Journal of Ubiquitous Computing and Communication Technologies (ISSN: 2582-337X)
www.irojournals.com/jucct/

Intelligent Drowsiness Detection using Haar Cascade Classifier and Convolutional Neural Network

Ashokkumar Janarthanan[1], Arjun Paramarthalingam[2], Sindhuja Sundaresan[3], Yokeshvaran S.[4]

[1, 2, 4]University College of Engineering, Villupuram, India.

[3]Surya Group of Institutions, Villupuram, India.

E-mail: [1]jana.ashokkumar@yahoo.com

Abstract

Advanced driver alert systems are enabled by computer vision technology. They warn drivers of drowsiness and fatigue in a bid to save lives from horrific accidents. Drivers should stay alert, read traffic signs, and drive defensively, as driver sleepiness is a huge threat. The drowsiness detection support systems are now a necessity in the automotive industry. By keeping drivers' eyes on the road, they assist in highway safety and reduce deaths. Their usage reduces drowsy driving by a significant percentage and provides all people with safer roads to travel on. Eye monitoring has revolutionized driver safety. Haar-Cascade-based computer vision sensors and CNN algorithm-based computer vision sensors track facial landmarks in real-time. These systems calculate eye-to-eye ratios to infer signs of sleep, providing critical information about driver status, alertness, and fatigue. The emerging technology translates raw data into usable intelligence, with road safety taking precedence over exact face recognition and neural network processing. Face recognition identifies the eyes, whether left or right, and determines eye state open or closed based on intensity values and the space between eyebrows and eyelashes. Threshold crossing establishes open eyes; below the threshold indicates closed eyes. A sequence of closed-eye frames raises an alarm. The algorithm proves to be 90% effective on varied faces. Low computational requirements and

DOI: https://doi.org/10.36548/jucct.2025.2.006
Received: 08.05.2025, received in revised form: 12.06.2025, accepted: 01.07.2025, published: 15.07.2025

real-time processing make it an ideal application for surveillance. The accuracy and efficiency of the system make it an investment that should be considered for surveillance activities.

Keywords: Drowsiness detection, HAAR Cascade Algorithm, Convolutional Neural Network (CNN), Eye tracking, Facial recognition, Driver monitoring system, Deep Learning.

1. Introduction

Driver drowsiness is also one of the most frequent causes of highway accidents. It is still on the rise every year, claiming lives worldwide. Our project revolves around decreasing the frequency of recurring cases of drowsiness-related accidents. With computer vision, our project presents a new means of detecting drowsiness. Our goal is to Develope an efficient yet affordable sleeping driver alarm detection system. The technique captures sleepiness by measuring geometric eye and mouth size [1]. Our mission is to produce an alarm detection system, unbowed by the ghastly tragedy of overlooked fatigue. Comparative research on alarm and driver drowsiness detection is continuing. Mathematically, detection of driver drowsiness also has a solution. Fatigue is determined by the new system through minimal eye movement. These movements are measured in real-time by a camera positioned at its designated location. The PERCLOS algorithm for yawn driver detection uses Haar-based cascade classifiers in measurements for driving conditions. We can separate indicators of drowsiness from reflex blinking using this method. And the icing on the cake, it's entirely non-invasive [2]. With accurate eye tracking and robust facial landmark localization, shape prediction models uncover latent indicators of drowsiness. Facial landmarks are used as the inputs, robustly cropped by potent detection algorithms. The module also carries an Eye Aspect Ratio (EAR) feature that calculates horizontal to vertical eye marker ratio for distance, waking dozing drivers, and along with related alarm [3].

Newly emerging research is revealing most of the path to detect this evil foe before it takes hold. Four approaches are described by Albadawi et al. [4] for this admirable cause. First, via the vehicle itself, tracking the driver's steering wheel and lane deviation. Then the bio-signals: electrocardiography (ECG), electroencephalography (EEG), and electrooculography (EOG). These are accurate but invasive to drivers. There is image processing whose region of interest is the eyes, the mouth, and the position of the driver's head. It is a preferred method because it is not invasive, and therefore the drivers will feel at ease. Besides anything else, drowsiness also descends upon the face, so s it is just a matter of determining if one will be dozing off while at the wheel. Among those environments, only a few databases are conscious of drowsiness, like NTHU-DDD [6], YawDDD [7], MRL Eye [8], UTA-RLDD [9], and the pick of the bunch, NITYMED [10]. The latter treasure contains video clips of drivers driving in real chaos, where conditions of continuously changing lights keep introducing themselves to test drowsiness with their mouths and eyes.

Clearly duplicated from the rest are the real ones that NITYMED captures. More realistically, sleepy drivers in real driving conditions are duplicated. Based on computer vision inspection with deep technologies like CNNs, it inspects status of the eyes open or closed [11]. Point correction close to eyes for this improvement is a step characterized by the lens adjusting to fit the area of interest. All these developments position NITYMED at a state-of-the-art sleep driver study agency. Figure 1 illustrates the entire procedure of drowsiness detection.

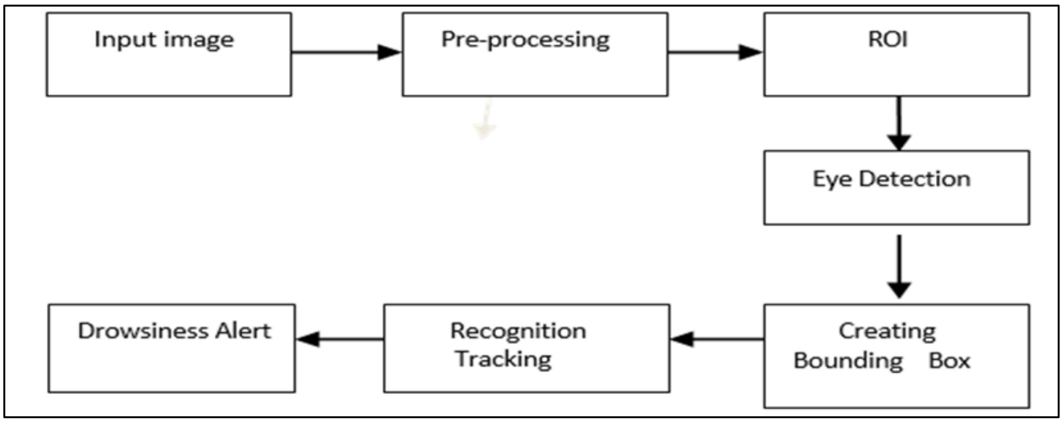


Figure 1. System Overview of Driver Drowsiness Detection in Real-Time Processing

In Figure 1, '1' denotes sleepiness and '0' denotes alertness. We perform ROI feature extraction after pre-processing the input image. ROI extracts salient features, i.e. eyes, mouth, and head pose with the help of one of several algorithms. During classification, we calculate eye movement directly and label the eyes with a bounding box. This classifies the driver as drowsy or highly alert. With further contribution from Vikranth et al. [5], a total of three CNN models, including InceptionV3 [11], VGG16 [12], and ResNet50V2 [13], are used. These models use transfer learning [14] and Media Pipe [15] for effective facial point detection and ROI extraction. A Fully connected network has been used by researchers for binary drowsiness classification. To approximate drowsiness while real driving, we take into consideration the likelihood of the ROI belonging to the drowsiness class. According to [15], we take blink duration into account; it needs to be within 100 to 300ms. If the eyes are shut for over 300 ms continuously, the driver is classified as drowsy. The system for detecting drowsiness acts as an intelligent tech watchdog while one drives.

Through the use of a combination of sensors, algorithms, and sophisticated AI, it monitors the behavior of the driver all the time. The moment there is a hint of sleepiness, it gives an early warning call of alarm before catastrophes happen. Computer vision and deep learning are the stars of the ensemble. Real-time identification of anomalous driving behavior is the secret to enhanced safety. While phone-based methods can quantify the distinction between pathological and healthy driving, they will not capture large nuances. Finer methods must be used to decompose driving dynamics more finely. This includes using more sophisticated classification algorithms for the more refined discrimination of driving anomalies, such as driver profile and road type, and incorporating the output from a multitude of sensors into more elaborate analysis [16]. The privacy-preserving techniques keep the driver information a well-guarded secret. Additionally, the development of monitoring technology can result in novel approaches in data analysis and visualization. Driving behavior can be correctly explained only if a myriad of variables is brought into account: environmental parameters, road types, vehicle information, accident nature, and even physical and bodily condition. The final target of a drowsiness detection system is accident prevention, protecting protected vehicle occupants as well as commercial fleets. By foreseeing precursors to fatigue in advance, when general alertness has not yet been compromised, it delivers crucial alerts to maintain drivers' alertness and competent driving skills on the road. The paper is organized as follows: it briefly reviews the literature on drowsiness detection, then provides a thorough materials and methods section, followed by a presentation of results and analysis for each CNN architecture, and suggests a new model for drowsiness detection.

2. Related Work

There are a number of works in the literature on drowsiness detection which are discussed below.

Park et al. [17], in their initial work, propose a new architecture for the task of deep drowsiness detection (DDD). The new system captures RGB videos of the entire field of the driver's face. The DDDD model finely incorporates three extremely potent networks: AlexNet, VGG-FaceNet, and FlowImageNet. By synchronizing their outputs, it effectively classifies drowsiness between video frames. To validate their findings, the authors visited the NTHU drowsy driver detection (NTHU-DDD) database and achieved an average accuracy of 73.06% in their experiment. Chirra et al. [18] focused on zooming in on the eye region for improved detection. They employed the Haar Cascade algorithm by Viola-Jones for eye detection. Their ROI is inputted into their CNN, triggered by an expert training set database, achieving 98% training accuracy, 97% validation accuracy, and 96.42% total test accuracy. Zhao et al. [19] took a different approach by utilizing facial feature points for drowsiness recognition. They employed an MTCNN (multi-task cascaded convolutional network) for face detection and key point detection, which extracts ROIs from the eyes and mouth. These are fed into their EM-CNN model, which has two states for the eyes and two for the mouth. They achieved a rate of 93.623% accuracy on the Biteda database compared to numerous other architectures. Last but not least, Phan et al. [20] introduced two methods for drowsiness detection based on facial characteristic points, with particular focus on the mouth and eyes. The Dlib library is used to process these points, with the help of thresholds, in order to find blinks and yawns.The second method makes use of MobileNet-V2 and ResNet-50V2 networks with the help of transfer learning. The authors gathered images from different sources to build a robust dataset and accomplished their task with a very good average accuracy of 97%. In their research paper, Rajkar et al. [21] made use of Haar Cascade for eye detection. After facial detection, they cropped every region of interest for an eye separately. Their proposed architecture was trained on two databases, YawDDD and Closed Eyes in the Wild. This time-consuming process achieved an accuracy of 96.82%. In an interesting study by Hashemi et al. [22], two CNNs were trained with transfer learning from VGG16 and VGG19, and one was created by the authors themselves for a dynamic drowsiness detection system.Haar Cascade initiated the face detection, and Dlib eye landmarks identified the areas to train their three networks from the ZJU Eye Blink database. Their new network achieved a high accuracy of 98.15%. Alameen and Alhothali [23] went a step ahead and suggested an ambitious 3D-CNN for learning spatio-temporal features. These learned features were passed to a long short-term memory (LSTM) framework. They suggested two models, 3D-CNN+LSTM (A and B), tested on the MDAD and YawDDD datasets, with good accuracy of 96% on YawDDD. Gomaa et al. [24] finally proposed some new architectures by combining CNN with LSTM, where CNNs include ResNet, VGG-16, GoogleNet, and MobileNet. The authors introduced their new architecture, "CNN-LSTM." They conducted large-scale training and testing on the famous "NTHU" dataset. Their network obtained 98.3% accuracy on the training dataset and 97.31% accuracy on the testing dataset, both of which were better compared to four other networks. Singh et al. [25] suggested an EAR-based system fused with PERCLOS. The new fusion predicts eye closure percentage over time. They employed Dlib for precise extraction of eye points important in EAR. Their system recorded 80% accuracy. Lastly, Tibrewal et al. [26] proposed a state-of-the-art CNN architecture. It was trained and tested with the MRL eye database, where images of a single eye were provided. They utilized the Dlib library for extracting the eyes' region of interest. With a focus on the eyes' state, they attained an average drowsiness detection accuracy of 94%.

3. Proposed Drowsiness Detection System

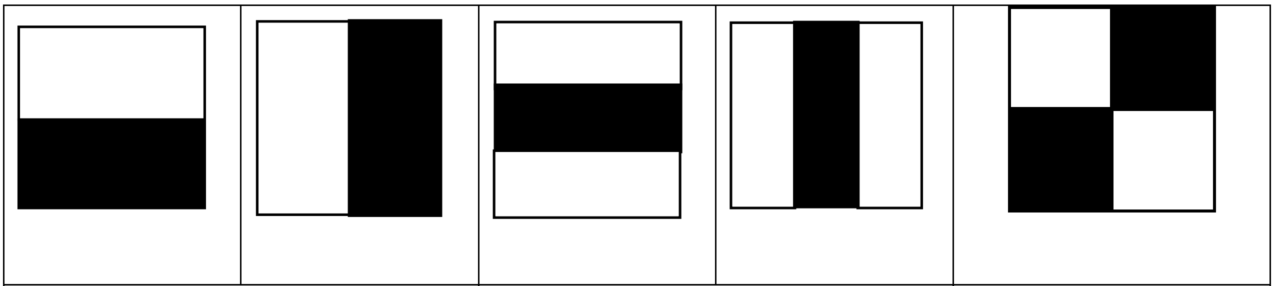
Proposed Drowsiness Detection System Drowsy driving is a silent threat to highway safety worldwide. It results in accidents, injuries, and gruesome deaths everywhere in the world. In spite of numerous attempts to bring attention to this subject, drowsy driving continues to be a persistent shadow. Drowsy driving casts an even longer shadow on long-distance truck drivers, night shift workers, and individuals who have undiagnosed sleep disorders. The real challenge lies in diagnosing drowsiness early enough to sound the alarm. Existing drowsiness detection systems are more likely to depend on one-tool solutions. Either physiological signal measurement or driving behavior monitoring may be unsuitable for actual traffic conditions. The systems also neglect to respond to individual variation in symptoms and driving styles.

Haar Cascade initiated the face detection, and Dlib eye landmarks identified the areas to train their three networks from the ZJU Eye Blink database. Their new network achieved a high accuracy of 98.15%. Alameen and Alhothali [23] went a step ahead and suggested an ambitious 3D-CNN for learning spatio-temporal features. These learned features were passed to a long short-term memory (LSTM) framework. They suggested two models, 3D-CNN+LSTM (A and B), tested on the MDAD and YawDDD datasets, with good accuracy of 96% on YawDDD. Gomaa et al. [24] finally proposed some new architectures by combining CNN with LSTM, where CNNs include ResNet, VGG-16, GoogleNet, and MobileNet. The authors introduced their new architecture, "CNN-LSTM." They conducted large-scale training and testing on the famous "NTHU" dataset. Their network obtained 98.3% accuracy on the training dataset and 97.31% accuracy on the testing dataset, both of which were better compared to four other networks. Singh et al. [25] suggested an EAR-based system fused with

PERCLOS. The new fusion predicts eye closure percentage over time. They employed Dlib for precise extraction of eye points important in EAR. Their system recorded 80% accuracy. Lastly, Tibrewal et al. [26] proposed a state-of-the-art CNN architecture. It was trained and tested with the MRL eye database, where images of a single eye were provided. They utilized the Dlib library for extracting the eyes' region of interest. With a focus on the eyes' state, they attained an average drowsiness detection accuracy of 94%.

3.1 HAAR Cascade Algorithm

The Haar cascade is a clear sight for object detection. It is akin to a fine painter, identifying shapes from images and finding scale and position. Because of its simplicity, it is capable of running in real-time. It uses cascading windows, each of which scans through its features to identify possible objects. While Haar cascades can deliver speed, they cannot quite match the accuracy of modern detection technologies.



a) Edge Features	b) Line Features	c) 4-rectangle feature

Figure 2. HAAR Feature Detectors a) Edge Features b) Line Features c) Four-Rectangle Features (Source citation https://sgino209.medium.com/custom-object-detection-with-haar-cascade-vj-93c197b59d5b)

Haar cascade is a limitation that gives too many false positives. Although simple to use, it can be computationally intensive. The AdaBoost-trained structured cascade offers a computationally efficient feature hierarchy estimate. The master engineering not only minimizes the cost of computation but also speeds up detection. Linear Discriminative Analysis (LDA) relies on two linearity basics' assumptions. First, it assumes linearity in face space, as well as a single, linear discrimination between classes.

Steps of the algorithm are now as follows:

Begin by calculating the within-class scatter matrix, denoted as *Sw*.

$$Sw=\sum\sum(x^{j}-\mu j)_{i}(x^{j}-\mu j)_{i} \qquad (1)$$

Where $x^{j}:i^{th}$ sample of class j C: Number of classes, N: Total samples in class j Compute the class scatter matrix *Sb*,

$$Sb=\sum(\mu j-\mu)(\mu j-\mu)^{T} \qquad (2)$$

Where μ represents the mean value of all classes. Calculate the eigen vectors of the projection matrix

$$W=(S^{-1}S) \qquad (3)$$

When there is a face in the frame, our powerful Haar cascade classifiers assume responsibility for eye detection. By considering eye closure rate and driver movement, the model reveals hidden drowsiness.

3.2 Convolutional Neural Networks (CNNs)

Eye conditions of a person are accurately classified through a CNN algorithm. The building blocks of the neural network platform that have played a role in the intricate properties of the feed-forward structure are the CNNs. The fundamental building blocks of a CNN are convolutional layers, activation functions, pooling layers, normalization layers, and fully connected layers.

- **Convolutional Layers:** Smoothly sliding image layers over input images with kernels. Kernels convolve, selecting features such as edges, texture, and patterns. One kernel sweeps a different feature map; different details are selected by different filters, and different step sizes vary as well, before binning the kernels for the best validation accuracy.

- **Activation Functions:** Activation functions serve as gateways to non-linearity in an attempt to render the network capable of learning intricate interdependencies between data. Excellent options that are commonly utilized include ReLU (Rectified Linear Unit) and its variants. They support sparsity and rapid convergence through straightforward reasoning: ReLU(x) = max(0, x).

- **Pooling Layers:** Pooling layers are like experienced carvers who shave away the irrelevant without their hands ever touching important details. Pooling layers decrease the spatial feature dimension of convolutional layer feature maps. Max pooling and average pooling are two of their tools, down-sampling based on the maximum or mean value within a window. This practice of selecting is called max pooling.

- **Normalization Layers:** Normalization layers, such as Batch Normalization, astonish by speeding up and stabilizing the learning process. Layer-wise normalization calms the variability in input data.

- **Fully Connected Layers:** In Figure 3, fully connected layers are seen at the top of the neural network to which all the preceding neurons are attached. Fully connected layers are usually placed at the end of the network and finalize classification, boxing features reduced by convolutional layers into useful knowledge.

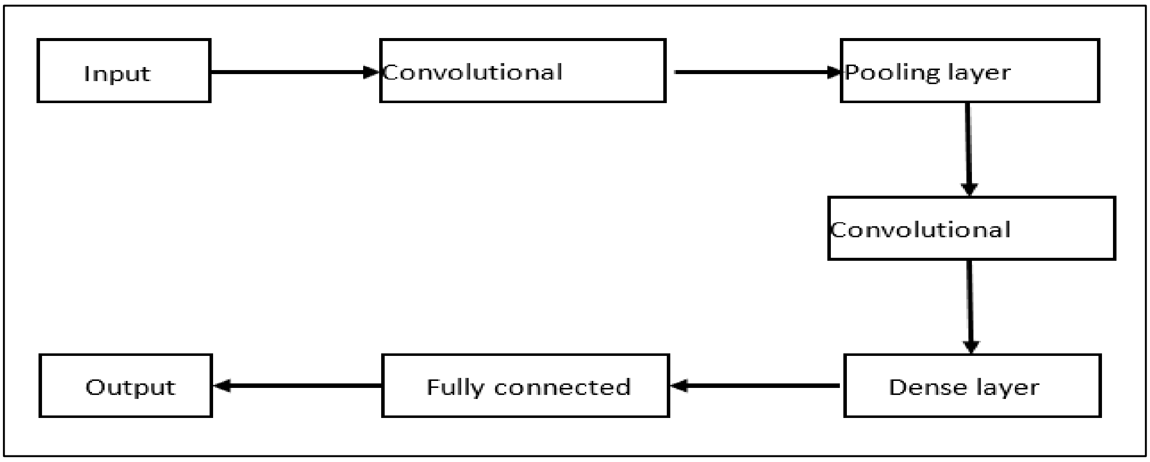


Figure 3. CNN Flow Diagram

4. Methodology

Algorithmic learning and information retrieval is a step-by-step process of deploying the drowsiness detection system, and it secures our roads. A step-by-step process of a drowsiness detection system is shown in Figure 4.

- **Accumulation of Information:** Starting with the world itself face shots, films. We're being sold sleepiness that will transform our lives in all directions and ways: fast asleep with eyes open, sleepy-eyed, or on the verge of falling over. The information will do its own talking multicolored and broken, against the image itself.

- **Preprocessing:** Alright, so the preprocessed data, we can use that. We are capturing images and videos and converting them into something understandable for the algorithm with no complexity. We normalize, resize, or crop to eliminate the noise and clean it, thus making it accurate.
- **Feature Extraction:** Hiya, Haar algorithm our dashing feature extraction beau. It is responsible for separating all the facial features, including our windows the eyes and the smiley face of the mouth. It accomplishes this via a cascade of Haar classifiers, working behind the scenes to find patterns hidden within the visible information.
- **Classification:** We used the CNN algorithm here. This is something that deep learning sorcery can accomplish, where an image pattern-detection neural network is trained to detect patterns in the image. The image is marked as sleepy or awake based on what we have learned in our previous work.
- **Visualization:** And finally, our system is taking shape. Our technology comes alive on the highway. It flashes a visual warning or quietly vibrates, alerting drowsy drivers placing safety in high gear.

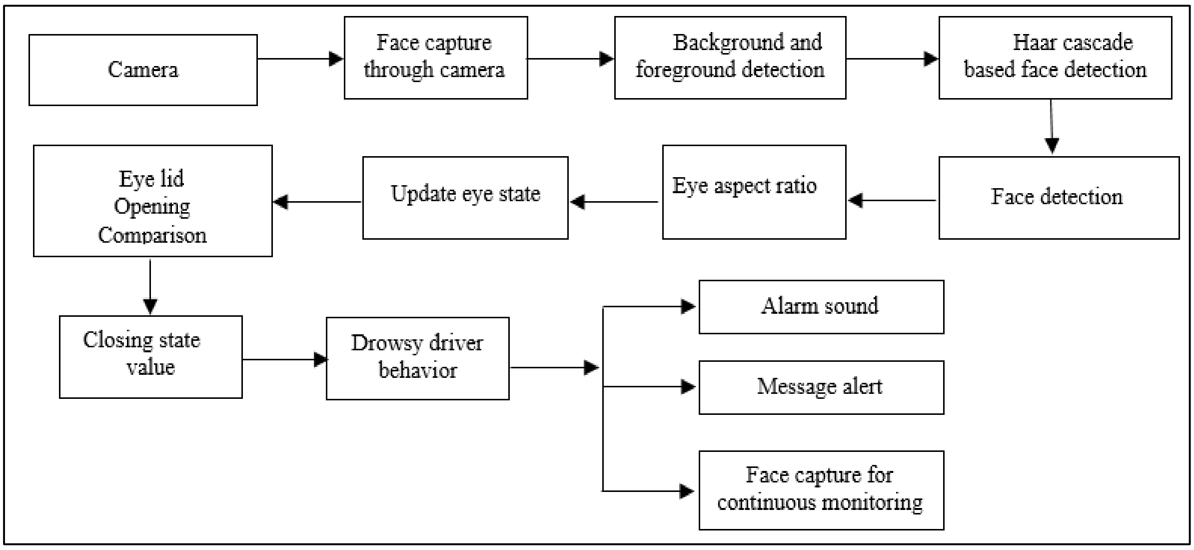


Figure 4: Step by Step Process Involved in Detection of Driver Drowsiness

In Figure 4, initially the camera identifies the foreground our driver's face and separates it from the background. This innovative approach employs a facial recognition method known as a HAAR cascade classifier, ensuring accurate face detection. Next, the Eye Aspect Ratio comes into play, calculating the width-to-height ratio of driver’s peepers. Normal behaviour establishes a baseline for what constitutes an abnormal eye aspect. The system updates the driver’s eye state by comparing this calculated ratio to the norm. Should the ratio dip below a specific threshold, the system declares that the driver’s eyelids have surrendered to sleep. It intelligently monitors the eyelid's dance, distinguishing between eyes wide open and shut. If those eyes closed for long period of time, it sounds the alarm bells: the driver may be drifting into drowsiness. A warning alarm reverberates through the car, jolting the driver back to alertness. But that’s not all this method can send a text alert to the driver’s phone or email a designated recipient, like a fleet manager. If those messages fail to reach the driver, an audible alarm will ring out, ensuring wakefulness prevails over drowsiness.

5. Implementation

The implementation of drowsiness detection weaves together two powerful allies: the Haar algorithm and the CNN algorithm. By uniting their strengths, we enhance both the accuracy and reliability of identifying fatigue. The Haar algorithm deftly pinpoints facial features, while the CNN algorithm classifies images with precision. Together, they form a dynamic duo, adept at spotting drowsiness in everyday situations. Recent works have made attention on their gaze toward smartphones, eliminating the need for bulky infrastructure. By harnessing accelerometers, magnetometers, and GPS sensors, these devices can detect erratic driving behaviours. They assess potential high-risk motorcycle accidents, cleverly analysing driving styles and identifying signs of impaired navigation such as sudden swerves that hint at drunk driving. Visualisation and gamification take this examination to another level, enhancing engagement while providing valuable insights.

In this exploration, we delve into modules designed specifically for real-time driver monitoring and drowsiness detection. With innovation at the helm, we’re steering towards a safe driving experience for everyone on the road. The major components used in the proposed system are listed below.

1. **Interface Creation:** Advancing technology to combat drowsiness behind the wheel is a roaring challenge in accident prevention. To keep drivers alert, we need a savvy

system that detects waning vigilance and sparks their awareness. Enter the Drowsy Driver Detection System, a heroic creation using non-intrusive, machine vision wizardry. This innovative approach tracks driver eye movements, revealing their fatigue levels while navigating the roads. Our module captures driver faces in real-time with a keen eye, and they are registered in the admin interface, ready for action.

2. **Face Capture:** In this module, we unleash the power of face detection using the HAAR Cascade algorithm. This clever tool employs this Haar classifiers, a marvel of object detection technology [26]. Following machine learning journey, it cascades through images to sniff out faces hidden in each snapshot. Our system excels at identifying both faces and facial expressions, completing tasks with finesse. By feeding the classifier a mix of positive and negative images, we train it to recognise features. Each distinctive characteristic emerges, derived by measuring pixel differences between white and black rectangles. This technique identifies faces across varied environments, showcasing the versatility of Haar-like features. we calculate features at lightning speed, predicting inter-class and intra-class variationscapturing every nuance of eyes, noses, and cheeks.

3. **Eye Detection:** Every face caught in our digital net hangs around for half a second, just long enough to sniff out its eye situation – mask or glasses, anyone? Enter our clever algorithm for eye detection! We slice the face into two horizontal segments: the upper from forehead to eyes and the lower from nose to chin. The lower segment is left out of our spotlight. Focusing on the upper, we divide it once more into two parts: upper, from forehead to eyebrow, and lower, from eyebrow to lower eyelash. Once we pry the eyes from the image, the current frame gracefully steps aside for a fresh one. Voila! The extracted eyes take centre stage, neatly categorised into left and right through vertical calibration.

4. **Abnormal Prediction:** With eyes now in focus, our next adventure begins: spotting whether they're open or closed. Intensity values come into play as we sketch a graph, analysing the distance through eyelashes and eyebrows to gauge eye status. A wider distance means sleepy eyes, while a closer distance indicates bright-eyed alertness. We boil down this distance by examining images, binarizing both eyes to find the magic threshold. If our system spies five consecutive frames with closed eyes, it sounds the alarm for the next five frames.

5. **Notification System:** This module serves double duty, alerting both admin and user when an abnormal prediction arises. If the eyes closed for less than 50% of the time, we serve a voice alert to encourage self-assessment. Then, like a digital courier, an SMS alert flies out via SMS gateway services. And behold, an abnormal face capture promptly makes its way into an email!

6. Experimental Analysis

Our framework focuses on deep learning for face drowsiness detection. Using benchmark accuracy metrics, we assess its performance. The proposed algorithm shines with a higher accuracy rate compared to traditional machine learning techniques, is illustrated in this section. Notably, the system demonstrates that CNNs deliver superior accuracy over several other algorithms. Additionally, our solution significantly reduces the false positive rate, ensuring more reliable results. We enhance our models' precision by weaving together various factors. Key elements like eye state and head tilt play crucial roles. Our emotion classification model seamlessly integrates into the system, trimming away irrelevant frames. This speeds up feature extraction and detection processes with finesse. Even with glasses or masks, our system excels in accuracy. We gauge the level of drowsiness, providing timely warnings. By doing so, we help drivers steer clear of unwanted accidents. We classify camera-captured frames into two prominent categories:

- Drowsy Frames
- Non-Drowsy Frames.

The experimental result screenshots for both Drowsy Frames and Non-Drowsy Frames on four different cases are shown in figure 5 and figure 6 respectively. Case 1: A face caught in the camera's vigilant gaze. Case 2: Masked expressions, yet glimmers of personality peek through the lens. Case 3: Spectacles perched on a nose, eyes gleaming, stealing the spotlight. Case 4: Eyes closed, Drowsiness detected.

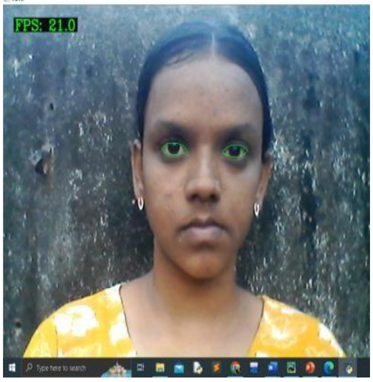
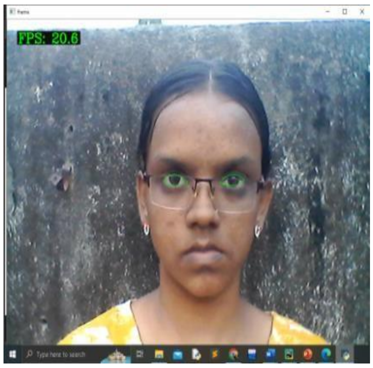
		
Case 1. Face Detected in Camera	Case 2. Face and Mask Detected in Wearing Mask	Case 3. Eyes Detected in Wearing Spectacles

Figure 5. Three Cases of Non-Drowsy Frames

Case 4. Drowsiness detected

Figure 6. Frames Classified as Drowsy

The actions taken for the above four cases are given in the Table 1. It explains the eye detection with system testing. It identifies moments when a driver's eyes are closed for those critical seconds. Upon detection, an alert spring into action, jolting the driver awake. Utilizing deep learning, we propose a novel framework to categorize eye conditions: open or closed.

Table 1. System Testing Results on Four Cases

Case	**Eyes detected**	**Eye closure status**	**Result**
Case 1	Yes	No	Eyes open
Case 2	Yes	No	Eyes open

Case 3	Yes	No	Eyes open
Case 4	Yes	Yes	Eyes closed, alarm sound, alert message

The Table 2, explains the comparison of our results with similar techniques by reducing the false rate. Our framework focuses on deep learning for face drowsiness detection. Using accuracy metrics, we assess its performance. The proposed algorithm shines with a higher accuracy rate compared to traditional machine learning techniques. Notably, the system demonstrates that CNNs deliver superior accuracy over several other algorithms. Additionally, our solution significantly reduces the false positive rate, ensuring more reliable results.

Table 2. Comparison of the Proposed Approach with Similar Techniques

Driver drowsiness detection studies	**Sensing method**	**Algorithm**	**Feature**	**Accuracy**
[21]	Camera based	CNN	Eye	92.42%
[26]	Camera based	SVM and AdaBoost	Eye	95.74%
Proposed system	Camera based	Haar cascade classifier and CNN	Eye	96.73%

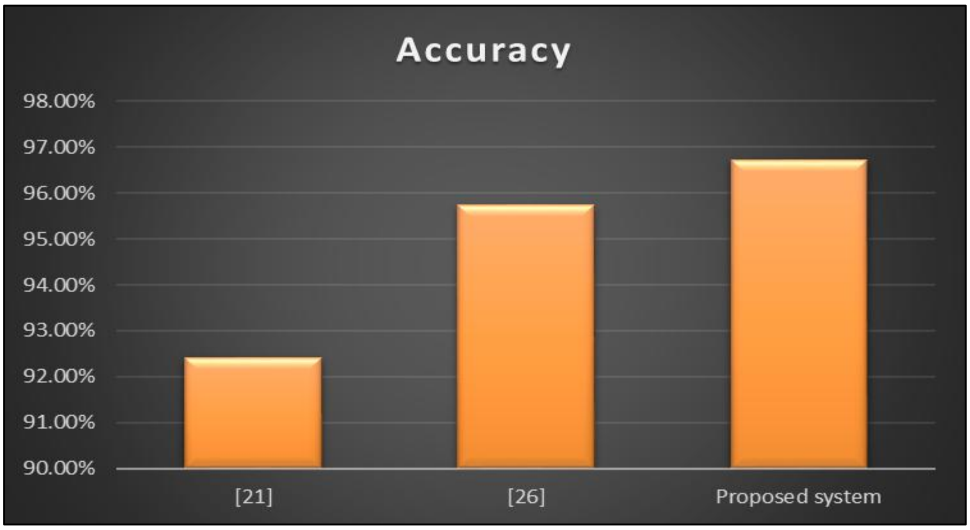


Figure 7. Accuracy Value of the Proposed System

4. Conclusion

Sleep and fatigue are road safety's worst enemies; they weigh down our response. The sleepier and drowsier the driver, particularly when driving late at night, the more devastating the effect. Motor accidents are phantoms waiting to seize life and society as a whole. The imagination mechanism is a catalyst for road safety among drivers. It establishes eye conditions as a drowsiness detection tool in the guise of an alarm guard. By applying the HAAR default algorithm for face detection, we detect all the eyes through our suggested crop Eye algorithm. The operation separates the face and can effectively distinguish between the left and right eyes. We check whether the eyes are open or closed by examining intensity values. We verify the difference between the eyelash and eyebrow, and we conclude for the eyes' sleep. Assuming a distance greater than some threshold value, we determine that the eyes are closed. Alarmed closed eyes are detected automatically in two adjacent frames. We have tested this in car scenes and lab scenes, and it functions smoothly. It operates in real-time with no processing overhead and is suitable for deployment in surveillance applications. On more than 90% of mixed faces, our system is a champion of road safety. However, there is one significant loophole: finding the eyes of spectacle-wearers. Refractive objects in the driver's rear view can obstruct the trajectory. In the days to come, we must address such loopholes and move towards embedded systems for development.

References

[1] Wang, Jing, Wenjing Li, Fang Li, Jun Zhang, Zhongcheng Wu, Zhun Zhong, and Nicu Sebe. "100-driver: A large-scale, diverse dataset for distracted driver classification." IEEE Transactions on Intelligent Transportation Systems 24, no. 7 (2023): 7061-7072.

[2] V B Navya Kiran, Raksha R, Anisoor Rahman, Varsha K. N, Nagamani NP, "Driver Drowsiness Detection", IJERT, Vol. 8, Issue 15, pp 33-35, 2020.

[3] Jain, Mahek, Bhavya Bhagerathi, and C. N. Sowmyarani. "Real-time driver drowsiness detection using computer vision." Int J Eng Adv Technol 11, no. 5 (2021).

[4] Reddy, P. V., J. D'Souza, S. Rakshit, S. Bavariya, and P. Badrinath. "A Survey on Driver Safety Systems using Internet of Things." Int. J. Eng. Res. Technol.(IJERT) 11 (2022): 180-185.

[5] Weng, Ching-Hua, Ying-Hsiu Lai, and Shang-Hong Lai. "Driver drowsiness detection via a hierarchical temporal deep belief network." In Computer Vision–ACCV 2016 Workshops: ACCV 2016 International Workshops, Taipei, Taiwan, November, (2016): 20-24, Revised Selected Papers, Part III 13, Springer International Publishing, (2017): 117-133.

[6] Abtahi, Shabnam, Mona Omidyeganeh, Shervin Shirmohammadi, and Behnoosh Hariri. "YawDD: A yawning detection dataset." In Proceedings of the 5th ACM multimedia systems conference, pp. 24-28. 2014.

[7] Fusek, Radovan. "Pupil localization using geodesic distance." In International symposium on visual computing, Cham: Springer International Publishing, (2018): 433-444.

[8] Ghoddoosian, Reza, Marnim Galib, and Vassilis Athitsos. "A realistic dataset and baseline temporal model for early drowsiness detection." In Proceedings of the ieee/cvf conference on computer vision and pattern recognition workshops, 2019.

[9] Petrellis, Nikos, Stavros Zogas, Panagiotis Christakos, Panagiotis Mousouliotis, Georgios Keramidas, Nikolaos Voros, and Christos Antonopoulos. "Software acceleration of the deformable shape tracking application: How to eliminate the eigen library overhead." In Proceedings of the 2021 European Symposium on Software Engineering, (2021): 51-57.

[10] Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. "Rethinking the inception architecture for computer vision." In Proceedings of the IEEE conference on computer vision and pattern recognition, (2016): 2818-2826.

[13] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Identity mappings in deep residual networks." In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14, Springer International Publishing, (2016): 630-645.

[14] Torrey, Lisa, and Jude Shavlik. "Transfer learning." In Handbook of research on machine learning applications and trends: algorithms, methods, and techniques, IGI global, (2010): 242-264.

[15] Lugaresi, Camillo, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang et al. "Mediapipe: A framework for building perception pipelines." arXiv preprint arXiv:1906.08172 (2019).

[16] Paramarthalingam, Arjun, Ashokkumar Janarthanan, Amirthasaravanan Arivunambi, Srinivas Saravanan Ariyangavu, Hariprasath Senthamaraikannan, and Rekha Ganapathy. "AI-Powered Virtual Mouse Control Through Hand Gestures With Computer Vision." In Harnessing AI in Geospatial Technology for Environmental Monitoring and Management, IGI Global Scientific Publishing, (2025): 61-76.

[17] Park, Sanghyuk, Fei Pan, Sunghun Kang, and Chang D. Yoo. "Driver drowsiness detection system based on feature representation learning using various deep networks." In Asian conference on computer vision, Cham: Springer International Publishing, (2016): 154-164.

[18] Chirra, Venkata Rami Reddy, Srinivasulu Reddy Uyyala, and Venkata Krishna Kishore Kolli. "Deep CNN: A Machine Learning Approach for Driver Drowsiness Detection Based on Eye State." Rev. d'Intelligence Artif. 33, no. 6 (2019): 461-466.

[19] Zhao, Zuopeng, Nana Zhou, Lan Zhang, Hualin Yan, Yi Xu, and Zhongxin Zhang. "Driver fatigue detection based on convolutional neural networks using em-CNN." Computational intelligence and neuroscience 2020, no. 1 (2020): 7251280.

[20] Phan, Anh-Cang, Ngoc-Hoang-Quyen Nguyen, Thanh-Ngoan Trieu, and Thuong-Cang Phan. "An efficient approach for detecting driver drowsiness based on deep learning." Applied Sciences 11, no. 18 (2021): 8441.

[21] Rajkar, Ajinkya, Nilima Kulkarni, and Aniket Raut. "Driver drowsiness detection using deep learning." In Applied Information Processing Systems: Proceedings of ICCET 2021, Springer Singapore, (2022): 73-82.

[22] Hashemi, Maryam, Alireza Mirrashid, and Aliasghar Beheshti Shirazi. "Driver safety development: Real-time driver drowsiness detection system based on convolutional neural network." SN Computer Science 1, no. 5 (2020): 289.

[23] Alameen, Sara A., and Areej M. Alhothali. "A Lightweight Driver Drowsiness Detection System Using 3DCNN With LSTM." Computer Systems Science & Engineering 44, no. 1 (2023).

[24] Gomaa, Mohamed Waheed, Rasha O. Mahmoud, and Amany M. Sarhan. "A cnn-lstm-based deep learning approach for driver drowsiness prediction." Journal of Engineering Research 6, no. 3 (2022): 59-70.

[25] Singh, Jagbeer, Ritika Kanojia, Rishika Singh, Rishita Bansal, and Sakshi Bansal. "Driver drowsiness detection system: an approach by machine learning application." arXiv preprint arXiv:2303.06310 (2023).

[26] Tibrewal, Madhav, Aayush Srivastava, and R. Kayalvizhi. "A deep learning approach to detect driver drowsiness." Int. J. Eng. Res. Technol 10 (2021): 183-189.

[27] Anand, S., M. Tamilarasan, and P. Arjun. "A study on curvature scale space." International Journal of Innovative Research in Computer and Communication Engineering 2, no. 3 (2014): 168-174.