Journal of Informatics
Electrical and Electronics Engineering, 2023,
Vol. 04, Iss. 01, S. No. 006, pp. 1-8
ISSN (Online): 2582-7006

Using Machine Learning to Determine the Motorist Somnolence

Sakshi Pandey[1], Sheenu Rizvi[2]

[1,2]Department of Computer Science & Engineering, Amity School of Engineering and Technology Lucknow, Amity University Uttar Pradesh, India
[1]itspandey777@gmail.com, [2]srizvi@amity.edu

How to cite this paper: S. Pandey and S. Rizvi, "Paper Title," *Journal of Informatics Electrical and Electronics Engineering* (*JIEEE*), Vol. 04, Iss. 01, S No. 006, pp. 1–8, 2023.

https://doi.org/10.54060/jieee.v4i1.88

Received: 01/04/2023
Accepted: 023/04/2023
Published: 25/04/2023

Abstract

Traffic accidents pose an increasing threat to society, and researchers are dedicated to preventing accidents and reducing fatalities, as highlighted by the World Health Organization. One significant cause of accidents is drowsy driving, which often leads to severe injuries and loss of life. The objective of this research is to create a fatigue detection system that can effectively minimize accidents associated with exhaustion. The system utilizes facial recognition technology to identify drowsy drivers by analyzing eye patterns through video processing. When the level of fatigue surpasses a predetermined threshold, the system alerts the driver and adjusts the vehicle's acceleration accordingly. The implementation of OpenCv libraries, such as Haar-cascade, along with Raspberry Pi facilitates seamless integration of the system. This dissertation evaluates advancements in computational engineering for the development of a fatigue detection system to mitigate accidents caused by drowsiness. It offers valuable insights and recommendations to enhance comprehension and optimize the system's effectiveness, ultimately leading to safer road travel.

Keywords

Fatigue Detection, OpenCV, Haar-cascade, Raspberry Pi

1. Introduction

1.1. General Introduction

Drowsy state: Sleep deprivation is reduced consciousness with tiredness and difficulty staying awake, easily triggered by stimuli. It can result from lack of sleep, medication, substance dependence, or mental disorders. Fatigue, mentally and physically draining, is the main cause. Muscular weariness is a temporary decline in muscle function. Mental weariness impairs cognitive performance. Prolonged intellectual activity leads to psychological tiredness, influenced by sleep and health. Psychological exhaustion reduces bodily effectiveness, causing drowsiness and impaired thinking. Motorist intoxication contributes to road accidents, causing fatalities, injuries, and economic losses. Nodding off at the wheel risks losing control and colliding. Monitoring motorist fatigue significantly reduces accidents.

1.2. Introduction of Work

Highway crashes in India are increasing, with 5.67 lakh collisions in 2015, averaging over one per minute. Various factors contribute, including vehicle condition, weather, negligence, motorist disengagement, lack of sleep, and physical problems. Motorist fatigue is a factor in about 30% of collisions. Indirect methods, such as sensors on the accelerator and steering wheel, measure tiredness. Steering wheel movement (SWM) and standard deviation of lane position (SDLP) assess fatigue levels. However, evaluation accuracy depends on environmental and vehicle-related factors. Behavioral indicators, measured through EEG, EOG, EMG, or ECG, can anticipate sleepiness accurately, but attaching sensors on the motorist is inconvenient and distracting [3]. Subjective assessments like self-ratings or questionnaires can yield anomalous results. Prototype-based mobility monitoring using visual motion offers precision but requires powerful computers and is noise-sensitive. Our technique uses the pupil opening ratio as a starting point, considering behavioral data without intrusive devices [4]. Raspberry Pi surveillance captures the driver's face without interruption. The Haar Cascade classifier identifies facial features, and retinal blinking alerts the motorist if deviating from the average ratio. The vehicle owner receives electronic messages if the motorist frequently nods off. Raspberry Pi controls the system, and a Pi camera tracks eye patterns [2].

1.3. Introduction of components used in The Work

The structure of the system was developed using a variety of techniques. Here are a few of them:

- **Tensor Flow** is an open-source framework for machine learning and deep neural networks. It uses data flow graphs to perform numerical computations. Operations are nodes in the graph, and data is tensors passed between nodes as edges.

Two API groups:

- **Low-Level API** provides precise control, suitable for scholars.
- **High-Level API** is simpler and enhances usability.

TensorFlow represents computations as stateful data flow graphs. Neural networks operate on tensors, multidimensional arrays[7].

- **Open CV** is open-source computer vision library for image analysis, AI, and real-time processing. Used in robotics, autonomous vehicles, and photo editing. Supports multiple platforms, languages, is BSD license, efficient, customizable architecture [6].
- **Haar-Cascade Classifier** is efficient object detection method using machine learning. Introduced by Viola and Jones and is trained with positive and negative images.[5][1]
- **Raspberry Pi** is Affordable, compact computer for learning and programming and can also perform various tasks like gaming and data sensing. It is popular for education and DIY projects [1].
- **Deep Learning**. Artificial neural networks (ANNs) form the framework for deep learning. ANNs process information through interconnected layers of nodes or neurons. Deep neural networks have multiple hidden layers for complex modeling. Deep learning includes supervised, unsupervised, and reinforcement learning.
- **Supervised learning** uses labeled datasets to train networks for prediction or classification tasks. Convolutional and recurrent networks are used for tasks like sentiment analysis and image recognition.
- **Unsupervised learning** discovers patterns in unlabeled data using techniques like auto encoders and generative algo-

rithms.

- **Reinforcement learning** maximizes rewards in an environment using techniques like Deep Q networks and Deep Deterministic Policy Gradient (DDPG).

The rest of the paper is organized as follows. Section 2 gives an algorithm for motorist fatigue monitoring. Section 3 describes the design, and software techniques. The setup used to work performance of this model are presented in section 4. In section 5, the innovation criteria for the work are described. Section 6 contains the limitations of the work. Section 7 gives the future plan for the current work. Section 8 includes the result of the work. Section 9 concludes the research findings, contribution of the work.

2. Algorithm for the Motorist Fatigue Monitoring Device

In the following Python endeavor, we will be making use of OpenCV to capture web camera photos and input them to create a Deep Learning algorithm that will figure out regardless of whether the gaze of the individual are "Open" to begin or "Closed." In order to complete the Python Work, the developers will use the following strategy:

Step 1: -Using a device's input from the user, consider a photograph.
Step 2: -Make something called the Region of Interest (ROI) around the part of the face you can see in the picture.
Step 3: -To provide input to the classification algorithm, find the eyeballs in the ROI.
Step 4: -It will classify if either of the eyes are partially or fully opened.
Step 5: -Determine the rating in order to determine if the subject is sleepy.[3]

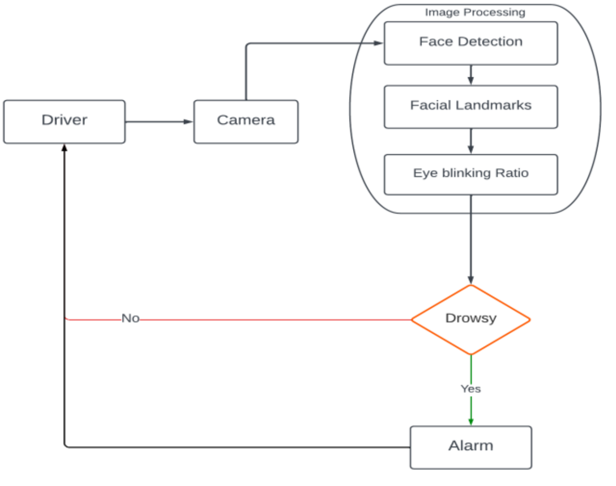

Figure 1. Flowchart of the System.

3. System Design

The concept for the system includes the three distinct stages listed below:

a. **Capturing:** The motorist's face is photographed using facial surveillance footage that is positioned on the instrument panel. The use of infrared imaging devices allows them to operate effectively well at the end of the day
b. **Detection:** The expanded/closed status of the human eye is determined by analyzing the collected picture. HARR classifier cascades incorporated within OpenCV are used to infer what the motorist's contemporary behavior during driving pattern. Through the use of an effective recognition of faces method, faces and characteristics of the face, particularly those surrounding the mouth and the eyes, are extracted from videos.

c. **Correction:** This stage is in charge of carrying out the necessary remedial measures for the specific problematic behavior that was discovered. The remedial measures include installing car displays and alarms. The essential remedial steps are carried out by the Raspberry Pi feel the need to point-board a laptop or desktop which at first is linked sequentially to the desktop.

4. Working of the Work

The notification system needs to be installed facing the driver's face. During startup, the computer system detects the driver's head and utilizes facial recognition to identify specific features. In this case, we focus on the driver's eyelids for recognition. When the motorist blinks, no alarm is activated. However, if the driver's eyes remain closed for more than three seconds, the safety system will notify them with a loud alert sound. This is part of the discussion on the various monitoring stages.

Object detection: Finding and recognizing specific objects in images. Various techniques exist, including color recognition, but it's not effective due to multiple objects of the same color and size. Haar-like features proposed by Viola and Jones offer a better method. These features resemble rectangular prism-shaped patches with light and dark traits, similar to facial features. Cascade classifier consists of multiple stages, each with weak characteristics. The program scans the image, assigning positive or negative results at each stage. Positive results indicate object detection; negative results indicate absence of the object.

Facial Recognition: Viola-Jones' face recognition technique uses Haar feature-based cascade classifier. Machine learning is employed to train a cascade function with positive and negative images. Haar characteristics resemble convolutional neural networks kernels and are assigned a number based on pixel differences. The face area is detected using cascaded Adaboost classifier with Haar-like features. The classifier combines weak classifiers and determines face presence.

Detecting eyes: Facial landmark prediction is implemented for eye identification and other tasks. Facial landmarks, such as mouth, jawline, nose, and eye features, are localized and depicted using morphology prediction techniques. Two steps are involved in identifying facial landmarks: detecting the face in the photo and identifying important facial components within the face's region of interest (ROI). Haar feature-based cascade classifiers are used for face localization. The main facial features, including the mouth, eyebrows, eyes, and nose, are recognized within the ROI. The dlib module utilizes a version based on Kazemi and Sullivan's work for facial landmark detection. Pre-trained detectors estimate 68 (x, y)-coordinates corresponding to facial structures. These coordinates can be used to locate and analyze eye regions. Different datasets can be used for training facial landmark detectors, and the dlib approach can be applied regardless of the dataset used.

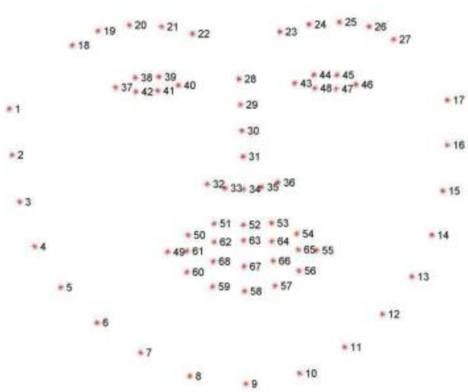

Figure 2. 68 Points representing Facial Landmarks.

Recognizing the State of the Eyes

Eye region detection can be achieved using visual flow, patch-based monitoring, or a combination of adaptive thresh-olding and frame-to-frame intensity divergence. The state of eye opening can be inferred from a single image using var-ious ap-proaches, such as correlation coefficients with open and closed eye templates, intensity analysis over the eye ar-ea, para-metric model fitting for eyelid identification, or active shape models. Previous methods often have strict re-quirements on factors like face-camera position, resolution, lighting, and motion dynamics. Our updated facial land-mark-based algorithm offers a simple yet effective solution for eye blink detection. Landmarks are used to generate a scalar variable representing the degree of eye opening. An SVM classifier trained on blinking and non-blinking sequences is employed to detect eye blinks using a series of lid-opening estimations for each frame.

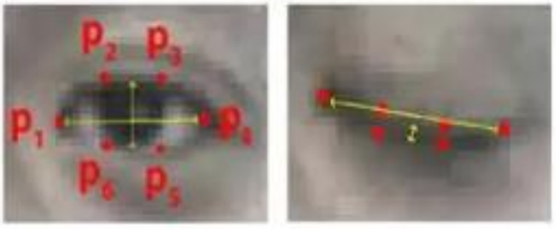

Figure 3. Example of blinking of eyes with landmark points.

Calculating the eye anticipated percent

The eye landmarks have been determined for every frame of the footage. Calculated is an eye aspect ratio (EAR), which compares the shape of the eye's elevation and breadth.

$$\mathrm{EAR} = \frac{||p2 - p6|| + ||p3 - p5||}{2||p1 - p4||} \tag{1}$$

- where the 2D landmark positions shown in figure are p1,..., and p6. The EAR is generally stable. When one eye is closed and the other is open, as the two distances approach zero. Insensitive head position and half person. The perspective proportion of the extended eye varies just slightly between people and is completely unaffected by consistent picture resizing and directly facial movement. Given that the two eyes blink simultaneously, the EAR from each eye is averaged.

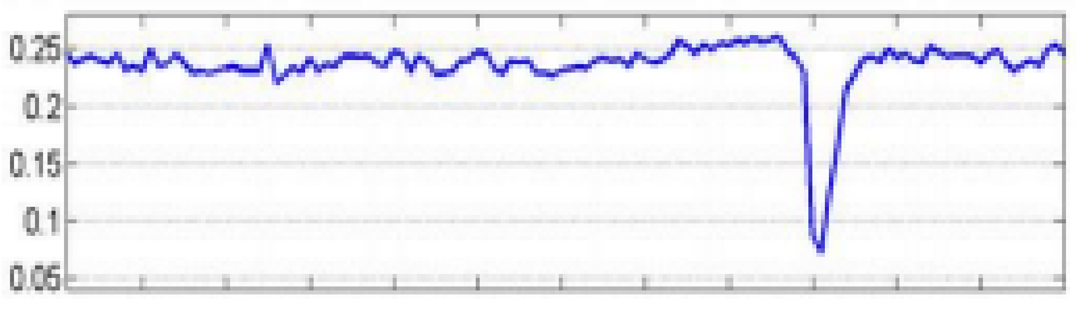

Figure 4. EAR for each eye opening and closing

- Effectively detecting landmarks with both open as well as closed eyes. The graph for a number of frames in the footage is the eye aspect ratio (EAR) from Equations. Operator adaptability setup Features: The end user may change the EAR's percentage setting from between low and high using this user interface. Additionally, it enables setting a timer's sensitivity, as seen in figure.

5. Innovation in Work

This alarm system's ineffective operation can be avoided by connecting it to the tachometer of the vehicle. For instance, there is no need for the warning system operating if the motorist is seated in the seat in charge and under the alarm system's monitoring when the car is turned on but not moving. This is because there is little chance that an accident will occur because the motorist is drowsy in a still automobile.

The second novel approach may involve integrating this alarm system. in the backwards, using the gear. As an illustration, if the motorist puts the car in the opposite direction, he or they become fewer likely to be able to make out to the forward unless they are exclusively utilizing the rearview mirror to do so. Rare reflections also have vision that is slightly elevated towards the top of the windscreen, which can lead to mistakes in the alert system's ability to correctly identify the motorist's eyes. As a result, the security system may occasionally sound, but it is useless given that the driver is paying attention to the road's backside rather than being feeling tired.

6. Limitations of The system

Every wonderful item has benefits, but it also has drawbacks. We have encountered some limitations with the eye detection system, but they can be overcome by enhancing the program's implementation, as mentioned in the "I. Potential Range" section. Let's discuss these restrictions:

6.1. Eyewear usage

Identifying the shape of an eye becomes slightly challenging when the driver wears sunglasses or shades. A high-quality cam-era in the alarm device can improve eye detection accuracy in such cases.

6.2. Multiple faces

If the camera captures multiple faces, it becomes difficult for the system to identify the driver's eye accurately, leading to unreliable data. Placing the camera close to the driver's face helps resolve this issue.

6.3. Additional distractions

Activities like eating, talking on the phone, or using headphones while driving can occasionally cause distractions. The system may fail to identify these distractions. However, we propose using artificial intelligence and machine learning to address this problem, as discussed in the following section.

6.4. Impact of surrounding lighting

Environmental lighting conditions significantly affect the eye's physical representation. Changes in lighting can cause errors in the algorithm's output. Using a camera that utilizes infrared radiation or LED light can help overcome this issue.

6.5. Distance between webcam and eyes

We assumed a close proximity (around 100 cm) between the recording device and the driver's facial features for optimal results. Different car models may vary in accommodating this distance, impacting the intended design and production.

6.6. Processing power limitations

We utilized the Raspberry Pi computer with a 700 MHz CPU, which may not be sufficiently fast for handling videos. An operating system with high speed is required, potentially increasing the product's cost.

7. Plans for the Future

We expanded our understanding of warning systems using academic articles. It's important to avoid distractions like head-phones, eating, and mobile phone use while driving as they can be fatal. We can improve the warning mechanism by using object identification and machine learning.

The system should not give a useless alarm if the driver looks out the window while reversing. Our algorithm detects drowsiness and issues an audible alert. However, quick reaction may not always prevent accidents. To address this, we can install a motor-driven system that slows down the vehicle when the alert is received. We also offer an Android app to track sleepiness levels during travel.

The algorithm tracks closed eyelids and may not provide a warning at a specific moment. Observing facial expressions or using 3D photos could improve alert triggering. Our vehicles can park autonomously and activate the braking system, parking lights, and detect available parking spots without human involvement. A pressure detector on the steering wheel can warn against nodding off. Wireless technology like Car Talk2000 can transmit information to other vehicles if the driver has a cardiac arrest or is intoxicated.

8. Result

To monitor sleepiness, we used Python and OpenCV to capture real-time footage. The recorded footage was analyzed frame by frame. The algorithm detects the face and then locates the eyes. If the driver closes their eyes for several consecutive frames, they are considered sleepy; otherwise, they are considered to be blinking normally. This process is repeated continuously. However, the implementation currently lacks the ability to mark the eye with upward or downward lines when the user is asleep or if the eye is not detected, in which case a notification is displayed.

The image below shows the eye detection and indicates the proportion of the eye when it is open. The image represents the condition of an eye when the driver is considered drowsy. Regular eye blinking does not affect the algorithm's outcome. However, if the raised eyelids are detected for a longer period than the required threshold, a DROWSINESS ALERT! is triggered. This sleepiness indication is represented by an alerting sound in the alarm system to awaken the driver's awareness.

9. Conclusion

This research introduces a method for automatically detecting driver inattention and tiredness from footage. Previous techniques focused on hypothesized actions indicating fatigue. However, this study autonomously monitored facial expressions to investigate impulsive behavior during periods of tiredness. It is the first study to identify meaningful relationships between facial expressions and weariness beyond eye blinks. The experiment also found connections between neck movement, motorist tiredness, and skull rolling during drowsiness. Interestingly, yawning, often considered a sign of fatigue, was a poor predictor of imminent collapse within the 60-second period. Motorists actually yawned less frequently as they approached slumber. This highlights the importance of using realistic instances of weariness and sleepiness when studying nodding off.

Daily, the number of traffic accidents is rapidly increasing. Fatigue-induced inattention is a major contributing factor to vehicle crashes. The present research introduces an efficient method for detecting weariness using data processing techniques to continuously track eye and facial regions. If the person is identified as sleepy, an alert is sent via Bluetooth to microcontrollers, which then use pulse width modulation to reduce motor torque. The initial version has proven to be an effective and cost-efficient approach to preventing tiredness-related accidents.

References

[1]. P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, 2005.

[2]. "Image Processing Based Anti-Sleep Alarm System for Drowsy Drivers," SDIWC Organization, Academia.edu [Online]. Available: https://www.academia.edu/37139204/Image_Processing_Based_Anti_Sleep_Alarm_System_for_Drowsy_Drivers [Accessed: 17-Jun-2023].

[3]. M. Srivastava, S. A. Idrisi, and T. Gupta, "Driver drowsiness detection system with OpenCV & keras," in 2021 International Conference on Simulation, Automation & Smart Manufacturing (SASM), 2021.

[4]. A. Shukla, N. Singh, R. Kumar, and Anand, "Drowsy detection system for bus and car drivers," SSRN Electron. J., 2022.

[5]. B. G. Amira, M. M. Zoulikha, and P. Hector, "Driver drowsiness detection and tracking based on Yolo with Haar cascades and ERNN," Int. J. Saf. Secur. Eng., vol. 11, no. 1, pp. 35–42, 2021.

[6]. J. Cicolani, "An Introduction to OpenCV," in Beginning Robotics with Raspberry Pi and Arduino, Berkeley, CA: Apress, 2018, pp. 297–341.

[7]. S. Ahlawat, "Introduction to TensorFlow," in Reinforcement Learning for Finance, Berkeley, CA: Apress, 2023, pp. 5–137.