AN ROBUST PREDICTIVE DRIVER DROWSINESS DETECTION USING OPEN CV BASED SYSTEM

J. VYJAYANTHI[1], G.SARATH KUMAR [2]

[1] PG Scholar, Dept Of C.S.E, Eswar College Of Engineering, Narasaraopet, Guntur Dt.

[2] Asst Professor, Dept Of C.S.E, Eswar College Of Engineering, Narasaraopet, Guntur Dt.

ABSTRACT: It is just a short time until self-driving vehicles become omnipresent; be that as it may, human driving management will stay a need for quite a long time. Driver Drowsiness is one of the significant reasons of roadways accidents these days. Hence fatigue and drowsiness detection play a major role in preventing the road accidents. Every year, because of this there is an increase in the number of deaths and injuries globally. Recently, in this decade, many images processing-based approaches were created and used to detect driver's drowsiness status. To minimize the number of accidents, a method is proposed in this paper. The algorithm focuses on the eye closure and yawning ratios. The driver is alarmed, if he/she is feeling sleepy.

1. INTRODUCTION

National Highway Safety Administration reports that almost 36,061 fatal vehicle crashes happen each year in United States. Alcohol intoxication, distraction and drowsiness have a considerable proportion in these crashes (by respectively 31%, 29%, and 2.5%) [2, 3]. However, lack of physical traceability of distraction and drowsiness cause their roles to be underestimated. Each year more than 80,000 crashes occur in Nation's highways because of drowsy driving and result at least 850 fatalities and many injuries. Using data from the 100-car naturalistic study [4], researchers have found that drowsy driving is the reason of 22% to 24% of crashes and near-crashes observed. National Sleep Foundation's 2012 annual Sleep in America survey shows that 20 percent of workers had driven drowsy at least once per month in the past year [5]. Approximately 40 percent of train operators said they have driven drowsy at least once in the past month. More than 25 percent of those who had driven These facts and their consequences are reasons for concern. Saving lives lost due to drowsy driving needs that research in this field is continued. Prior research in alcohol impairment detection shows that the machine learning algorithms can help to prevent and reduce these crashes and their consequences. The difference between an impaired driver from an unimpaired driver can be recognized reliably by these algorithms with use of driving performance metrics based on signature pattern of lane position and steering. In this work, given a set of driving runs by drowsy and non-drowsy drivers we try to detect the drowsy drivers. drowsy had fallen asleep. Studies show that sleepiness can impair driving performance as much as or more than alcohol Drowsiness of the drivers is one of the key issues for majority of road accidents. Drowsiness threatens the road safety and causes severe injuries sometimes, resulting in fatality of the victim and economical losses. Drowsiness implies

feeling lethargic, lack of concentration, tired eyes of the drivers while driving vehicles. Most of the accidents happen in India due to the lack of concentration of the driver. Performance of the driver gradually deteriorates owing to drowsiness. To avoid this anomaly, we developed a system that is able to detect the drowsiness nature of the driver and alert him immediately. This system captures images as a video stream through a camera, detects the face and localizes the eyes. The eyes are then analyzed for drowsiness. Based on the result, the driver is alerted for drowsiness through an alarm system.

Thus, the main novelty of this work is the use of a non-intrusive system that is capable of detecting fatigue from sequences of images, which at the moment is an open problem. In most of the available works, the experimental methodology consists of extracting and classifying individual frames from each video and verifying whether the classification is correct or not, but that approach does not consider the intrinsic relationship between consecutive images, and their measures of false positives are less reliable. Currently, there are few works that test the systems on complete videos and count the number of alarms emitted during each video (which is necessary when evaluating the number of false alarms raised during a period of time). Therefore, the proposals presented in this paper can be considered a starting point for the design of such systems.

In this context, we propose two different solutions to approach the fatigue detection problem:

- The first one is focused on using deep learning to analyze a sequence of images of the driver.
- The second one uses a combination of AI and deep learning techniques to extract the important features from the image and, after that, the obtained data are introduced on a fuzzy inference system that evaluates whether the driver is drowsy or not.

The implementation of the first proposed solution consists of a combination of a CNN and a recurrent neural network (RNN), which is a type of neural network that is specialized in feature extraction from sequences of data

2. LITERATURE REVIEW

From “Real-time monitoring of driver drowsiness on mobile platforms using 3D neural networks”. In this paper, they have used depth-wise separable 3D convolution operations to detect drowsiness in drivers from real-time face video and also, they have identified micro sleeps and alerted the drivers. The results obtained show that the method can decide which features are important and it does not depend on the developer to pre-specify a set of features because they might miss some features like nose wrinkles, eyelid movement, and other facial gestures. The limitations of the paper are that the dataset which is used consists of just 18 persons and also, the frames were not labeled properly. From“The detection of drowsiness using a driver monitoring system”. In this paper, they have made use of a driver monitoring system (DMS) to detect drowsiness along with different kinds of sensors. They have also collected data in the form of signals from other vehicle-based sensors. The results obtained show that the

models were effective at dividing the drowsiness into three levels - low, moderate, and severe drowsiness. But, while differentiating between moderate and severe levels, the model was not efficient enough.

From "Driver Drowsiness Detection System Using Computer Vision". The objective of the paper is to detect driver drowsiness by analyzing human eye blinks using a recent facial landmark detection and to make use of E.A.R(eye aspect ratio) for easy, fast, and efficient blink detection. The results showed that the system was successful in driver drowsiness detection by providing a reliably precise enough estimation of the level of eye openness. This alert system can be used in real-time due to a very negligible performance cost experienced in facial landmark detection. The limitations of the paper are that a fixed blink duration is assumed even though everyone's blink duration lasts differently. EAR is estimated from two-dimensional data which cannot account for out-of-plane head orientation and the model depends only on eyes for drowsiness detection. From "Drowsiness Detection Based on Eye Closure and Yawning Detection". In this paper, driver's eye and mouth movements are tracked using Haar-cascade classifiers. This will help to detect the closing of eyes and frequency of yawning. The system also generates alarm sounds if the driver is drowsy or already asleep. As a result, in 85% of the cases, the system detects faces and facial features which are required, accurately. The system is prompt in detecting drowsiness once the feature detection of the face is positive.

3. EXISTING SYSTEM

The current drowsiness detection systems include the usage of the devices that detect the respiration rate, heart rate, blood pressure, etc. These devices can cause the driver to be uncomfortable for driving. Cannot be assured that the drivers wear these devices all the time while driving.May get lost or improper functioning which may lead to low accuracy in the result. The existing system does not produce good results in low light conditions

4. PROPOSED SYSTEM

For detecting the drowsiness of the driver. First of all the system captures images through the webcam and after capturing it detects the face through It uses haar features which can detect the face. If the system founds it as face the it will proceed for next phase i.e eye detection. The eye is also detected using haar cascade features and it is used for blink frequency. Through this algorithm we can find the percentage of time the eye lids remains closed. If it found eyes in closed state then it detects driver in drowsy state and alerts him by an alarm.

5. IMPLEMENTATION

The basic drowsiness detection system has three blocks/modules

ACQUISITION SYSTEM:

The video is recorded using webcam and the frames are extracted and processed in a laptop. After extracting the frames, image processing techniques are applied on these 2D images. Presently, synthetic driver data has been generated. The volunteers are asked to look at the webcam with intermittent eye blinking, eye closing, yawning and head bending. The video is captured for 14 seconds duration.

PROCESSING SYSTEM:

OpenCV is used for the processing of the image. System makes use of the number of eye blinks for detecting the state of drowsiness in a driver. The system makes use of OpenCV with a single camera view. The eye status is obtained through image processing algorithms.

WARNING SYSTEM:

The system immediately triggers the LED warning light module and at the same time emits a warning sound from the buzzer to warn the driver not to continue driving. This driver drowsiness detection and alert system architecture Take Image as Input from a Camera With a webcam, we will take images as input. So to access the webcam, we made an infinite loop that will capture each frame. We use the method provided by OpenCV, cv2.VideoCapture(0) to access the camera and set the capture object (cap). cap.read() will read each frame and we store the image in a frame variable.

Detect Face in the Image and Create a Region of Interest (ROI)

To detect the face in the image, we need to first convert the image into grayscale as the OpenCV algorithm for object detection takes gray images in the input. We don't need color information to detect the objects. We will be using haar cascade classifier to detect faces. It returns an array of detections with x,y coordinates, and height, the width of the boundary box of the object. Now we can iterate over the faces and draw boundary boxes for each face.

Detect the eyes from ROI and feed it to the classifier

The same procedure to detect faces is used to detect eyes. First, we set the cascade classifier for eyes in left eye and right eye respectively then detect the eyes using.

Machine learning uses data to detect various patterns in a given dataset.

It can learn from past data and improve automatically.

How does Machine Learning Work?

A Machine Learning system learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately. Machine learning tasks are classified into several broad categories. In supervised learning, the algorithm builds a mathematical model from a set of data that contains both the inputs and the desired outputs. For example, if the task were determining whether an image contained a certain object, the training data for a supervised learning algorithm would include images with and without that object (the input), and each image would have a label (the output) designating whether it contained the object. In special cases, the input may be only partially available, or restricted to special feedbackSemi-supervised learning algorithms develop mathematical models from incomplete training data, where a portion of the sample input doesn't have labels.

Classification algorithms

and regression algorithms are types of supervised learning. Classification algorithms are used when the outputs are restricted to a limited set of values. For a classification algorithm that filters emails, the input would be an incoming email, and

the output would be the name of the folder in which to file the email.

meaning they may have any value within a range. Examples of a continuous value are the temperature, length, or price of an object. In unsupervised learning, the algorithm builds a mathematical model from a set of data that contains only inputs and no desired output labels. Unsupervised learning algorithms are used to find structure in the data, like grouping or clustering of data points. Unsupervised learning can discover patterns in the data, and can group the inputs into categories, as in feature learning. Dimensionality reduction is the process of reducing the number of features, or inputs, in a set of data.

Relation to data mining

Machine learning and data mining often employ the same methods and overlap significantly, but while machine learning focuses on prediction, based on known properties learned from the training data, data mining focuses on the discovery of

(previously) unknown properties in the data (this is the analysis step of knowledge discovery in databases). Data mining uses many machine learning methods, but with different goals; on the other hand, machine learning also employs data mining methods as unsupervised learning or as a preprocessing step to improve learner accuracy.

Types of learning algorithms

The types of machine learning algorithms differ in their approach, the type of data they input and output, and the type of task or problem that they are intended to solve.

- Supervised learning
- Unsupervised learning
- Reinforcement learning

Supervised learning

Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs. The data is known as training data, and consists of a set of training examples. Each training example has one or more inputs and the desired output, also known as a supervisory signal. Supervised learning algorithms

include classification and regression. Classif ication algorithms are used when the outputs are restricted to a limited set of values, and regression algorithms are used when the outputs may have any numerical value within a range. Similarity learning is an area of supervised machine learning closely related to regression and classification, but the goal is to learn from examples using a similarity function that measures how similar or related two objects are. It has applications in ranking, recommendation systems, visual identity tracking, face verification, and speaker verification.

Supervised learning can be grouped further in two categories of algorithms:

1.Classification

2.Regression

Unsupervised learning

Unsupervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms, therefore, learn from test data that has not been labeled, classified or categorized. Instead of responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on

the presence or absence of such commonalities in each new piece of data. A central application of unsupervised learning is in the field of density estimation in statistics, though unsupervised learning encompasses other domains involving summarizing and explaining data features.

Cluster analysis is the assignment of a set of observations into subsets (called clusters) so that observations within the same cluster are similar according to one or more predestinated criteria, while observations drawn from different clusters are dissimilar. Different clustering techniques make different assumptions on the structure of the data, often defined by some similarity metric and evaluated, for example, by internal compactness, or the similarity between members of the same cluster, and separation, the difference between clusters. Other methods are based on estimated density and graph connectivity.

Unsupervised learning

Unsupervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms, therefore, learn from test data that has not been labeled, classified or categorized. Instead of responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on the presence or absence of such commonalities in each new piece of data. A central application of unsupervised learning is in the field of density estimation in statistics, though unsupervised learning encompasses other domains involving summarizing and explaining data features.

Cluster analysis is the assignment of a set of observations into subsets (called clusters) so that observations within the same cluster are similar according to one or more predestinated criteria, while observations drawn from different clusters are dissimilar. Different clustering techniques make different assumptions on the structure of the data, often defined by some similarity metric and evaluated, for example, by internal compactness, or the similarity between members of the same cluster, and separation, the difference between clusters. Other methods are based on estimated density and graph connectivity.

K-Nearest Neighbor(KNN) Algorithm for Machine Learning

K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on Supervised Learning technique.K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.

Example: Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most

similar features it will put it in either cat or dog category.

Support Vector Machine Algorithm

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyper plane.

SVM chooses the extreme points/vectors that help in creating the hyper plane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyper plane:

6.RESULT

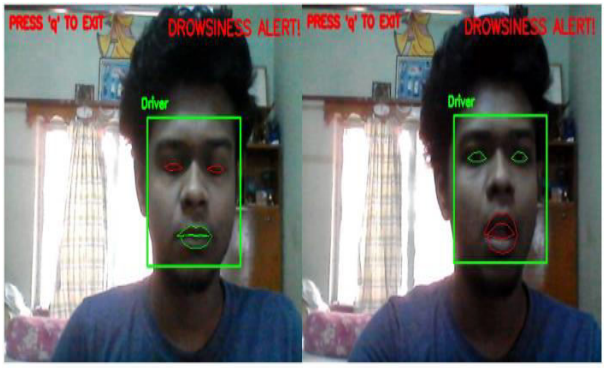


FIG: Alerting when the person is sleepy or yawning.

7.CONCLUSION

In this work, The current study developed an automated system for detecting drowsiness of the driver. The continuous video stream is read from the system and is used for detecting the drowsiness. It is detected by using haar cascade algorithm. The haar cascade algorithm uses haar features to detect face and eyes. Haar features are predefined are used for detecting different things. The haar features are applied on the image and blink frequency is calculated using perclos algorithm. If the value remains 0 for some amount of time then it detects as sleepy and alerts driver by activating an alarm. If the value remains constant for longer periods then the driver is said to be distracted then also an alarm is activated

REFERENCES

1. Rahul AtulBhope, “Computer Vision based drowsiness detection for motorized vehicles with Web Push Notifications”, IEEE 4th International Conference on Internet of Things, IEEE, Ghaziabad, India, 2019.
2. Jasper S. Wijnands, Jason Thompson, Kerry A. Nice, Gideon D. P, Aschwanden& Mark Stevenson, “Real-time monitoring of driver drowsiness on mobile platforms using 3D neural networks”, Neural Computing and Applications, 2019.
3. Chris Schwarz, John Gaspar, Thomas Miller & Reza Yousefian, “The detection of drowsiness using a driver monitoring system” , in Journal of Traffic Injury Prevention (Taylor and Francis Online), 2019.
4. AdityaRanjan, Karan Vyas, SujayGhadge, Siddharth Patel, Suvarna Sanjay Pawar, “Driver Drowsiness Detection System Using Computer Vision.”, in International Research Journal of Engineering and Technology(IRJET), 2020.
5. B.Mohana, C.M.Sheela Rani, “Drowsiness Detection Based on Eye Closure and Yawning Detection”, in International

Research Journal of Engineering and Technology(IRJET), 2019.

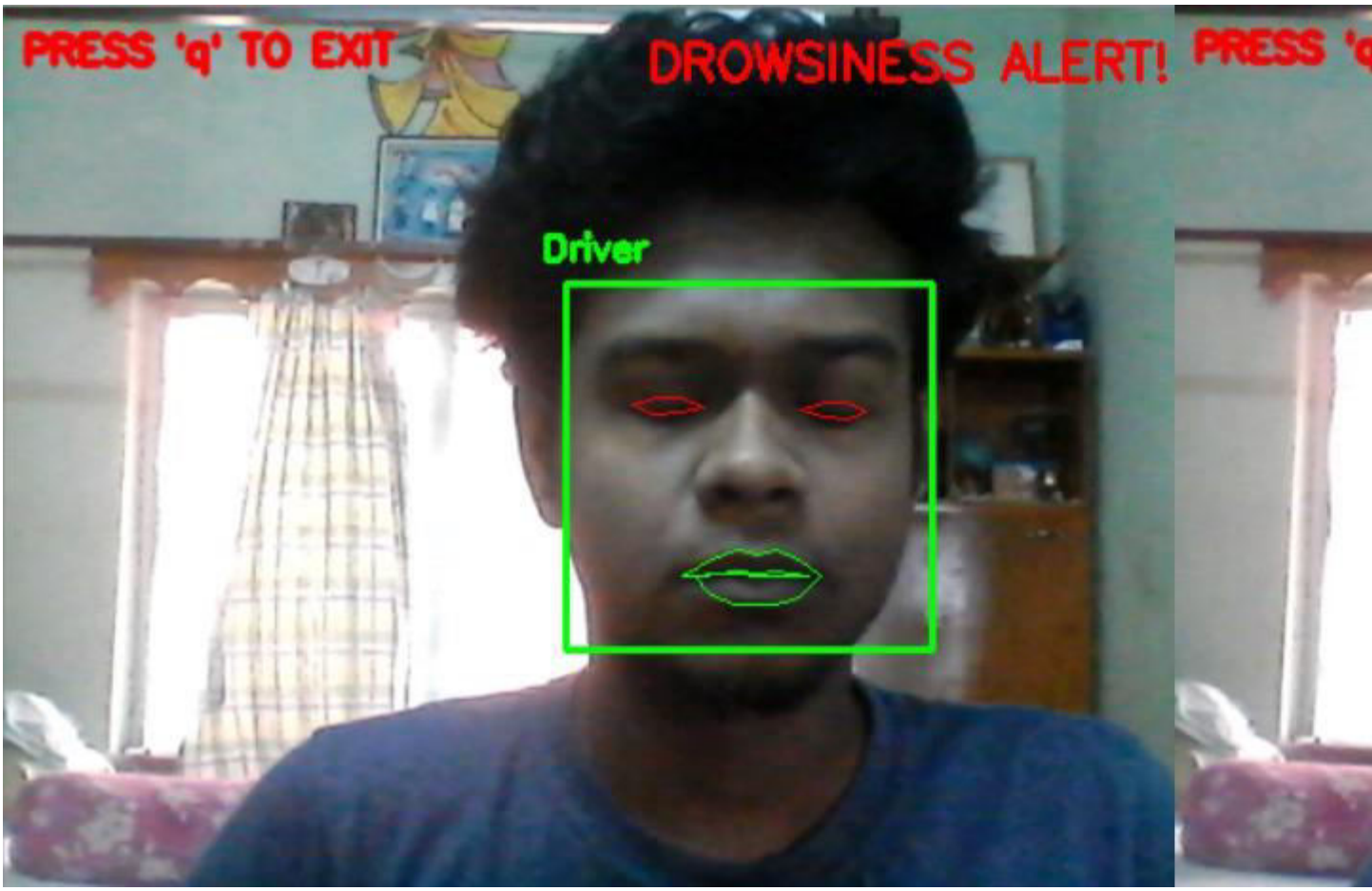
6. Driver Alert Control (DAC). (2016, Feb 10). Retrieved from http://support.volvocars.com/uk/cars/Pages/owners-manual.aspx?mc=Y555&my=2015&sw=14w20&article=2e82f6fc0 d1139c2c0a801e800329d4e

7. Z. Mardi, S. N. Ashtiani, and M. Mikaili, “EEG-based drowsiness detection for safe driving using chaotic features and statistical tests,” Journal of Medical Signals and Sensors, vol. 1, pp. 130–137, 2011.

8. T. Danisman, I.M. Bilasco, C. Djeraba and N. Ihaddadene, “Drowsy driver detection system using eye blink patterns,” Universite Lille 1 & Telecom Lille 1, Marconi, France, 2010.

9. B. Hariri, S. Abtahi, S. Shirmohammadi, and L. Martel, “A yawning measurement method to detect driver drowsiness,” Distributed and Collaborative Virtual Environments Research Laboratory, University of Ottawa, Ottawa, ON, Canada, 2011.

10. L. Li, Y. Chen and Z. Li, "Yawning detection for monitoring driver fatigue based on two cameras", Proceedings of the 12th International IEEE Conference Intelligent Transportation Systems., pp. 1-6, Oct. 2009.


PRESS 'q' TO EXIT
DROWSINESS ALERT!
Driver