

Vehicular Security: Drowsy Driver Detection System



Pranavi Pendyala, Aviva Munshi, Anoushka Mehra

Abstract: Detecting the driver's drowsiness in a consistent and confident manner is a difficult job because it necessitates careful observation of facial behaviour such as eye-closure, blinking, and yawning. It's much more difficult to deal with when they're wearing sunglasses or a scarf, as seen in the data collection for this competition. A drowsy person makes a variety of facial gestures, such as quick and repetitive blinking, shaking their heads, and yawning often. Drivers' drowsiness levels are commonly determined by assessing their abnormal behaviours using computerised, nonintrusive behavioural approaches. Using computer vision techniques to track a driver's sleepiness in a non-invasive manner. The aim of this paper is to calculate the current behaviour of the driver's eyes, which is visualised by the camera, so that we can check the driver's drowsiness. We present a drowsiness detection framework that uses Python, OpenCV, and Keras to notify the driver when he feels sleepy. We will use OpenCV to gather images from a webcam and feed them into a Deep Learning model that will classify whether the person's eyes are "Open" or "Closed" in this article.

Keyword: Abnormal Behaviours, Current Activity, Deep Learning, Facial Behaviour

I. INTRODUCTION

The risky combination of driving, sleepiness, and exhaustion is known as drowsy driving. This typically occurs when a driver has not had enough sleep, but it may also occur as a result of untreated sleep problems, drugs, alcohol use, or shift work. No one can pinpoint the precise moment when sleep overtakes their body. Although falling asleep at the wheel is obviously risky, being sleepy has an impact on your ability to drive safely even though you are not asleep. To avoid accidents caused by sleepiness, we'll use Python, OpenCV, and Keras to create a framework that will alarm the driver when he feels sleepy. Face recognition is the process of recognising and interpreting people's faces. Automatic face detection and recognition has exploded in popularity in the military and other top-secret organisations in recent years. When there are few examples and it is difficult to collect photos for the dataset, training becomes extremely difficult. There are a variety of methods for extracting features from a face today. Principal Components Analysis is one of them (PCA).

Manuscript received on June 02, 2021.

Revised Manuscript received on June 09, 2021.

Manuscript published on June 30, 2021.

* Correspondence Author

Pranavi Pendyala, Department of Computer Science, Vellore Institute of Technology, Vellore (Tamil Nadu), India. Email: pranavipendyala7@gmail.com

Aviva Munshi*, Department of Computer Science, Vellore Institute of Technology, Vellore (Tamil Nadu), India. Email: avivamunshi@gmail.com

Anoushka Mehra, Department of Computer Science, Vellore Institute of Technology, Vellore (Tamil Nadu), India. Email: anoushka.mehra27@gmail.com

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](https://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

The characteristics of various faces are detected using eigen values. Linear discriminant analysis (LDA), also known as the Fisher Face Process, is another method. Both of these techniques are examples of holistic features.

The use of multilayer convolutional neural networks to generate features responsible for decision making provides a new perspective on individual drowsiness recognition. After good eye detection and tracking using computer vision and deep learning techniques, the mentioned model alerts drowsy drivers with an alarm.

II. HISTORICAL ANALYSIS

Face features such as the mouth, nose, and eyes were extracted by the first facial recognition algorithm. These features were chosen in a region type to perform identification. The face and classifier datasets were then used to train classifiers based on fixed functionalities, resulting in a result. While this method was successful, it did not produce substantial results due to the small amount of data used. In facial recognition, we just look at the features of the face and forget the background. As a consequence, object-class identification is another name for it.

Fischer and Eschinger suggested a system in which features were compared based on scale. This approach was based on local template matching, in which features that fit certain proposed faces can be represented as a collection of geometric parameters, with these parameters being clubbed and distinguished. In 1973, Kenade developed the first fully developed face recognition system. He compared automated and manual abstraction and found that the recognition rate was 45-75 percent.

Mark Nixon's contribution was also beneficial, as it included a schematic measurement for eye spacing. The template matching design was improved with schemes like "deformable templates."

In 2001, Viola and Jones developed a frontal view face recognition device. The Haar Features-based algorithm is implemented in OpenCV. Although these characteristics are distinct, they share certain properties. The area around the eye is darker than the cheeks, and the nose bridge is lighter than the eyes, so local feature matching includes matching different regions of the face.

There are a variety of recognising tools available, including LBP, Fisherman, Laplacian, and others, but LBP is the best because it can be used in a variety of settings and is appropriate for various light intensities. We use rectangular representations made up of 2D waves in Haar detection. In the Haar Cascade Algorithm, black (for value minus one) and white (for value plus one) rectangles are commonly used as visual representations.



With the aid of various visual cues and observed facial features, several deep learning-based techniques have been developed for non-intrusive, real-time detection of driver sleep states. For extracting the eyes, Alshaqaqi et al proposed a detection method focused on edge detection and exploiting the symmetry of facial features. Using the Hough transform for circles and comparing the intersection of the Hough transform and the edge picture with a threshold, the state of the eyes is calculated as open or closed. Percentage of Eyelid Closure (PERCLOS), a scientifically associated indicator of drowsiness associated with infrared light at various wavelengths, is then used to assess the state of drowsiness. At fixed wavelengths, two photographs of the driver's face are taken. The discrepancy between these images is used to calculate the proportion of closed eyes.

The second approach, which is still in its early stages, employs a neural network to predict PERCLOS by identifying the optimal combinations of driver output variables. Malla and colleagues devised a method for detecting microsleep. To obtain the video, the device uses a remotely positioned camera with near infrared illumination. The Haar object detection algorithm is used to detect a face, with anthropometric parameters used to detect the eye area of interest. The ratio of the closed portion of the eye to the average height of the open portion is used to detect eye closure. In view of the above, strategies for detecting drowsiness have included detection of the ears, pupils, and (or) facial features.

Based on a comprehensive literature survey, a number of different methods that took varied inputs were found to detect if a driver is alert and attentive. Motion picture analysis has been used to create a tool for tracking a driver's blinking in real time. A person's drowsiness can be detected using a variety of methods. They are divided into three categories: image processing, electroencephalography, and artificial neural networks. Template-based image processing, yawning-based image processing, and eye blink measurement-based image processing are also available. Furthermore, a method for predicting consciousness loss based on the blink period has been developed. Individual people's blinking waveforms vary, as does the waveform of the same person over time.

With ten subjects, the blink extraction rate has increased from 16 percent to 95 percent, and we might be able to predict the degree of consciousness. Face expressions are used to detect sleeplessness, unlike traditional detection methods that rely solely on eye states. Detection systems pose a number of difficulties. Changes in intensity due to lighting conditions, the presence of glasses, and the presence of a beard on the person's face are all significant factors.

A new study is attempting to determine whether or not the drivers' eyes are red due to a lack of sleep. The sclera (the white outer layer of the eye) has a distinct and stable blood vessel structure that can be used to distinguish people. The iris area and the external form of the eye are the subject of this research. Another technique for detecting the iris-sclera pattern included three levels of eye recognition. Elliptical approximation and template matching techniques were used in the first stage of the face detection process. The percentage of eye closure (PERCLOS) measure was used to assess the driver's fatigued condition in the third level.

III. METHODOLOGY

A. Proposed Work

We'll use OpenCV to gather images from a webcam and feed them into a Deep Learning model that will classify if the person's eyes are open or closed in this Python project. For this Python project, we'll take the following approach:

Step 1

Image Acquisition: For demonstration purposes, a webcam will be used to capture images in an infinite loop.

Step 2

Region of Interest (ROI) Detection Eye: The face will be detected first, followed by the face, and finally our region of interest, the eyes. Since a colour image isn't required for this, the image will be converted to grayscale. To detect faces, OpenCV's Haar cascade classifier will be used.

Step 3

Observation of the eye : We'll be looking at a series of pictures to see if they're blinking quickly, concentrating their eyes to one side, or if they've closed their eyes.

Step 4

Warning the driver: If any drowsiness is detected in this step, an alarm sound will be produced to alert the driver.

Step 5

Make a call to an emergency number: If the driver hasn't woken up or is asleep, the software will terminate.

B. Model Architecture

We developed the model with Keras and Convolutional Neural Networks (CNN). A convolutional neural network is a type of deep neural network that performs exceptionally well when it comes to image classification. A CNN is made up of three layers: an input layer, an output layer, and a hidden layer of several layers. These layers are convolution using a filter that performs 2D matrix multiplication on both the layer and the filter.

The layers that make up the CNN model are as follows:

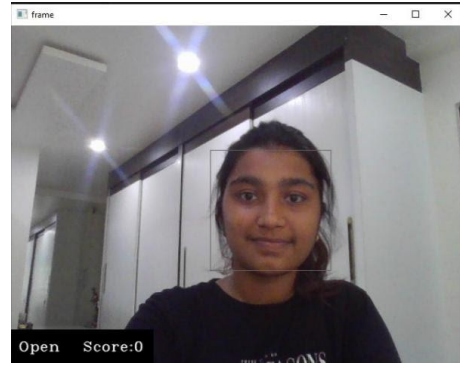
- Convolutional layer; 32 nodes, kernel size
- Convolutional layer; 32 nodes, kernel size 3
- Convolutional layer; 64 nodes, kernel size 3
- Fully connected layer; 128 nodes.

The final layer has two nodes and is also a completely linked layer. Except for the output layer, which uses Softmax, all of the layers use a Relu activation feature.

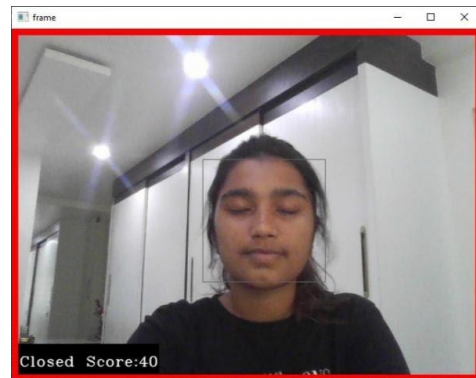


IV. RESULTS AND DISCUSSION

Initially, when eyes are open, the score is 0 with word “Open” as can be seen in the following screenshot at the bottom left:



When the eyes are closed i.e. when the driver gets drowsy we get a closed score. After 15 seconds if the eyes are still closed a buzzing noise is generated (alarm).



After 20 seconds if the alarm isn't turned off a message is sent to an emergency contact. This is done using an API from Twilio.com.

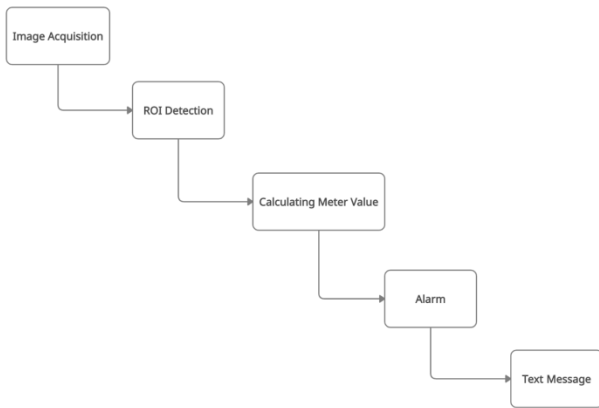


Fig. 1: Model Architecture Diagram

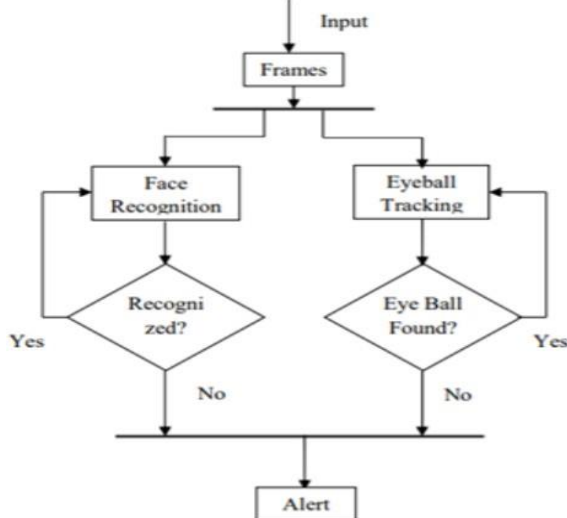


Fig. 2: Data Flow Diagram

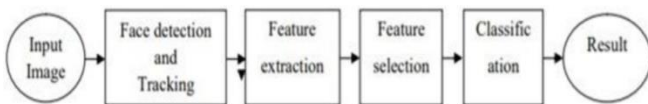


Fig. 3: System Diagram

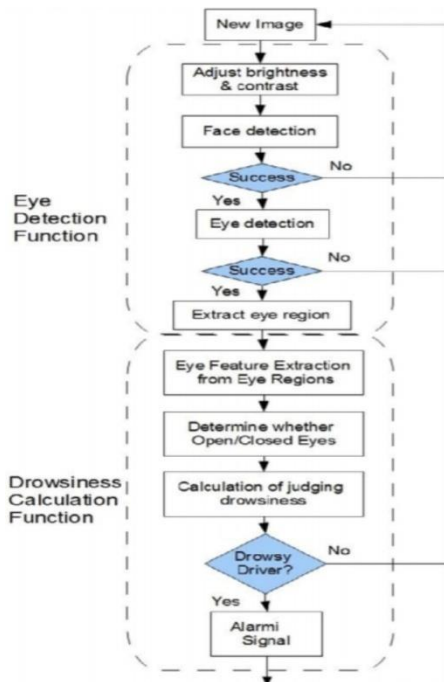


Fig. 4: System Architecture Diagram

V. CONCLUSION AND FUTURE WORK

This project proposes a representation learning-based algorithm for detecting dizziness in people. The use of multi-layer convolutional neural networks to generate features responsible for decision making provides a new perspective on individual dizziness detection. After effective eye detection and tracking with computer vision and deep learning techniques (CNN and Keras) with an accuracy of

88.5 percent, the described model alerts drowsy drivers with an alarm. To detect facial features, a model was developed using a curated dataset. The eyes were then removed from the face as the focus of our investigation. This was used to decide whether or not a driver's eyes were open. An alarm was triggered if they were closed for more than 15 seconds. If the alarm is not turned off after 20 seconds, a message is sent to the driver's emergency contact. This will help to reduce the number of road accidents caused by sleep deprivation and exhaustion.

We may also provide a function that sends the drivers location along with the message to the emergency contact. If necessary, the contacts may be able to dispatch an ambulance or other assistance.

REFERENCES

1. Ashwin S., Babu T., Muthukumaaran C., Naidu M., Ravi Raghavan C. (2019). Sleep Detection and Alert System for Automobiles. Springer, Singapore Sahayadhas, Arun & Sundaraj, Kenneth & M, Murugappan. (2012). Detecting Driver Drowsiness Based on Sensors: A Review. Sensors (Basel, Switzerland). 12. 16937-16953. 10.3390/s121216937.
2. T Danisman, IM Bilasco, C Djeraba, N Ihaddadene (2010). Drowsy driver detection system using eye blink patterns. Machine and Web Intelligence (ICMWD) IEEE, pp. 230-233.
3. K Dwivedi, K Biswaranjan, A Sethi (2014). Drowsy driver detection using representation learning. Advance Computing Conference (IACC), IEEE, pp. 995-999.
4. CH Weng, YH Lai, SH Lai (2016). Driver Drowsiness Detection via a Hierarchical Temporal Deep Belief Network. Asian Conference on Computer Vision, Springer, pp. 117-133.
5. OM Parkhi, A Vedaldi, A Zisserman (2015). Deep Face Recognition. British Machine Vision Conference (BMVC),1,p.6.
6. <https://sci-hub.si/10.1109/itsc.2002.1041208>
7. <https://www.ijedr.org/papers/IJEDR1303017.pdf>
8. [https://www.safetylit.org/citations/index.php?fuseaction=citations.viwdetails&citationIds\[\]=citjournmalarticle_245681_38](https://www.safetylit.org/citations/index.php?fuseaction=citations.viwdetails&citationIds[]=citjournmalarticle_245681_38)
9. <https://sci-hub.si/10.1109/VNIS.1994.396873>
10. <https://sci-hub.si/10.1109/ICSIPA.2011.6144162>
11. <https://journals.sagepub.com/doi/10.1243/0954407011528>

AUTHORS PROFILE



Pranavi Pendyala, UG third year Student at Vellore Institute of Technology, actively pursuing Computer Science and Engineering. Core Interests: Algorithm's development, Artificial Intelligence, Data Science and Convolution Neural Networks. Email ID: pranavipendyala7@gmail.com



Aviva Munshi, UG third year Student at Vellore Institute of Technology, actively pursuing Computer Science and Engineering. Core Interests: Data Science, Artificial Intelligence, Machine Learning. Email Id: avivamunshi@gmail.com



Anoushka Mehra, UG third year Student at Vellore Institute of Technology, actively pursuing Computer Science and Engineering. Core Interests: Machine Learning, Data Visualization, Data Analysis. Email Id: anoushka.mehra27@gmail.com

