized, they are classified either as open or closed using three different techniques which are Mean Intensity, Support Vector Machine (SVM) and Scale Invariant Feature Transform (SIFT). When the eyes are found closed in a specified number of frames, it is considered as micro-sleep and the alarm is generated to alert the driver. For the evaluation of results, we use accuracy, specificity and sensitivity as performance measures. The results show that SIFT has better performance as compared to the mean intensity and the SVM.

The rest of this paper is organized as follows. A review of existing research literature is given in Section II. In Section III, the working of the proposed scheme is explained. Experimental results of the proposed scheme and the performance evaluation is presented in Section IV. Finally, we conclude this paper in Section V.

II. RELATED WORK

Most of the related work in driver fatigue detection involves the use of computer vision and machine learning techniques, and their combination. The computer vision techniques used for this purpose include Template Based Matching, Feature Based Matching, Histograms of Oriented Gradients (HOG), Gabor Wavelet Transform, Circular Hough Transform, and Landmark Model Matching [6], [7], [8], [9], [10]. The machine learning techniques used in the past for fatigue

* Corresponding author.

E-mail address: sh.ziauddin@gmail.com.

detection include Markov Chain Framework, Fuzzy Logic, Support Vector Machines (SVM), Naive Bayes (NB), AdaBoost, Particle Swarm Optimization (PSO), and Neural Networks [11], [1], [12], [4]. In the following subsections, we take a brief overview of computer vision and machine learning techniques as have been used in literature.

A. Computer Vision Techniques

Use of non-invasive methods, such as making a video of the driver and alerting him/her on using cues that may help in anticipating the presence of a sleep pattern, can be a useful way to detect driver fatigue [13], [14]. PERCLOS (PERcentage of CLOSure of eyes) is a commonly used method for detection of driver fatigue [2], [15], [12]. It determines the percentage of eye closure by taking the number of frames in which driver's eyes are closed and dividing this by the total number of frames over a specified period of time. Different researchers have used different time windows for PERCLOS calculation such as 20 seconds [16], 30 seconds [36] or 3 minutes [37]. To avoid accidents, it is important to generate an alarm as early as possible when the first symptoms of fatigue are detected. In this paper, we take eye closure in consecutive frames as a measure of fatigue detection instead of PERCLOS. This helps us in generating an alarm much earlier (one second for our selected parameters) in case driver goes in micro-sleep state as compared to PERCLOS.

Localizing the eyes is the first key step towards achieving the task. Facial features in images have been localized by researchers using different techniques. These techniques can be categorized into two groups: feature-based and template-based techniques [7]. Template-based techniques use the object's shape for matching, while feature-based techniques use different geometric features and constraints for their working [1], [16].

Tock and Craw [17] used stored templates and thresholds for detection of regions of interest. For eye detection, the darkest pixel was used to reduce the computational cost. Pupil being the darkest part, is likely to contain the darkest pixel. Eriksson and Papanikolopoulos [1] used both feature-based matching and template-based matching techniques to determine the exact locations of different features. They used reduced regions of the image to detect eyes. They used the heuristic that the regions surrounding the eyes are darker than the other regions in the vicinity.

Khan and Mansoor [2] first detected the face using local Successive Mean Quantization Transform (SMQT) [18]. Next, they drew a square around the face with its center specified. To localize eyes, they divided the square into three parts, with the assumption that the eyes would lie in the top most part of the face. They assumed that the eyes would be open in the beginning and used the first frame in the real-time video to generate an on-line template for open-eyes, which could later be used to determine if the eyes were open or closed. The information regarding the state of eyes (opened or closed) in a sequence of frames was used to detect whether an alarm should be generated or not. They achieved an accuracy of up to 90%.

Brandt et al. [5] presented a visual surveillance system to monitor drivers' head motion and eye blinking pattern. Based on measured features, fatigue was detected by the system. They adopted a coarse to fine strategy to achieve their goal. First, they found the face using Haar wavelets and then they localized the eyes in the already detected face image.

Cherif et al. [19] used the measurement of gaze position to indicate the areas that attract the subject's attention in an image. A calibrated infrared light device was used to provide the horizontal and vertical eye movement. A polynomial transformation of higher order was used to model this mapping by using a mean square error criterion. This helped them to better choose the optimal order to correct the data.

Clement et al. [34] developed a fatigue detection system using output of different low pass and band pass filters on Electro-OculoGram (EOG) and Electro-Encephalogram (EEG) signals. They did not report their results in terms of alarm generation accuracy. Also, their system is not practical as drivers would resist wearing any specific hardware gear while driving on the road.

B. Machine Learning Techniques

Machine Learning techniques have also been increasingly used to detect micro-sleeps owing to their ability to identify patterns. The techniques learn the patterns of closed eyes and open eyes and separate them into two classes. In this section, we discuss machine learning techniques used in the literature to address driver fatigue detection problem.

Dong and Wu [12] combined different cues, such as PERCLOS, head nodding frequency, slouching frequency and Postural Adjustment (PA) for better performance. They used machine learning techniques such as Support Vector Machines and Naive Bayes for classification. The Landmark Model Matching (LMM) [9], [20] was used, which in turn used Particle Swarm Optimization (PSO) [11], [21], [22] to detect the face in the image. Using different cues and fusing them using fuzzy logic achieved better performance than using PERCLOS alone. The use of fused cues reduced the classification error from 15.2% to 12.7%.

Coetzer and Hancke [23] presented a driver fatigue monitoring system. They used three different techniques for classification namely AdaBoost, SVM and Artificial Neural Networks. Devi and Bajaj [4] presented a fatigue detection system. They localized eyes and then tracked the eyes. Their approach also involved yawning detection. Fuzzy inference system was used to detect yawning and micro-sleeps.

Bagci et al. [24] used Markov chain framework to determine whether the eyes are open or closed. The method can be used in monitoring the drivers' alertness as well as in applications which require non-intrusive human computer interactions. Their eye detection and eye tracking algorithms were based on the color and geometrical features of the human face.

San et al. [35] used a deep generic model (DGM) and support vector machine to detect driver fatigue. They compared the performance of power spectrum density features-based SVM and deep generic model-based SVM and found that DGM-based SVM performed better in terms of sensitivity, specificity and accuracy. One drawback of their scheme is that the data is collected using EEG signals which reduces the practicality of the system.

III. PROPOSED SCHEME

In this paper, we use multiple techniques involving image processing/computer vision and machine learning. Before going into the details of the techniques used, first we give a high-level overview of the scheme. A schematic representation of the proposed scheme is given in Fig. 1. In Fig. 1, we show that first we read frames from the input video of the driver. For video generation, the driver is sitting on the driving seat and the video is captured using a single camera which is placed on the car's dashboard behind the steering wheel. In most of the past work, the camera is placed directly in front of the driver at eyes level which is obviously not practical as it obstructs the driver's view of the road ahead. In our dataset generation, we have placed the camera much lower than the eye level of the driver in such a way that it neither obstructs the driver's view of the scene, nor the steering wheel blocks the way between the camera and the driver. After reading the frames from the video, we detect the eyes in every tenth frame using Viola-Jones detector [25], [8]. Once the eyes have been detected, the next step is to classify the eyes into open or closed. The pseudo-code given in Fig. 2 further elaborates the high level working of the proposed scheme.

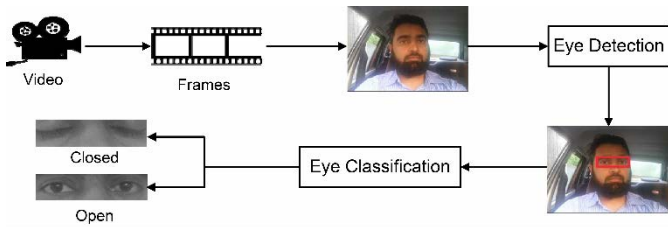


Fig. 1. A high-level overview of the proposed scheme.

```

Begin
  Input driver's video
  ClosedEyeCount := 0
  For every tenth frame in the video Do
    Localize eyes
    Check status of eyes
    If eyes are closed Then
      ClosedEyeCount := ClosedEyeCount + 1
    Else If eyes are open Then
      ClosedEyeCount := 0
    End If
    If ClosedEyeCount == 3 Then
      Generate Alarm
      ClosedEyeCount := 0
    End If
  End For
End

```

Fig. 2. Pseudocode for the proposed alarm generation algorithm.

All processing is done in every tenth frame if three consecutive observations are found where eyes are closed in each observation, and then an alarm is generated to alert the driver. On a video made at 30 frames/sec, this setting corresponds to one second of continuous eye closure which we feel is a reasonable trade-off. If an alarm is generated at a shorter eye closure interval than this, it results in many false positives as some blinks might be detected as micro-sleeps. On the other hand, if we generate alarm at a longer eye closer interval, then it might be too late and an accident might have already occurred before the driver is alerted and he/she can take some appropriate action (e.g., applying the brakes).

Next, we describe the techniques we have used in this paper. As mentioned earlier, we use three techniques namely Mean Intensity, SVM, and SIFT. In the three sub-sections, we describe each of these techniques in more detail.

A. Mean Intensity

Mean intensity is a simple statistical approach used for the classification of open and closed eyes in this paper. When an eye is open, the pupil and the iris are visible. Due to presence of these low intensity regions, the mean intensity of the closed eye image is low as compared to that for an open eye. On the other hand, when eyes are closed, pixels with higher intensity are more frequent, and when the eyes are open, due to the presence of iris, pixels with lower intensity increase.

For each input image (eye region only, detected using Viola-Jones algorithm), we find its mean intensity value and compare it with a threshold T . If mean intensity of the input image is greater than T , then we classify it as an open eye image, otherwise we consider it a closed eye image. The threshold T is selected empirically. A single threshold is selected and tried on all videos instead of selecting five separate thresholds for five videos in our dataset. Using separate thresholds would result in unrealistically higher accuracy of mean intensity

results.

B. Support Vector Machine (SVM)

In literature, a number of classification techniques have been used for the task. In this paper, we use SVM for the classification task. SVM being the optimal binary classifier suits well to the problem in hand. SVM is a supervised model for classification which is widely used in Machine Learning and Pattern Recognition [38], [39], [40]. The model is first trained by providing both positive and negative training examples. As SVM is a binary classifier, all training examples belong to one of the two categories. SVM models these training examples as points in space and marks a decision surface in such a way that the gap between the decision surface and any of the two categories is as large as possible. The examples that arrive for testing are mapped to the already created model and a decision is made for each new example whether it belongs to category 1 or category 2 depending upon which side of the border that new example lies.

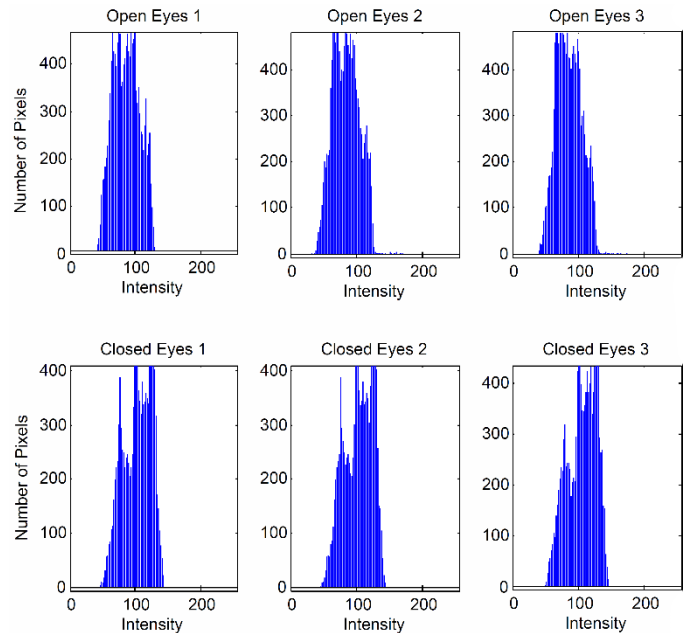


Fig. 3. Histograms of randomly picked three open and three closed eye images.

For the proper working of any classifier, the selection of appropriate features is really important. As we have already mentioned in Section 3.1, the intensity values of an eye image provide useful information regarding whether the eye is open or closed. We have used intensity histograms as features to be used with SVM. Fig. 3 displays histograms of six randomly picked eye images (three each for open and closed eyes). Though there are only subtle differences when these histograms are viewed visually but, as we will see in Section 4, SVM was able to detect these subtle differences in patterns of open and closed eye histograms and providing high classification accuracy based on the histogram features.

C. Scale Invariant Feature Transform (SIFT)

SIFT is one of the most popular algorithms in computer vision. It was presented by David Lowe in 2004 [26], and is patented by University of British Columbia. Since its birth, SIFT has widely been used for object detection and tracking, image registration, panorama stitching, robot localization and mapping, and dense correspondence across scenes. In this paper, we use SIFT to match eye images and classify them into open and closed. Next, we will briefly describe the working of SIFT algorithm. After that, we will explain how we have used SIFT in the proposed technique.

The SIFT algorithm comprises of the following four steps:

1. Scale space construction and extrema detection.
2. Keypoint localization.
3. Orientation assignment.
4. Keypoint description.

1) Scale Space Construction and Extrema Detection

The first step in SIFT is to create a scale space. This is done by creating a pyramid of octaves (set of images) by resampling the image at different sizes and smoothing the image at different scales. At each level of the pyramid, the size of the image is reduced to one-quarter of the size at lower level. For smoothing, the image is convolved with a Gaussian function as given below:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (1)$$

$$G(x, y, k\sigma) = \frac{1}{2\pi(k\sigma)^2} e^{-(x^2+y^2)/2k^2\sigma^2} \quad (2)$$

where I is the input image, x and y are the coordinates of the image, G is the Gaussian function, L is the resultant blurred image, σ is the smoothness value, and k is a constant that controls the amount of smoothing to be increased at each scale.

Next, Difference of Gaussian (DoG) matrices are generated by subtracting the Gaussian smoothed images at adjacent levels. Then potential features (keypoints) are selected by finding local extrema in a neighborhood of $3*3$ around the current pixel as well as around the corresponding pixels at the adjacent upper and lower scales.

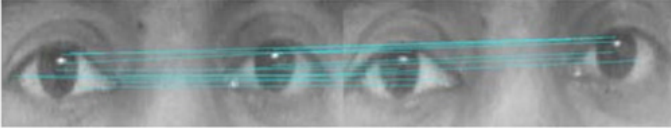


Fig 4. SIFT keypoint matching between the input image and the reference image. The figure is best viewed in color.

2) Keypoint Localization

There are too many candidate keypoints detected in the previous step. Some of these keypoints are unstable which are eliminated in this step. First, low-contrast candidates are eliminated. This is done by using the following second order Taylor series expansion:

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}} \mathbf{x} + \frac{1}{2} \mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2} \mathbf{x} \quad (3)$$

where D is the difference of Gaussian matrix, and $\mathbf{x} = [x, y, \sigma]^T$ is the candidate keypoint. The derivative of the above equation is taken to get the localized keypoint $\hat{\mathbf{x}}$:

$$\hat{\mathbf{x}} = \frac{\partial^2 D^{-1} \partial D}{\partial \mathbf{x}^2 \partial \mathbf{x}} \quad (4)$$

If the value of $D(\hat{\mathbf{x}})$ in (3) is below a threshold, this indicates that the candidate keypoint is a low contrast one and therefore it is discarded. In addition to low contrast candidates, candidates along the edges are also discarded and only those candidates are retained which are on the corners. This is done by computing a $2*2$ Hessian matrix of DoG and applying a threshold on its trace and determinant ratio.

3) Orientation Assignment

In this step, we find the magnitude m and the direction θ of each keypoint from the smoothed image $L(x, y)$ as follows:

$$m(x, y) = \sqrt{\frac{(L(x+1, y) - L(x-1, y))^2}{+ (L(x, y+1) - L(x, y-1))^2}} \quad (5)$$

$$\theta(x, y) = \tan^{-1} \left(\frac{L(x+1, y) - L(x-1, y)}{L(x, y+1) - L(x, y-1)} \right) \quad (6)$$

Next, in a neighborhood of each keypoint, a weighted histogram of orientations having 36 bins is created. The bin that gets the highest weighted sum is selected as the orientation of that keypoint.

4) Keypoint Description

In this step, a $16*16$ neighborhood around each keypoint is taken and, for each pixel in that neighborhood, the magnitude and orientation is calculated. Each $16*16$ block is further divided into 16 sub-blocks of $4*4$ size. For each sub-block, 8-bin orientation histogram is created. These values are represented as a 128-dimensional vector (concatenation of 8 bin values for 16 sub-blocks) to form the keypoint descriptor.

For object matching, the keypoints from the input image are matched against all those in the database. The matching is done using the keypoint descriptors. For each keypoint, nearest neighbor is identified from the database where nearest neighbor is that descriptor from the database which has the minimum Euclidean distance from the input keypoint descriptor.

In this paper, we perform SIFT keypoint extraction on all eye images. First, we have randomly selected two images (one each for open and closed eyes) to be used as reference images. Then for each video, SIFT keypoints from each eye image are matched with those from the two reference images. If the feature matching score is higher between the input image and the open eye reference image as compared to that between the input image and the closed eye reference image, the input image is considered as an open eye image. On the other hand, if the input image and the reference closed eye image have more keypoints in common, the input image is classified as a closed eye image.

Fig. 4 shows an example of an open eye input image where the keypoints matched with the reference open eye image are displayed. The image on left is the input image while that on right is the reference image. As it can be seen, there are many points which are matched by the SIFT keypoint matcher. On the other hand, when the same input image is matched with the sample closed eye image, only one keypoint matches between the pair of images.

IV. EXPERIMENTAL EVALUATION

A. Video Dataset

For our experiments, we have created a dataset by placing the camera inside the vehicle on the dashboard behind the steering wheel. The camera is placed in such a way that it does not occlude driver's view of the scene. Also, the steering wheel is not hiding driver's face from the camera. We collected our data under different conditions of lighting and drivers' clothing, and with varying degrees of cloud cover and natural daylight. It contains five videos (V1-V5) of a single driver, which are captured at the rate of 30 frames per second using a two mega-pixels smart phone camera. Each video is of approximately two minutes duration.

For eye classification and alarm generation, we processed every tenth frame of each video so that the frames with normal blink are not included in the processed frames. This setting results in approximately 360 frames to be processed for each video ($30/10 * 120$). We generated the ground truth by visually looking at each frame and deciding whether the eyes are open or closed in that particular frame. Table I provides further description of our dataset.

B. Eye Detection Results

For eye detection, we have used the well-known Viola-Jones detector [25], [8]. Viola-Jones object detection framework uses rectangular Haar features and creates an integral image from these features. As a learning algorithm, they used a variant of AdaBoost which performs feature selection and trains the classifier. Finally, the classifier cascading is done by doing the processing in different stages. At each stage, sub-windows are classified as may be face or definitely non-face. All sub-windows which are classified as definitely non-faces are discarded and hence reducing the number of sub-windows at each stage which results in decreased computational complexity of the algorithm.

Viola-Jones detector detected eyes correctly in 1806 out of 1819 frames resulting in a correct detection rate of 99.3% as shown in Table I. We discarded those 13 frames where eyes were not correctly detected and continued our experiments for eye classification and alarm generation using the remaining 1806 frames.

TABLE I. SUMMARY OF DATASET AND THE GROUND TRUTH

	V1	V2	V3	V4	V5	Total
No. of Frames	3590	3610	3640	3600	3750	18190
No. of Processed Frames	359	361	364	360	375	1819
Tracking Failures	1	0	3	3	6	13
No. of Eye Images	358	361	361	357	369	1806
No. of Instances with Closed Eyes	153	139	128	134	130	684
No. of Instances with Open Eyes	205	222	233	223	239	1122

C. Evaluation Metrics

For the task of eye classification, we have used three evaluation metrics namely *accuracy*, *specificity*, and *sensitivity*. The formulae for calculating these values are given as under:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

$$Specificity = \frac{TN}{TN + FP} \quad (8)$$

$$Sensitivity = \frac{TP}{TP + FN} \quad (9)$$

where TP (True Positives) are those cases where a closed eye is classified as closed and TN (True Negatives) are those when an open eye is classified as open. Similarly, FP (False Positives) corresponds to the cases where an open eye is classified as closed, and FN (False Negatives) to the cases where a closed eye is classified as open.

D. Eye Classification Results

We use mean intensity, SVM and SIFT to classify eyes as open or closed. For mean intensity and SIFT, we test the algorithm on all video frames and report our results here. On the other hand, for SVM, we use 5-fold cross validation where four of the five videos are used for training and the remaining one for testing in each iteration. Tables II, III, and IV show eye classification results for mean intensity, SVM, and SIFT, respectively. The values of False Positive (FP) Rate and False Negative (FN) Rate used in tables are calculated as follows:

$$FP\ Rate = \frac{FP}{FP + TN} \quad (10)$$

$$FN\ Rate = \frac{FN}{FN + TP} \quad (11)$$

From Tables II, III, and IV, we can see that SIFT provides highest accuracy among the three techniques.

During our experiments, we observed that the number of SIFT key points are much larger on average for the open eyes as compared to that for the closed eyes. Using this observation, we perform another experiment where we have classified the eyes based upon only the number of key points detected by the SIFT algorithm instead of matching those key points against the reference images. We have named this part of SIFT algorithm as SIFT-K. The eye classification results using SIFT-K are shown in Table V. The results of SIFT and SIFT-K are almost equal but SIFT-K has the advantage that it has better running time than that of SIFT.

Table VI compares the eye classification results obtained by mean intensity, SVM, SIFT, and SIFT-K. SIFT-K has the highest accuracy and specificity values while SIFT has highest sensitivity among all techniques.

TABLE II. EYE CLASSIFICATION RESULTS USING MEAN INTENSITY

	V1	V2	V3	V4	V5	Overall
FP Rate (%)	0.49	1.35	2.58	31.84	0.42	7.31
FN Rate (%)	5.88	2.16	0.78	0.00	6.15	3.07
Accuracy (%)	97.21	98.34	98.06	80.11	97.56	94.3
Specificity (%)	99.76	99.39	98.88	82.68	99.82	92.69
Sensitivity (%)	96.08	98.44	99.39	100.0	95.2	96.93

TABLE III. EYE CLASSIFICATION RESULTS USING SVM

	V1	V2	V3	V4	V5	Overall
FP Rate (%)	0.49	18.47	2.15	2.69	0.42	4.81
FN Rate (%)	3.27	28.06	6.25	0.00	6.15	8.77
Accuracy (%)	98.32	77.84	96.4	98.32	97.56	93.69
Specificity (%)	99.76	90.74	99.07	98.78	99.82	95.19
Sensitivity (%)	97.84	78.09	95.05	100.0	95.2	91.23

TABLE IV. EYE CLASSIFICATION RESULTS USING SIFT

	V1	V2	V3	V4	V5	Overall
FP Rate (%)	2.93	4.05	4.72	3.14	2.09	3.39
FN Rate (%)	0.00	0.72	3.13	2.24	1.54	1.46
Accuracy (%)	98.32	97.23	95.84	97.2	98.1	97.34
Specificity (%)	98.55	98.13	97.92	98.57	99.11	96.61
Sensitivity (%)	100.0	99.48	97.54	98.32	98.81	98.54

E. Alarm Generation Results

In a particular frame, if the eyes are classified as closed, this can mean one of the two things: either the driver is blinking his/her eyes, or he/she is in a state of micro-sleep. To differentiate between the two conditions, we used a threshold of three i.e. if the eyes are closed in three or more consecutive processed frames, it is considered a microsleep (see Fig. 2) and results in alarm generation. All the alarms generated using the proposed techniques are then compared against those in the ground truth to find out the missed alarms as well as the false alarms.

Tables VII, VIII, IX, and X show results for alarm generation using mean intensity, SVM, SIFT, and SIFT-K, respectively.

TABLE V. EYE CLASSIFICATION RESULTS USING SIFT-K

	V1	V2	V3	V4	V5	Overall
FP Rate (%)	1.46	3.15	2.58	0.9	2.51	2.14
FN Rate (%)	1.96	2.16	5.47	4.48	2.31	3.22
Accuracy (%)	98.32	97.23	96.40	97.76	97.56	97.45
Specificity (%)	99.28	98.55	98.88	99.60	98.93	97.86
Sensitivity (%)	98.71	98.44	95.68	96.62	98.22	96.78

TABLE VI. OVERALL EYE CLASSIFICATION RESULTS FOR MEAN INTENSITY, SVM, SIFT AND SIFT-K

Technique	Accuracy (%)	Specificity (%)	Sensitivity (%)
Mean Intensity	94.3	92.69	96.93
SVM	93.69	95.19	91.23
SIFT	97.34	96.61	98.54
SIFT-K	97.45	97.86	96.78

Finally, Fig. 5 shows and compares incorrect alarms generated by all techniques. As we can see, SIFT-K provides the best results here resulting in nine incorrect alarms (either missed or falsely generated). Though we have considered both false alarms and the missed alarms, it is obvious that missing an alarm is much more critical than generating a false alarm. In terms of missed alarms also, SIFT-K has the best performance followed by SIFT, SVM, and mean intensity, respectively.

TABLE VII. ALARM GENERATION RESULTS FOR MEAN INTENSITY

Criteria	V1	V2	V3	V4	V5	Overall
Number of Micro-sleeps	37	31	30	29	30	157
Alarms Generated	34	31	31	56	28	180
Correctly Detected Alarms	34	30	29	29	27	149
Missed Alarms (False Negatives)	3	1	1	0	3	8
False Alarms (False Positives)	0	1	2	27	1	31

TABLE VIII. ALARM GENERATION RESULTS FOR SVM

Criteria	V1	V2	V3	V4	V5	Overall
Number of Micro-sleeps	37	31	30	29	30	157
Alarms Generated	35	31	30	31	30	157
Correctly Detected Alarms	35	30	28	29	29	151
Missed Alarms (False Negatives)	2	1	2	0	1	6
False Alarms (False Positives)	0	1	2	2	1	6

TABLE IX. ALARM GENERATION RESULTS FOR SIFT

Criteria	V1	V2	V3	V4	V5	Overall
Number of Micro-sleeps	37	31	30	29	30	157
Alarms Generated	39	35	31	30	31	166
Correctly Detected Alarms	37	31	27	27	30	152
Missed Alarms (False Negatives)	0	0	3	2	0	5
False Alarms (False Positives)	2	4	4	3	1	14

TABLE X. ALARM GENERATION RESULTS FOR SIFT-K

Criteria	V1	V2	V3	V4	V5	Overall
Number of Micro-sleeps	37	31	30	29	30	157
Alarms Generated	37	33	31	28	31	160
Correctly Detected Alarms	36	31	29	28	30	154
Missed Alarms (False Negatives)	1	0	1	1	0	3
False Alarms (False Positives)	1	2	2	0	1	6

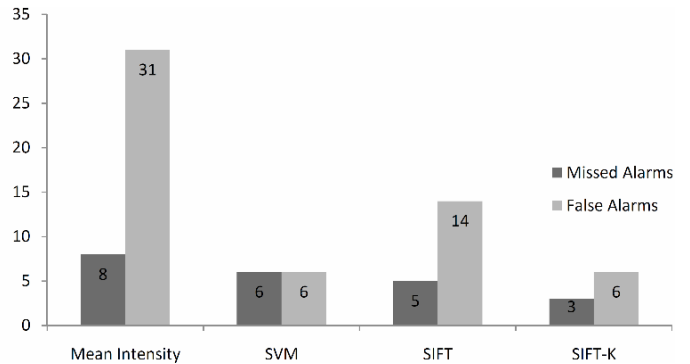


Fig. 5. Comparison of missed alarms and false alarms for different techniques.

F. Discussion on Results

In our experiments, we tried mean intensity, SVM, SIFT and SIFT-K for both eye classification and alarm generation. For eye classification, SIFT-K provided the best average accuracy (97.45%) followed by SIFT (97.34%). Though the average accuracies of mean intensity and SVM are also quite good (94.3% and 93.69%, respectively), it is the standard deviation where they are totally outperformed by SIFT and SIFT-K. SIFT and SIFT-K have standard deviations of 0.71 and 0.98, respectively. On the other hand, mean intensity has a standard deviation of 7.92 while SVM has 8.89. When we see the results of individual videos, we see that there is one video each for mean intensity (Video 4) and SVM (Video 2) where the accuracy is much lower than the other videos (see Tables II and III). SIFT and SIFT-K do not suffer from this variation. As we see from Tables IV and V, both techniques provide high accuracy for all the videos in the dataset. For SIFT, the minimum accuracy for any video is 95.84% while for SIFT-K, it is 96.4%. Overall, we can conclude that SIFT-K and SIFT perform much better as compared to the other two techniques.

V. CONCLUSION

In this paper, we presented a driver fatigue detection scheme. Eyes of the driver were detected in the first step and then it was analyzed whether the eyes were open or closed. If eyes remained closed for a specified amount of time then an alarm was generated to alert the driver. We used Viola-Jones algorithm to detect eyes and then three different approaches were used to classify open eyes and closed eyes i.e., mean intensity, SIFT and SVM. We saw that SIFT performed better for eye classification, alarm generation and fatigue detection as compared to the other tested techniques. In order to be more confident about our results, we need to enhance our dataset not just in terms of the number of drivers but also their characteristics. In future, we intend to create larger datasets with different subjects belonging to both the genders, with varying degrees of facial hair and other accessories such as glasses. The dataset would also be collected for the night time. Also, different camera locations on the dashboard would be experimented with evaluation of the best positions.

REFERENCES

- [1] M. Eriksson and N.P. Papanikolopoulos, "Driver fatigue: a vision based approach to automatic diagnosis," *Transportation Research Part C: Emerging Technologies*, vol. 9, no. 6, pp. 399–413, 2001.
- [2] M.I. Khan and A.B. Mansoor, "Real time eyes tracking and classification for driver fatigue detection," in *Image Analysis and Recognition*, pp. 729–738, Springer, 2008.
- [3] S.K. Lal and A. Craig, "Driver fatigue: electroencephalography and psychological assessment," *Psychophysiology*, vol. 39, no. 3, pp. 313–321, 2002.
- [4] M.S. Devi and P.R. Bajaj, "Fuzzy based driver fatigue detection," *Systems Man and Cybernetics (SMC)*, 2010 IEEE International Conference on, pp. 3139–3144, IEEE, 2010.
- [5] T. Brandt, R. Stemmer, and A. Rakotonirainy, "Affordable visual driver monitoring system for fatigue and monotony," *Systems, Man and Cybernetics*, 2004 IEEE International Conference on, pp. 6451–6456, IEEE, 2004.
- [6] I. Craw, H. Ellis, and J.R. Lishman, "Automatic extraction of face features," *Pattern Recognition Letters*, vol. 5, no. 2, pp. 183–187, 1987.
- [7] R. Brunelli and T. Poggio, "Face recognition: Features versus templates," *Pattern Analysis and Machine Intelligence*, IEEE Transactions on, vol. 15, no. 10, pp. 1042–1052, 1993.
- [8] P. Viola and M.J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.
- [9] R.S. Senaratne and S.K. Halgamuge, "Optimal weighting of landmarks for face recognition," *Journal of Multimedia*, vol. 1, no. 3, pp. 31–41, 2006.
- [10] J. Chen, Y. Lin, L. Wang, Z. Jiang, and G. Wang, "An algorithm for fatigue pattern detection based on parallel Gabor and 1-nearest neighbor," *Fourth International Conference on Digital Image Processing (ICDIP 2012)*, pp. 833410–833410, International Society for Optics and Photonics, 2012.
- [11] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," *Micro Machine and Human Science*, 1995. MHS'95, Proceedings of the Sixth International Symposium on, pp. 39–43, IEEE, 1995.
- [12] W. Dong and X. Wu, "Fatigue detection based on the distance of eyelid," *VLSI Design and Video Technology*, 2005. Proceedings of 2005 IEEE International Workshop on, pp. 365–368, IEEE, 2005.
- [13] Q. Ji, Z. Zhu, and P. Lan, "Real-time noninvasive monitoring and prediction of driver fatigue," *Vehicular Technology*, IEEE Transactions on, vol. 53, no. 4, pp. 1052–1068, 2004.
- [14] P. Smith, M. Shah, and N. da Vitoria Lobo, "Determining driver visual attention with one camera," *Intelligent Transportation Systems*, IEEE Transactions on, vol. 4, no. 4, pp. 205–218, 2003.
- [15] D.F. Dinges and R. Grace, "Perclous: A valid psychophysiological measure of alertness as assessed by psychomotor vigilance," *Federal Highway Administration. Office of motor carriers, Tech. Rep. MCRT-98-006*, 1998.
- [16] R. Senaratne, D. Hardy, B. Vanderaa, and S. Halgamuge, "Driver fatigue detection by fusing multiple cues," in *Advances in Neural Networks–ISNN 2007*, pp. 801–809, Springer, 2007.
- [17] D. Tock and I. Craw, "Tracking and measuring drivers' eyes," *Image and Vision Computing*, vol. 14, no. 8, pp. 541–547, 1996.
- [18] M. Nilsson, J. Nordberg, and I. Claesson, "Face detection using local smqt features and split up snow classifier," *Acoustics, Speech and Signal Processing*, 2007. ICASSP 2007. IEEE International Conference on, pp. II–589, IEEE, 2007.
- [19] Z.R. Cherif, A. Nait-Ali, J. Motsch, and M. Krebs, "An adaptive calibration of an infrared light device used for gaze tracking," *Instrumentation and Measurement Technology Conference*, 2002. IMTC/2002. Proceedings of the 19th IEEE, pp. 1029–1033, IEEE, 2002.
- [20] R. Senaratne, B. Jap, S. Lal, A. Hsu, S. Halgamuge, and P. Fischer, "Comparing two video-based techniques for driver fatigue detection: classification versus optical flow approach," *Machine Vision and Applications*, vol. 22, no. 4, pp. 597–618, 2011.
- [21] Y. Shi and R.C. Eberhart, "Empirical study of particle swarm optimization," *Evolutionary Computation*, 1999. CEC 99. Proceedings of the 1999 Congress on, IEEE, 1999.
- [22] A. Ratnaweera, S.K. Halgamuge, and H.C. Watson, "Self organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *Evolutionary Computation*, IEEE Transactions on, vol. 8, no. 3, pp. 240–255, 2004.
- [23] R.C. Coetzer and G.P. Hancke, "Eye detection for a real-time vehicle driver fatigue monitoring system," *Intelligent Vehicles Symposium (IV)*, 2011 IEEE, pp. 66–71, IEEE, 2011.
- [24] A. Bagci, R. Ansari, A. Khokhar, and E. Cetin, "Eye tracking using markov models," *Pattern Recognition*, 2004. ICPR 2004. Proceedings of the 17th International Conference on, pp. 818–821, IEEE, 2004.
- [25] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Computer Vision and Pattern Recognition*, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, pp. I–511, IEEE, 2001.
- [26] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [27] L.M. Bergasa, J. Nuevo, M.A. Sotelo, R. Barea, and M.E. Lopez, "Real-time system for monitoring driver vigilance," *Intelligent Transportation Systems*, IEEE Transactions on, vol. 7, no. 1, pp. 63–77, 2006.
- [28] G. Borghini, L. Astolfi, G. Vecchiato, D. Mattia, and F. Babiloni, "Measuring neurophysiological signals in aircraft pilots and car drivers for the assessment of mental workload, fatigue and drowsiness," *Neuroscience & Biobehavioral Reviews*, vol. 44, pp. 58–75, 2014.
- [29] A.A. Lenskiy and J.S. Lee, "Driver's eye blinking detection using novel color and texture segmentation algorithms," *International Journal of Control, Automation and Systems*, vol. 10, no. 2, pp. 317–327, 2012.
- [30] M. Sacco, R. Farrugia, et al., "Driver fatigue monitoring system using support vector machines," *Communications Control and Signal Processing (ISCCSP)*, 2012 5th International Symposium on, pp. 1–5, IEEE, 2012.
- [31] H. Shen and M. Xu, "Design and implementation of embedded driver fatigue monitor system," *2015 International Conference on Artificial Intelligence and Industrial Engineering*, Atlantis Press, 2015.
- [32] S. Singh, and N. P. Papanikolopoulos. "Monitoring driver fatigue using facial analysis techniques," *Intelligent Transportation Systems*, 1999. Proceedings. 1999 IEEE/IEE/JSAI International Conference on. IEEE, 1999.
- [33] A. Shabnam, B. Hariri, and S. Shirmohammadi. "Driver drowsiness monitoring based on yawning detection," *Instrumentation and Measurement Technology Conference (I2MTC)*, 2011 IEEE. IEEE, 2011.
- [34] F. S. C. Clement, A. Vashista, and M. E. Rane. "Driver Fatigue Detection System," *International Conference on Information Processing*, 2015.
- [35] P. P. San, S. H. Ling, R. Chai, Y. Tran, A. Craig and H. Nguyen, "EEG-based driver fatigue detection using hybrid deep generic model," *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Orlando, FL, 2016, pp. 800–803.
- [36] D. Prokhorov. *Computational intelligence in automotive applications*. Springer-Verlag, 2008.
- [37] S. Darshana, D. Fernando, S. Jayawardena, S. Wickramanayake and C. DeSilva, "Efficient PERCLOS and Gaze Measurement Methodologies to Estimate Driver Attention in Real Time," *2014 5th International Conference on Intelligent Systems, Modelling and Simulation*, Langkawi, 2014, pp. 289–294.
- [38] D. Pujari, J. R. Yakkundimath and A. S. Byadgi, "SVM and ANN Based

Classification of Plant Diseases Using Feature Reduction Technique,” International Journal of Interactive Multimedia and Artificial Intelligence, vol. 3, no. 7, pp. 6–14, 2016.

- [39] F. Kurtulmuş, I. Kavdir, “Detecting Corn Tassels using Computer Vision and Support Vector Machines,” Expert Systems with Applications, vol. 41, no. 16, pp. 7390-7397, 2014.
- [40] É. Niaf, R. Flamary, A. Rakotomamonjy, O. Rouvière and C. Lartizien, “SVM with feature selection and smooth prediction in images: Application to CAD of prostate cancer,” 2014 IEEE International Conference on Image Processing (ICIP), Paris, 2014, pp. 2246-2250.



Saima Naz

Saima Naz completed MS in Computer Science from COMSATS Institute of Technology in Islamabad, Pakistan. She is currently working as a Program Manager at iENGINEERING (Private) Limited in Islamabad. Her research interests include computer vision, image processing, pattern recognition and project management.



Sheikh Ziauddin

Sheikh Ziauddin is an assistant professor in department of computer science, COMSATS Institute of Information Technology in Islamabad, Pakistan. He did his PhD in Computer Science from Asian Institute of Technology in Bangkok, Thailand. He also completed Post-Doc from University of Nice in Nice, France. He is an author of more than 25 international conference and journal publications.

His research interests include computer vision, image processing, pattern recognition, and machine learning.



Ahmad R. Shahid

Ahmad R. Shahid is an assistant professor in department of computer science, COMSATS Institute of Information Technology in Islamabad, Pakistan. He did his PhD in Computer Science from University of York in York, United Kingdom. His research interests include machine learning, pattern recognition, computer vision, and natural language processing.