Fatigue Monitoring for Drivers in Advanced Driver-Assistance System

Chapter · January 2020

DOI: 10.4018/978-1-7998-0066-8.ch008

2 authors:

Sarvani Videla
K L University

Ashok Kumar P M
Lakireddy Bali Reddy College of Engineering

Examining Fractal Image Processing and Analysis

Soumya Ranjan Nayak
Chitkara University, India

Jibitesh Mishra
College of Engineering and Technology, India

A volume in the Advances in Computational Intelligence and Robotics (ACIR) Book Series

IGI Global
DISSEMINATOR OF KNOWLEDGE

Published in the United States of America by
IGI Global
Engineering Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA, USA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

Library of Congress Cataloging-in-Publication Data

Names: Nayak, Soumya Ranjan, 1984- editor. | Mishra, Jibitesh, editor.
Title: Examining fractal image processing and analysis / Soumya Ranjan Nayak and Jibitesh Mishra, editors.
Description: Hershey PA : Engineering Science Reference, an imprint of IGI Global, [2020]
Identifiers: LCCN 2019015490| ISBN 9781799800668 (hardcover) | ISBN 9781799800675 (softcover) | ISBN 9781799800682 (ebook)
Subjects: LCSH: Computer vision. | Image processing--Digital techniques. | Image analysis.
Classification: LCC TA1634 .E975 2020 | DDC 006.4/2--dc23 LC record available at https://lccn.loc.gov/2019015490

This book is published in the IGI Global book series Advances in Computational Intelligence and Robotics (ACIR) (ISSN: 2327-0411; eISSN: 2327-042X)

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

All work contributed to this book is new, previously-unpublished material.
The views expressed in this book are those of the authors, but not necessarily of the publisher.

For electronic access to this publication, please contact: eresources@igi-global.com.

Advances in Computational Intelligence and Robotics (ACIR) Book Series

ISSN:2327-0411
EISSN:2327-042X

Editor-in-Chief: Ivan Giannoccaro, University of Salento, Italy

MISSION

While intelligence is traditionally a term applied to humans and human cognition, technology has progressed in such a way to allow for the development of intelligent systems able to simulate many human traits. With this new era of simulated and artificial intelligence, much research is needed in order to continue to advance the field and also to evaluate the ethical and societal concerns of the existence of artificial life and machine learning.

The **Advances in Computational Intelligence and Robotics (ACIR) Book Series** encourages scholarly discourse on all topics pertaining to evolutionary computing, artificial life, computational intelligence, machine learning, and robotics. ACIR presents the latest research being conducted on diverse topics in intelligence technologies with the goal of advancing knowledge and applications in this rapidly evolving field.

COVERAGE

- Machine Learning
- Natural Language Processing
- Robotics
- Heuristics
- Automated Reasoning
- Pattern Recognition
- Computational Logic
- Artificial Life
- Agent technologies
- Algorithmic Learning

IGI Global is currently accepting manuscripts for publication within this series. To submit a proposal for a volume in this series, please contact our Acquisition Editors at Acquisitions@igi-global.com or visit: http://www.igi-global.com/publish/.

The Advances in Computational Intelligence and Robotics (ACIR) Book Series (ISSN 2327-0411) is published by IGI Global, 701 E. Chocolate Avenue, Hershey, PA 17033-1240, USA, www.igi-global.com. This series is composed of titles available for purchase individually; each title is edited to be contextually exclusive from any other title within the series. For pricing and ordering information please visit http://www.igi-global.com/book-series/advances-computational-intelligence-robotics/73674. Postmaster: Send all address changes to above address.

Chapter 8
Fatigue Monitoring for Drivers in Advanced Driver-Assistance System

Lakshmi Sarvani Videla
Koneru Lakshmaiah Education Foundation, India

M. Ashok Kumar P
Koneru Lakshmaiah Education Foundation, India

ABSTRACT

The detection of person fatigue is one of the important tasks to detect drowsiness in the domain of image processing. Though lots of work has been carried out in this regard, there is a void of work shows the exact correctness. In this chapter, the main objective is to present an efficient approach that is a combination of both eye state detection and yawn in unconstrained environments. In the first proposed method, the face region and then eyes and mouth are detected. Histograms of Oriented Gradients (HOG) features are extracted from detected eyes. These features are fed to Support Vector Machine (SVM) classifier that classifies the eye state as closed or not closed. Distance between intensity changes in the mouth map is used to detect yawn. In second proposed method, off-the-shelf face detectors and facial landmark detectors are used to detect the features, and a novel eye and mouth metric is proposed. The eye results obtained are checked for consistency with yawn detection results in both the proposed methods. If any one of the results is indicating fatigue, the result is considered as fatigue. Second proposed method outperforms first method on two standard data sets.

DOI: 10.4018/978-1-7998-0066-8.ch008

INTRODUCTION

The symptoms of fatigue are the driver feels difficulty to be awake. He will be constantly yawning or closing his eyes. The person's fatigue impacts the alertness and response of a driver which results in increasing rate of accident. There may be several factors that would result in accidents. One of the major factors in crashes is due to drowsiness. The rates of accidents are due to the fact that the sleepy person fails to make correct decisions when necessary. The National Transport Commission conveys that at least 45 percent of heavy vehicle persons were impaired by fatigue. The survey also revealed fifty percent of all long-distance truck people had nodded off while driving more than once.

Different techniques are used in detecting the person-fatigue. These techniques are divided into three categories. The first category includes intrusive techniques, which are mostly based on monitoring biomedical signals, and therefore require physical contact with the person. The second category includes non-intrusive techniques based on visual assessment of person's bio-behavior from face images such as head movement and eye state positions. The third category includes methods based on person's performance, which monitor vehicle behaviors such as moving course, steering angle, speed, brake.

There are different ways to reduce the person fatigue and behavior at the driving time by alerting them. Different measure involves Face detection, eye state measurement, lip detection, yawn detection, head tilt detection are the major visual symptoms considered.

BACKGROUND

The steps involved in fatigue detection are face detection, feature detection and then classifying whether the person is fatigue or not. In literature, the features mostly considered are eye state and yawn. Some researchers considered head tilt detection for determination of fatigue. Xie Y. et al (2018) built a model using deep neural network for yawn detection. The network learns from yawning video clips and also images using transfer and sequential learning. They are able to distinguish between yawning, talking and laughing. Also, yawn is detected even when the face is turned 70 degrees away from camera. Huang R et al. (2018) has built fatigue detection convolutional network (FDCN) based on common convolutional neural network (CNN). projection cores were incorporated into FDCN to make the features learnt invariant to scale. Achieving an accuracy of 94.9% to detect fatigue using eye state.

F. Zhang et al (2017) used AdaBoost and Local Binary Features to detect features and CNN for classifying and achieved an accuracy of 95%. B. N. Manu et al. (2016) used binary support vector machines to detect closed eye detection and yawn. They used linear kernel and achieved an accuracy of 95%.

Automatic fatigue detection of drivers through yawning analysis is done by Azim, Tayyaba, et al. (2009). After locating face in a video frame, the mouth region is extracted, where lips are searched using spatial fuzzy c-means (s-FCM) clustering. If the yawning state of the driver is detected for several consequent frames, the driver is said to be drowsy. Fuzzy C-means uses spectral and optimal information to segment lips region. Danisman et al. (2010) proposed variations in eyes location based on the symmetry feature along horizontal direction of the eyes. If the symmetry doesn't occur, it corresponds to a closed eye state. Omidyeganeh et al. (2011) detected drowsiness by analyzing eye and mouth states. The person's face is captured from a camera and then converted to color spaces of YCbCr and HSV. To extract eyes, Structural Similarity Measure (SSIM) was used which uses properties which are statistical such as mean and variance. The SSIM values vary between -1 and 1. The maximum value is gained when two images are the same. It is used along with a template to find out the best match for the eyes.

Different techniques used to detect face and facial features are presented in Table 1. Of all the face detection techniques Modified Viola –Jones technique proposed by Videla, L.S. et al. (2018) and Histogram of Oriented Gradients coined by Navneet Dalal et. al. (2006). HOG+ Support Vector Machine (SVM) based face detector gave state of the art results. A brief description of Viola Jones and Histogram of Oriented Gradients is presented in this paper.

Viola Jones Face Detection Algorithm

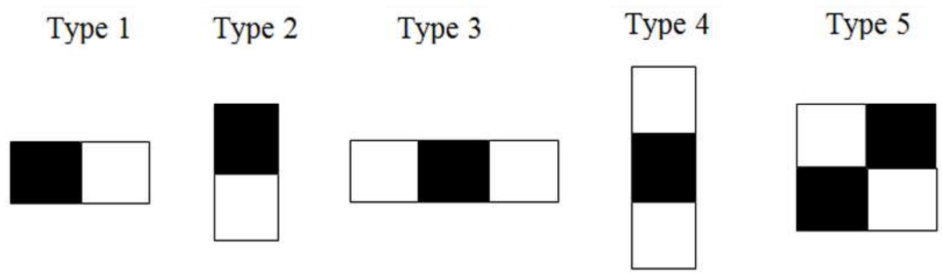
The Viola-Jones algorithm is proposed by Viola, P. et al. (2004) is used for object detection. It specifies whether an image contains a human face or not. For detecting face objects from frames, we use Haar classifiers. Haar Features impulse a real-time performance with high detection and low false positive rates. At first, the integral image is calculated and Haar classifiers are applied on each frame of the input image to find the location of objects. The integral image makes feature extraction faster. The different Haar cascades used to find the features are:

Haar classifier has two rectangle features which are computed using the difference between the sums of pixels within two rectangular regions. A three-rectangle feature calculates the sum of two outside rectangles subtracted from the sum in a center rectangle. The feature is a four-rectangle feature is determined by finding the difference between diagonal pairs of rectangles.

Table 1. Different techniques to detect features

	Technique used	Advantages	Disadvantages
Face Detection	Viola-Jones is used by Azim, Tayyaba, et al. (2009) and by Danisman, Taner, et al (2010).	Matches face exactly using similarity measures like sum of absolute differences (SAD), the sum of squared difference (SSD), and the normalized cross correlation (NCC).	Yields better result when combined with Linear Discriminant Analysis (LDA).
	Principal Component Analysis (PCA) approach is used by Karim, Tahia Fahrin, et al.(2010).		
	Detected using skin color technique by Omidyeganeh et. al.		
	Karhunen–Loève transform (KLT) used by Qian et. al. (2011).	Face recognition is done by using the Euclidean distance measure along with different tilt angles.	Percentage of face recognition decreases when database has many samples.
	Combination of Color image segmentation, Belief Propagation and PCA used by Kamencay et. al.	Belief propagation used to find marginal probabilities.	Gives better result using LDA and 2D PCA.
	SIFT - PCA and Graph Based Segmentation is used by Kamencay et. al. (2012).	(Scale-invariant feature transform) SIFT is insensitive to illumination.	
	Segmentation using thresholding is used by Rezaee, Khosro, et al (2013).	Gives better result in different varying light conditions.	It is not robust to extreme expression variation.
	Tiesheng, Pengfei, et al. (2005) detects using YCbCr, HSV color spaces along with RGB color space. Then it matches with the database by using template matching. It then uses Daubechies function.	Detects face exactly.	Lengthy process to detect the face.
Mouth and Yawn	Spatial fuzzy c-means (s-FCM) by Azim, Tayyaba, et al. (2009).	Better performance than k-means cluster.	Number of clusters should be known in advance.
	Haar Features and Linear Support Vector Machine is used by Kumar, K., et al (2012).	It finds the minimum distance between the classes.	Selection of structural elements plays very crucial.
	K-means clustering is used by Rezaee, Khosro, et al (2013).		Numbers of clusters should be known in advance.
Eye	Neural Network is used by Danisman, Taner, et al (2010).	Various conditions of eye blinks are measured by using symmetry property.	Presence of glass affects eye detection.
	Structural Similarity Measure was used by Omidyeganeh et. al.	Obtains similarity by using some statistical measures such as mean and variance.	Doesn't yield better result when the template doesn't match with eye.
	Morphological top hat and bottom hat operations are done by Kumar, K., et al (2012).	The exact eyes are identified using eye localization process.	It varies based on the structuring element taken.
	Edge detection using Sobel is used by Rezaee, Khosro, et al (2013).	Using convolution process of mask on image gives exact result of edges.	Not much performance observed when compared to canny edge detection.
Head	Eye Localization process and mouth contour are used to detect head angle by Kumar, K., et al (2012)	Triggers alarm as soon as the (Hierarchical Dirichlet Process) HDP values crosses the threshold value.	Detect angle when angle of deviation is much higher.
	Detection of head lowering using HDP is used by Rezaee, Khosro, et al (2013).		Varies based on the color values of Red.

Figure 1. Haar classifiers used in Viola Jones



Here the feature detection of input image is done by means of sub-window capable of detecting features. Then the sliding window is made traversed over the image for each scale increment. Here a cascade of classifiers is used to detect the presence of face regions. In each stage, it rapidly rejects regions which do not contain the face part. If the desired face location is not found at any stage in the cascade, the detector rejects the region and the process is terminated.

For each scale increment, the sliding window traverses over the image by producing multiple detections around the face. These multiple detections are merged into a single bounding box of the target object. The size of the final output box is the average of the sizes of the bounding box for individual detections.

The Viola-Jones algorithm steps are as below:

1. Integral Image Calculation
2. Feature Computation
3. Adaboost Feature Selection
4. Classifier Cascade

All the Haar features in different scales are used to produce approximately 1,80,000 features. Viola Jones uses 24x24 windows as the base window size to evaluate the Haar features.

Integral Image Calculation

Integral Image of Figure 2 is calculated as follows:

For each pixel, we draw a line as follows. All the pixel intensities above the line must be added to get the integral image.

$\begin{matrix} 5 & 2 \\ 3 & 6 \end{matrix}$ The value of first pixel remains the same.

Figure 2. 4x4 pixel image with pixel intensities

5	2	5	2
3	6	3	6
5	2	5	2
3	6	3	6

$\begin{array}{cc|} 5 & 2 \\ \hline 3 & 6 \end{array}$ The value of first row second column value changes from 2 to 7

$\begin{array}{cc|} 5 & 2 \\ 3 & 6 \\ \hline \end{array}$ So, in place of 6 we get 6+5+2+3 = 16

We calculate like this for all the pixels in the image and when a Haar classifier is run on the image. We sum all the pixels under black region and subtract from sum of all pixels under white region. This computation will be easy, if we calculate the integral image.

Feature Computation

The feature computation is done by overlapping the Haar Classifier over the integral image and is shown in Figure 3.

Sum of pixels under black region = 5+32-(7+16) =14 (same as 6+2+6 =14 in given image)

Sum of pixels under white region = 7+48-(12+32) =11 (same as 3+5+3 in given image)

Figure 3. Feature computation to detect objects

5	7	12	14
8	16	24	62
13	23	36	46
16	32	48	64

AdaBoost Classifier

Adaboost is used to train Strong Classifier which is linear combination of weak classifier. It also decides whether a feature is relevant or not. The steps in Adaboost are:

1. Training set of positive and negative examples (Ex: faces and non-faces images).
2. Initially all the positive training images are given weights equal to 1/(2*number of positive examples) and negative training images are given weights equal to 1/(2*number of negative examples).
3. All the 1,80,000 Haar features or weak classifiers are run on the training images
4. A good threshold (for ex: decision tree) such that any image above threshold is face and below threshold is non-face is determined.
5. Now, Error rate is calculated as sum of weights of images misclassified by each weak classifier. Of the 1,80,000 error rates choose the weak classifier with lowest error rate.

The chosen weak classifier is added to the strong classifier. Now, increase the weights of misclassified images and decrease the weights of correctly classified by normalizing the weights. Repeat the steps 3 to 5 for 1,80,000 times and all the Haar features are run on the images with updated weights and each round selects one weak classifier, which is added as linear combination to obtain final Strong Classifier. The output of weak classifier is 1 or 0 for classifying the image as face or non-face.

Cascading of Stages

After all the rounds of Adaboost, we build a strong classifier which is a linear combination of selected weak classifiers (let's say, 2,000). Instead of running all the 2,000 weak classifiers on the 24x24 window of test image, we build a cascade of classifiers.

To train a cascade, we must choose

- Number of stages or Strong classifiers in cascade
- Number of weak classifiers in strong Classifier (which is done by Adaboost)

A heuristic algorithm, Manual Tweaking is used to train the cascade as follows

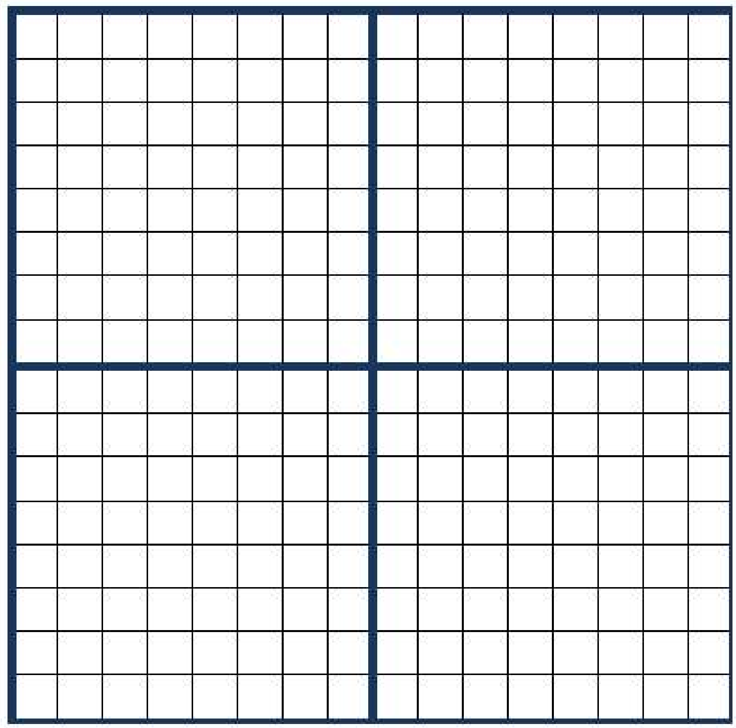
1. Select Maximum Acceptable False Positive rate.
2. Select Minimum Acceptable True Positive rate.
3. Threshold for each Strong Classifier (which decided by Adaboost)

```
Let the User select the Target Overall False Positive for all
the stages
Until Target Overall False Positive is met
    Add new Stage
   Until Maximum Acceptable False Positive rate and Minimum
Acceptable
   True Positive rate are met
    Keep adding weak classifiers and train Strong Classifier
using Adaboost
```

Histogram of Oriented Gradients

In Histogram of Oriented Gradients proposed by Navneet Dalal et. al. (2006), first Gaussian Smoothing Mask is applied on the image. The given image is divided into 16x16 blocks each of 2x2 cell and each cell is 8x8. If the image is of size 64x128, we get 105 blocks. Each block is as shown in Figure 4

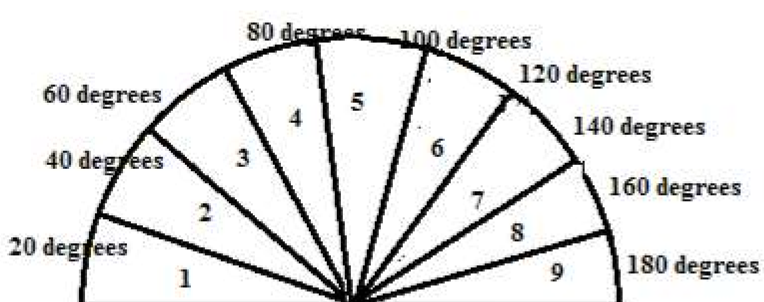
Figure 4. Blocks in histogram of oriented gradients



Gradient magnitude and gradient direction is computed for each block. By looking at Gradient direction, gradient orientation is quantized into any of 9 bins (0- 180 degrees) as shown in Figure 5.

If direction is not in one of the bins, interpolation is used. All descriptors i.e., 105 blocks each of 9 dimensions as we have 9 bins are concatenated. Hence a total of

Figure 5. Quantization of gradient orientation into bins



3780 descriptors are used to plot a histogram. X-axis of histogram is the bin values say (20 degrees, 40 degrees...as shown in the above figure 4). Y-axis is the count. One counts how strong the gradient direction is using gradient magnitude(vote)

Proposed Method

In proposed method1, the eyes and mouth are detected in test image using Viola – Jones Eye and mouth detection algorithm. Then HOG features are computed for both eyes. These features are fed to SVM classifier that classifies the test image as eye state closed or not. Mouth map area is computed for the detected mouth by separating the chromium parts of blue and red. The distance between upper lip and lower lip is calculated for normal mouth using mouth map. If the distance between upper lip and lower lip of mouth is greater than the distance of normal mouth, then yawning is detected. If either the eye state and mouth determines fatigue, it is considered as fatigue. But this method is not fully automatic and requires calculating of normal mouth distance.

Modified Feature Extraction proposed by Videla, L.S. et al. (2018) is used for Eye Detection. HOG features are extracted from the detected eye bounding box as follows

1. Convert the image into gray scale
2. Divide the cells and group of cells are called blocks
3. Calculate the gradients in x and y direction

$$\left|G\right| = \sqrt{G_x^{\ 2} + G_y^{\ 2}}$$

where

D_x=[-1 0 1] and D_y=[-1 0 1]^T

G_x=I*D_x

G_y=I*D_y

where * denotes convolution operator.

4. 4. Orientation is computed by

$$\theta = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

5. The angle transformed into degrees is which gives values in range α=θ*180/Π which gives values in range of (-180, 180] in degrees
6. 6. For signed gradient, we need to change the range of gradient (-180, 180] to [0,360) degrees. It is done using the formula

$$a_{signed} = \begin{cases} a \;\; a \geq 0 \\ a + 360 \, a < 0 \end{cases}$$

7. Group the cells into large spatially connected blocks. Normalize over overlapping spatial blocks. Hog is then vector of components of normalized cell histograms.

Let v be the non-normalized vector which has all histograms in a given block and $\| v_k \|$ be k-norm. Then the normalization factor used by Song, Fengyi, et. al. (2014) is

L1-norm is $\frac{v}{\|v\|_1 + e}$

L2-norm is $\frac{v}{\sqrt{\|v\|_2^2 + e^2}}$

8. This vector is fed to the SVM classifier to detect the closed eye state.
9. End

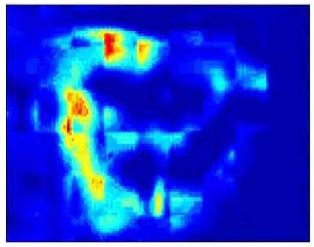
Initial distance is calculated dynamically by considering the distance between intensity changes in the mouth map of normal mouth. If the calculated distance is greater than the initial distance, then it is considered as yawning.

The Figure 6 shows the mouth map area detected during yawn by separating the chromium parts of blue and red

Proposed method 2

In this method Dlib's HOG+SVM based face detector proposed by D. E. King et. al (2009) which is based on HOG is used. HOG descriptors are calculated for the given image and SVM classifier is used to detect the faces. Facial landmarks are detected for each face detected. Dlib facial landmark detector is used to detect the

Figure 6. Mouth map area for yawning



landmarks by Christos Sagonas et. al. (2013) and Vahid Kazemi et. al.(2014) . An ensemble of regression trees is used in Dlib facial landmark detector. Gradient boosting is used to detect the pixel intensities. These intensities are used to detect the landmarks. The landmarks determined by Dlib for eyebrow, eye and mouth are same as proposed by Ralph Gross et. al (2010) and are as shown in Figure 6 and 8. Eye aspect ratio is used by Cech, J. et. al (2016) to detect the eye state. In this paper eye state is determined as Eye distances are depicted in Figure 7.

EyeDistance1=E1~E2 where
E1 = Euclidean distance between landmark 19 and 38;
E2 = Euclidean distance between landmark 19 and 42
EyeDistance2=E3~E4 where
E3 = Euclidean distance between landmark 20 and 38;
E4 = Euclidean distance between landmark 20 and 42
EyeDistance3=E5 ~ E6 where
E5 = Euclidean distance between landmark 21 and 38;
E6 = Euclidean distance between landmark 21 and 41

If eye distance is less than or equal to 3 then the eye is closed else eye is open. The eye distances are calculated for both the eyes. The Eye distances for 1001 images provided by Song, Fengyi, et al (2014) are as shown in Figure 8 and the threshold considered is 3.

Figure 7. Eye distances

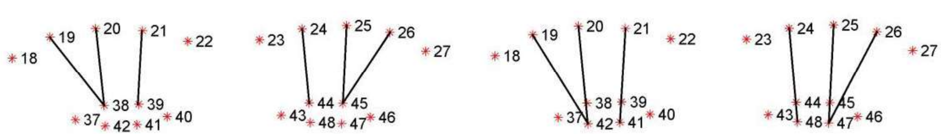
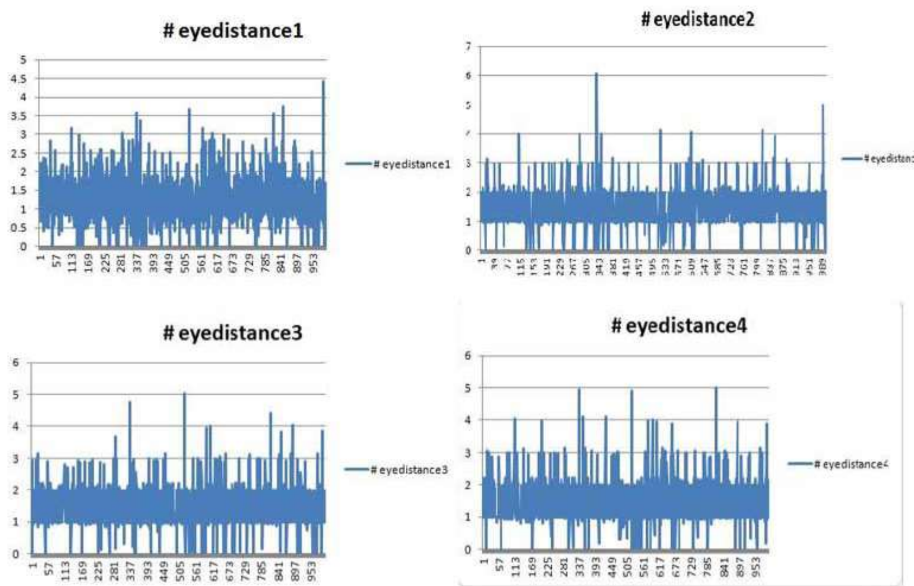


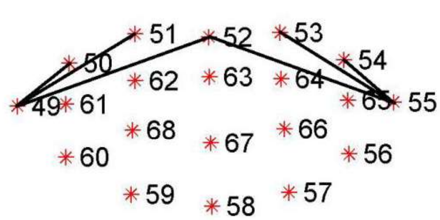
Figure 8. Eye distances plotted for 1001 images taken from dataset by Song, Fengyi, et al (2014)



New method has been proposed to detect yawning. For simultaneous yawning detection, the mouth distance is calculated from Figure 9 as follows

Mouth Distance1= Euclidean distance between landmark 49 and 52
Mouth Distance2= Euclidean distance between landmark 55 and 52
Mouth Distance3= Euclidean distance between landmark 49 and 51
Mouth Distance4= Euclidean distance between landmark 55 and 51
Mouth Distance5= Euclidean distance between landmark 49 and 50
Mouth Distance6= Euclidean distance between landmark 55 and 50

Figure 9. Euclidean distances considered for yawn detection



After analysis it is found that if the mouth distance1 or mouth distance2 >16 then a yawn is detected. Also, if mouth distance 3 or mouth distance4 > 13 then a yawn is detected and mouth distance5 or mouth distance6 >7 then a yawn is detected. When the eyes are occluded by glasses, the mouth distance helps to evaluate the drowsiness of the person. If either or both closed eye state and yawn are detected, the person is determined to be fatigue

Yawn images are extracted from the videos in YawnDD dataset provided by Abtahi, S. et. al. (2014). The images on which face is detected are only considered. These images are used as test images to analyze the accuracy of the performance of proposed methods. All the images are resized to 60x60 pixels. Results of proposed method2 are shown in Figure 10 and the performance of proposed methods for detecting yawn are shown in Table 2.

Closed Eyes in the Wild dataset provided by Song, Fengyi, et. al. (2014) consists of 2423 subjects, among which 1192 subjects with both eyes open are downloaded

Figure 10. Results of proposed method 2 on YawnDD dataset provided by Abtahi, S. et. al. (2014)

Table 2. Performance of proposed methods for detecting yawn

Camera inserted	Number of images considered	Proposed Method1 Yawn detection Accuracy	Proposed Method2 Yawn detection accuracy
In the front mirror of car	80	55%	96.55%
On Dash of Car	29	79.31%	96.55%

from internet and 1231 subjects with open eyes are provided by G. B. Huang et. al. (2007). Results of proposed method2 are shown in Figure 11.

Performance of proposed methods for detecting closed eye state in CEW dataset are shown in Table 3.

CONCLUSION

A novel metric has been proposed to detect the closed eye state and yawn detection. The state of the art face detection and landmark detection are used to measure the accuracy of the proposed method. The proposed method gives high accuracy on two standard datasets. The work can be can be extended to real time videos.

Figure 11. Results of proposed method 2 on CEW dataset

Table 3. Performance of proposed methods on CEW dataset

Number of images considered	Proposed Method1 Eye State detection Accuracy	Proposed Method2 Eye State detection accuracy
2000	85%	96%

NOTE

In this chapter, we presented two algorithms for detecting human fatigue. Algorithm that uses off the shelf facial landmark detectors performed better. A novel mouth and eye metric is proposed to detect yawning and eye closure.

REFERENCES

Abtahi, S., Omidyeganeh, M., Shirmohammadi, S., & Hariri, B. (2014). YawDD: A Yawning Detection Dataset. *Proc. ACM Multimedia Systems*. 10.1145/2557642.2563678

Azim, T. (2009). *Automatic fatigue detection of drivers through Yawning analysis*. Signal Processing, Image Processing and Pattern Recognition. Springer Berlin Heidelberg. doi:10.1007/978-3-642-10546-3_16

Cech, J., & Soukupova, T. (2016). *Real-Time Eye Blink Detection using Facial Landmarks. 21st Comput*. Vis. Winter Work.

Dalal, N., Triggs, B., & Schmid, C. (2006). Human detection using oriented histograms of flow and appearance. In *European conference on computer vision*, (pp. 428–441). Springer. 10.1007/11744047_33

Danisman, T. (2010). Drowsy driver detection system using eye blink patterns. In *Machine and Web Intelligence (ICMWI), 2010 International Conference on*. IEEE. 10.1109/ICMWI.2010.5648121

Gross, R., Matthews, I., Cohn, J., Kanade, T., & Baker, S. (2010). Multi-pie. *Image and Vision Computing*, *28*(5), 807–813. doi:10.1016/j.imavis.2009.08.002 PMID:20490373

Huang, R. Berg, & Learned-Miller. (2007). *Labeled faces in the wild: A database for studying face recognition in unconstrained environments* (Tech. Rep. 07-49). University of Massachusetts, Amherst.

Huang, R., Wang, Y., & Guo, L. (2018, October). P-FDCN Based Eye State Analysis for Fatigue Detection. In *2018 IEEE 18th International Conference on Communication Technology (ICCT)* (pp. 1174-1178). IEEE. 10.1109/ICCT.2018.8599947

Kamencay, Jelsovka, & Zachariasova. (2011). The impact of segmentation on face recognition using the principal component analysis (PCA). *Signal Processing Algorithms, Architectures, Arrangements, and Applications Conference Proceedings (SPA)*. IEEE.

Kamencay, P. (2012). Improved face recognition method based on segmentation algorithm using SIFT-PCA. *Telecommunications and Signal Processing (TSP), 2012 35th International Conference on*. IEEE. 10.1109/TSP.2012.6256399

Karim, T. F. (2010). Face recognition using PCA-based method. In *Advanced Management Science (ICAMS), 2010 IEEE International Conference on., 3*. IEEE.

Kazemi & Josephine. (2014). One millisecond face alignment with an ensemble of regression trees. In *27th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014* (pp. 1867–1874). IEEE Computer Society.

King, D. E. (2009). Dlib-ml: A Machine Learning Toolkit. *Journal of Machine Learning Research*, *10*, 1755–1758. doi:10.1145/1577069.1755843

Kumar, K. (2012). Morphology based facial feature extraction and facial expression recognition for driver vigilance. *International Journal of Computers and Applications*, *51*(2), 17–24. doi:10.5120/8578-2317

Manu, B. N. (2017). Facial features monitoring for real time drowsiness detection. *Proc. 2016 12th Int. Conf. Innov. Inf. Technol. IIT 2016*, 78–81.

Omidyeganeh, M., Javadtalab, A., & Shirmohammadi, S. (2011). Intelligent driver drowsiness detection through fusion of yawning and eye closure. In *Virtual Environments Human-Computer Interfaces and Measurement Systems (VECIMS), 2011 IEEE International Conference on*. IEEE. 10.1109/VECIMS.2011.6053857

Qian, J. (2011). Face detection and recognition method based on skin color and depth information. In *Consumer Electronics, Communications and Networks (CECNet), 2011 International Conference on*. IEEE. 10.1109/CECNET.2011.5768500

Rezaee, K. (2013). Real-time intelligent alarm system of driver fatigue based on video sequences. In *Robotics and Mechatronics (ICRoM), 2013 First RSI/ISM International Conference on*. IEEE. 10.1109/ICRoM.2013.6510137

Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., & Pantic, M. (2013). 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, (pp. 397–403). IEEE.

Song. (2014). Eyes closeness detection from still images with multi-scale histograms of principal oriented gradients. *Pattern Recognition, 47*(9).

Tiesheng, P. (2005). Yawning detection for determining driver drowsiness. *VLSI Design and Video Technology, 2005. Proceedings of 2005 IEEE International Workshop*, 373 – 376.

Videla, L.S., & Ashok Kumar, M. (2018, March). Modified Feature Extraction Using Viola Jones Algorithm. *Journal of Advanced Research in Dynamical & Control Systems, 10*(3), 528-538.

Viola, P., & Jones, M. J. (2004). Robust real-time face detection. *International Journal of Computer Vision*, *57*(2), 137–154. doi:10.1023/B:VISI.0000013087.49260.fb

Xie, Y., Chen, K., & Murphey, Y. L. (2018, November). Real-time and Robust Driver Yawning Detection with Deep Neural Networks. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 532-538). IEEE.

Zhang, F., Su, J., Geng, L., & Xiao, Z. (2017). Driver fatigue detection based on eye state recognition. *Proc. - 2017 Int. Conf. Mach. Vis. Inf. Technol. C. 2017*, 105–110. 10.1109/CMVIT.2017.25

KEY TERMS AND DEFINITIONS

Gradient Boosting: Gradient boosting is a machine learning technique for regression and classification problems, which produces a strong classifier in the form of an ensemble of weak classifiers. Gradient boosting combines weak classifiers in iteratively. Gradient boosting generalizes by minimizing loss function and loss function must be differentiable. Gradient boosting involves weak classifiers, a loss function that has to be minimized and an additive model to add weak classifiers to minimize loss function.

Mouth Map: Color map is matrix of values. Color map can be of any length and depends on the number of colors that make the color space. Each row in the matrix defines one color. For example, in RGB image, each pixel is combination of intensities of red, green and blue colors. Each row of color map matrix contains 3 columns for storing the intensities of red, green and blue colors. Image with only red component can be used for image processing. Red component dominates around mouth area than blue component in humans. Mouth can be prominently identified by considering only chromium component in YCbCr color space. Mouth Map is generally image of the detected mouth with only one component used.

Support Vector Machines: A support vector machine (SVM) is a classifier that separates classifiers by outputting a hyperplane. In supervised learning where labeled training data is given, SVM generates a hyperplane such that data belonging to same class will be on one side of hyperplane.