Journal of Autonomous Intelligence (2024) Volume 7 Issue 3
doi: 10.32629/jai.v7i3.1117

ORIGINAL RESEARCH ARTICLE

SleepyWheels: An ensemble model for drowsiness detection leading to accident prevention

Jomin Jose[1], Andrew J[2,*], Kumudha Raimond[1], Shweta Vincent[3], Jennifer Eunice R[3,*]

[1] *Department of Computer Science and Engineering, Karunya Institute of Technology and Sciences, Karunya Nagar, Coimbatore 641114, India*

[2] *Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal 576104, India*

[3] *Department of Mechatronics, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal 576104, India*

*** Corresponding authors:** Andrew J, andrew.j@manipal.edu; Jennifer Eunice R, jennifer23.phd@gmail.com

ABSTRACT

Approximately 30% of traffic accidents in the world are attributed to driver drowsiness. Various approaches have been suggested by different research teams to detect drowsiness, but their methods have drawbacks. Some involve invasive techniques that cause driver discomfort, some involve too many false positives that cause distractions while driving while others rely on complex models that are too resource-intensive. In this paper, we present SleepyWheels, a novel drowsiness detection system. Our key insight is the combined use of two lightweight neural networks: a binary classifier and a facial landmark detector. This innovative approach minimizes false positives and proves resilient across diverse testing scenarios, such as different camera positions, variations in skin tone and when there is obscurement of facial features by objects like eyewear. Research outcomes include a working prototype of the system, a custom dataset for training the classifier and a trained model, which attains an impressive 97% accuracy rate. Deployment and testing were performed on Windows 10 but the system can be deployed on edge devices like Raspberry Pi. The lightweight nature of the models unlocks possibilities of deployment on mobile and embedded devices for use in vehicles.

Keywords: accident prevention; convolutional neural networks; deep learning; facial recognition

ARTICLE INFO

Received: 7 August 2023
Accepted: 12 October 2023
Available online: 5 January 2024

1. Introduction

According to WHO, sleepiness while driving accounts for up to 30% of the world's traffic accidents[1]. Up to 20% of traffic accidents in the USA[2], more than 20% in India[3] and 20% to 30% in Australia[4] are a result of drivers experiencing fatigue while operating their vehicle. The gravity of the situation becomes apparent when considering the annual estimates of fatalities. Approximately 150,000 individuals in India lose their lives each year in road-related incidents, with a significant portion of 60,000 fatalities attributed to driver drowsiness. The U.S. National Highway Traffic Safety Administration (NHTSA) further underscores the severity of this problem, reporting an excess of 100,000 traffic accidents annually, leading to a distressing toll of 1500 fatalities and over 71,000 individuals sustaining injuries[2]. These stark figures unequivocally emphasize the crucial necessity for vigilant oversight and timely intervention on the roads.

The gruesome effect of drowsy driving on society is amplified further when statistics of Vulnerable Road Users (VRUs) are taken into account. VRUs are the people other than the drivers who are involved in a crash, like pedestrians, cyclists, motorcyclists, etc. They do not have the protection offered by air bags and the vehicle's enclosure, unlike drowsy drivers, which makes them more susceptible to fatality during accidents. According to a report by PIARC Road Safety Knowledge 2023, in 2016, there were 1.35 million road traffic deaths worldwide, of which 54% were VRUs[5]. In the USA, there were 10,386 VRU fatalities in 2012, which increased to 12,530 in 2019[6]. While there is a lack of statistics about the percentage of VRUs directly affected by drowsy driving, the following facts help assess the situation. According to Sun Coast Traffic School, about two-thirds of pedestrian deaths in the USA occur at night or under low-light conditions[7]. According to the CDC, about 1 in 5 of the people who died in crashes involving a distracted driver in 2019 were not in vehicles[8].

Brake fails, worn-out tires, slippery roads, confusing traffic signs, absence of streetlights and inclined roads on hills are some of the leading causes of road accidents that are independent of the driver's alertness[9,10]. In contrast to these, forfeiting the use of protective gear, road rage, ignoring the speed limit, distractions, drunk driving, ignoring traffic lights, and driving while sleepy are the leading causes of road accidents that are directly linked to the driver's decision-making[11,12]. While road and vehicle maintenance can deal with driver-independent factors and law enforcement can deal with driver-related factors like drunkenness, red light jumping and the lack of safety gear, there exists a huge void in the area of keeping a check on how sleepy the driver is. This article focuses on monitoring a very important but overlooked factor that compromises the alertness of a driver: sleepiness. Our article outlines the creation and testing of the SleepyWheels app for drowsiness detection in drivers undertaking long travels.

The article is divided into the following sections. The second section describes the problem statement, motivation and the uniqueness of our solution. The third section presents a brief comparison between solutions suggested in state-of-the-art literature and how our system outperforms existing systems. The fourth section describes our proposed solution. The fifth section describes the components of our system. The sixth section is about system architecture and the steps involved in its workflow. The seventh section describes the dataset creation process. The eighth and ninth sections explain the process of training the convolutional neural network. The tenth section describes how the system was deployed and tested. The eleventh section presents the results obtained. The twelfth section concludes the article by summarizing its main points and suggesting areas for future research.

Motivation

Statistics confirm that most drowsiness-related accidents happen between 2 and 5 AM[2,13]. People like oil tanker drivers and truck drivers often have no choice but to work at these odd times. Behavioral monitoring[14], ECG[15] and other similar forms of supervision require special equipment attached to the driver, which might make driving for long periods of time uncomfortable. Vehicle or steering monitoring[16] also needs special equipment attached to working vehicle parts while being unable to solve the problem of numerous false positive detections. However, a solution based on computer vision does not impose such requirements. The main motivation behind choosing computer vision to tackle this problem is its practical advantages such as not requiring invasive gear that physically interferes with driving and not needing any more sophisticated equipment than a camera and a speaker.

Past research works have proposed putting multiple models to use in unison[17] and deploying isolated models independently. However, such systems have their shortcomings. Too many parallel machines bring confusion, computational load and a lack of decisiveness. In contrast, isolated deployments of such systems yield unpromising results. The approach of choosing just two models in unison proves to be a balanced solution in this trade-off. It is also worth noting that deep learning technology has long been seen as an improver of lifestyle and efficiency of tasks and as the key to reliable automation. Its ability to save lives has been restricted largely to breakthroughs that are put to use within the four walls of a medical facility. Hence, this project stands as a testament to prove that technology has the ability to save lives before a victim becomes a patient and reaches the hospital.

Numerous scientific studies in recent years have sought solutions to address the issue of accidents resulting from driver drowsiness. Some techniques involve the harnessing of physiological signals, including ElectroEncephaloGram (EEG)[18,19], ElectroOculoGram (EOG)[20,21], ElectroCardioGram (ECG)[15], and ElectroMyoGram (EMG)[22]. These methods involve the measurement of electrical activity across distinct areas of the human body, encompassing the brain, eyes, heart, and skeletal muscles. By analyzing the collected data, insights into the driver's real-time alertness can be derived. However, the implementation of these approaches mandates the placement of specialized equipment on the driver's body, causing discomfort and potential distraction during driving.

Alternatively, techniques reliant on vehicle status have been employed to gauge driver alertness. These methods monitor parameters such as the steering wheel's rotation, pressure on the accelerator pedal, and frequency of lane changes. Such strategies assess the driver's performance by correlating real-time vehicle data with measurements of physiological signals. Deng et al.[23] devised a mechanism leveraging facial landmarks to track the driver's facial expressions, identifying signs of drowsiness like yawning, eye closure, and frequent blinking. Verma et al.[24] adopted a dual VGG16 model ensemble for emotion detection. Notably, this approach employed image-derived regions of interest (ROI) for one CNN and facial landmark points for the other. Combining the scores from both models yielded the final decision. While the architecture utilized in this approach is outdated, its methodology served as a source of inspiration for the work detailed in this article.

In a different vein, Xing et al.[25] aimed to detect routine driver actions, including mirror checks, phone answering, and setting up in-vehicle video devices. They employed a Kinect camera to record these actions and extracted 42 features from the video feeds. Utilizing random forests to predict feature importance, they subsequently trained a feed-forward network achieving an 80.7% accuracy rate in predicting alertness. Yawn detection unbiased by factors like race, complexion, or facial accessories was the focus of Abtahi et al.[26]. They curated a diverse dataset to encompass a broad range of scenarios, which in turn inspired the diversity of the SleepyWheels dataset. Notably, their work yielded a reported accuracy of 60% in yawn detection. The unique solution proposed in this article is to parallelly combine the two best techniques available: facial landmark detection and convolutional neural network classification, in a simple but effective two-threaded ensemble model. The facial landmark detector is Google's BlazeFace, using which aspect ratios of the eyes and the mouth are calculated to detect sleepiness. The convolutional neural network architecture chosen is the EfficientNetV2B0, which once adapted for this use case via transfer learning, classifies a frame from the camera's feed as sleepy or alert.

2. Literature survey

Many solutions have been proposed aimed at tackling the problem of drowsiness detection in drivers. "A Review of Recent Developments in Driver Drowsiness Detection Systems" by Albadawi et al.[27] and "Trends and Future Prospects of the Drowsiness Detection and Estimation Technology" by Toshiya Arakawa et al.[28] are notable works that summarize and classify solutions published in the past decade.

The first class of techniques is based on a comparatively early approach which is based on analyzing driving patterns. This method employs a camera feed to observe the rotations of the steering wheel, subsequently comparing these rotations with deviations from the anticipated trajectory based on driving behavior analysis. The main challenge faced by these solutions is that they depend on potentially unreliable factors such as road conditions, vehicle status and driver expertise. Following is a discussion of solutions that fall in this category. Krajewski et al.[29] harnessed this connection between subtle adjustments of the steering wheel aimed at staying on the intended course and the driver's level of alertness, resulting in an 86% accuracy rate. Ma et al.[30] developed a drowsy driver detection model based on lateral distance, using lane curvature, position, and curvature derivative data obtained from a front bumper-mounted camera and a transportable instrumentation package system. Facial and head movement data collected from real-time video recording were used as ground truth. Signals were analyzed, and SVM and neural network algorithms achieved detection accuracies over 90%. Arefnezhad et al.[31] developed a non-invasive system using steering wheel data and enhancing classification accuracy through feature selection. They employed adaptive neuro-fuzzy inference systems (ANFIS), filter algorithms, and an SVM classifier on a dataset of 39 bus drivers, resulting in a 98.12% accuracy rate after optimizing ANFIS parameters with particle swarm optimization. Jeon et al.[32] introduced a method for detecting drowsy driving based on vehicle sensor data from the steering wheel and pedal pressure. They classified drowsy driving into long-time and short-time types and developed an ensemble network model with convolutional neural networks specialized for each type. Imbalanced data handling was used for efficient training. Using a dataset of 198.3 hours of in-vehicle sensor data, the model achieved up to 94.2% accuracy in detecting drowsiness during driving.

The second class of techniques analyzes neuromuscular data generated using Electrocardiography (ECG), Electrooculography (EOG), Electroencephalography (EEG), etc. to judge alertness. These techniques can yield accuracies of over 90%. However, the main challenges faced are that they are expensive and mandate the attachment of numerous sensors on the body which make these solutions impractical for regular use. Here is a discourse on some solutions from this category. Li et al.[33] introduced a driver drowsiness detection system using EEG signals and an SVM-based posterior probabilistic model. The driver wore a smartwatch and a Bluetooth EEG. Unlike traditional systems with discrete labels, this model provides a continuous measure for drowsiness between 0 and 1, allowing more granularity. The system achieved accuracy rates of 91.92% for drowsy, 91.25% for alert, and 83.78% for early warning states. Taran and Bajaj[34] introduced a drowsy driver detection method based on adaptive Hermite decomposition of EEG signals. They used evolutionary optimization algorithms to adaptively select Hermite functions for each signal from the MIT/BIH database. Extracted features from Hermite coefficients were tested with various classifiers, with Extreme Learning Machine (ELM) achieving the highest accuracy of 92.28%. Guede-Fernández et al. introduced a novel algorithm based on respiratory signal variations. They used three sensors for respiratory signal tracking, conducting tests with 20 volunteers in a simulator cabin. Their algorithm analyzed respiratory rate variability (RRV) to detect driver alertness changes, achieving a 90.3% sensitivity and 96.6% specificity by combining two methods for error reduction. Koh et al.[35] developed a method utilizing PPG signals from sensors on fingers and earlobes. With 20 subjects in a driving simulator, they analyzed LF, HF, and LF/HF values, using Telescan and KITECH programs. Drowsiness was detected through decreases in LF and LF/HF values and an increase in HF value, showing significant differences in PPG signals for alert and drowsy states.

The third category of methods utilizes computer vision techniques. This involves the analysis of intermittently captured images or video frames from a camera to identify behavioral indicators such as yawning, recurrently lowering one's head, frequent blinking, extended periods of closed eyes, etc. These methods are less intrusive and comparatively cheap. The main challenges faced when implementing these methods are the requirement of a well-lit environment, numerous false positive alarms and the use of large performance-intensive models that cannot run on edge devices. Here is a discussion of notable works in this domain. Danisman et al.[36] devised an approach that establishes a correlation between the degree of eyelid closure and the level of drowsiness. Their methodology is grounded in the assumption that the frequency of blinks per minute rises proportionally with sleepiness. Consequently, this parameter was employed to classify drivers into the three distinct levels of drowsiness delineated within their research study. You et al.[37] introduced a real-time driver fatigue detection algorithm based on facial motion information entropy. It consists of four modules: face positioning, feature vector extraction, face feature vectors, and fatigue judgment. Using an open-source dataset, they achieved 94.32% accuracy, employing SVM for judgment based on motion information entropy. Knapik and Cyganek[38] introduced a driver fatigue detection system using long-range infrared thermal imaging, focusing on yawning detection. They created a specialized dataset and employed cascaded modules for face, eye corners, and yawn detection. By using thermal voxel sum methods, they achieved 71% accuracy for cold voxels and 87% for hot voxels, triggering an alarm upon detecting fatigue. Dua et al.[39] modified the Viola-Jones object detection algorithm to measure drowsiness by studying the nature, duration and frequency of yawns. In modern times, techniques that use Convolutional Neural Networks (CNNs) have been very effective for use cases like image classification/ segmentation, object detection/tracking and recognition of emotion. Dwivedi et al.[40] employed shallow CNNs to achieve 78% accuracy in detecting drowsiness in drivers. Park et al.[41] developed a triple-layered architecture employing an AlexNet[42], a VGG-FaceNet[43] and a FlowImageNet[44] for extracting low-level to complex behavioral features in a step-by-step manner, garnering an accuracy of 73%. Deep learning networks like CNNs detect sleepiness very accurately but typically have high memory and processing requirements which makes their installation on edge devices unideal. However, in recent times, EfficientNets are closing the gap between performance and lightweight models.

The fourth category consists of hybrid solutions that use a combination of the solutions that fall under previously discussed categories. Here is a discussion of some notable works in this category. Saito et al.[45] introduced a driver assistance system identifying drowsiness through eyelid, steering wheel, and lane departure cues. It takes control if the driver doesn't respond, achieving 100% accuracy in specific conditions with 20 participants. Mehreen et al.[46] developed a non-invasive system using a wearable headband with accelerometer, gyroscope, and EEG electrodes. They achieved 92% accuracy with linear SVM after feature selection, combining multiple signals and analyzing driver behavior. Gwak et al. explored early drowsiness detection using vehicle-based, physiological, and behavioral metrics with 16 participants. They extracted 80 features from data and videos, then used random forest and majority voting classifiers. Random forest achieved 82.4% accuracy for alert vs. slightly drowsy, and majority voting reached 95.4% for alert vs. moderately drowsy states. Leng et al.[47] introduced a wearable device that combines motion and biomedical sensors to detect driver drowsiness via a mobile app. Their system uses a wristband with galvanic skin response and photoplethysmogram sensors, plus a smartwatch with accelerometer and gyroscope sensors to monitor steering wheel movement. An SVM algorithm analyzes data, achieving 98.3% accuracy.

The world's top automobile manufacturers already offer drowsiness detection as a feature in their products. Examples are Volvo's Driver Alert system, Volkswagen's Fatigue Detection System, Renault's Tiredness Detection Warning (TDW), Nissan's Driver Attention Alert (DAA), Jaguar Land Rover's Driver Fatigue Alert, Subaru's Wobble Alarm system, Honda's Driver Attention Monitor and BMW's Active Driving Assistant to name a few. But most of these rely solely on driving patterns formulated based on the driver's interaction with the pedals and steering wheel. The intent of the SleepyWheels research is to prove the efficacy of a simple computer vision-based ensemble model over the traditional methods used in today's automobile products. **Table 1** showcases a comparative analysis of a few systems developed using the idea of computer vision.

Table 1. Modern drowsiness detection systems.

Ref.	Sensor	Features	Algorithm	Metric	Dataset
Sleepy Wheels	Camera	Video feed of upper body	Ensemble model: landmark detector and CNN	Accuracy: 97%	Custom dataset
[48]	Smartphone camera	Frequency and duration of yawns and blinks, pulse, heart beat	Multichannel second-order blind identification	Sensitivity: 94%	Custom dataset
[49]	Smartphone	Touch response, PERCLOS, vocal data	Linear Support Vector Machines	Accuracy: 93%	Custom dataset called "Invedrifac"
[50]	EOG, ECG & EEG sensors	Frequency & duration of blinks/closed eyes, heart rate, alpha and beta bands power	Fisher's linear discriminant analysis method	Accuracy: 79.2%	MIT/BIH polysomnographic database
[51]	Camera	Facial expression, movements of eyelids, gaze & head	Kalman filtering tracking	AECS: 95% Yawn: 82% PERCLOS: 86%	Custom dataset
[52]	Camera	Driver's performance data, movement of the eyes	Support Vector Machines	Accuracy: 81.1%	Custom dataset
[53]	PPG, sensor, accelerometer, and gyroscope	Linear acceleration and radian speed of steering wheel, pulse, heart & respiratory rate and variability, stress level	Support Vector Machines	Accuracy: 98.3%	Custom dataset
[45]	Computers that generate images, modified steering system, automatic gearbox	Speed, steering angle, lateral position, opening degree, yaw angle, driver's input torque, eyelid, etc.	Series of mathematical operations	Up to 100% accuracy in assuming vehicle control when specified conditions were met	Custom dataset
[54]	Monitoring system, simulator	LF/HF, PERCLOS, SWA, etc.	RF and majority voting (KNN, SVM, logistic regression) classifiers	Accuracy, precision, sensitivity: RF classifier: Labels for slightly drowsy: 82.4%, 81.6%, 84.1% Majority voting: Labels for Moderately drowsy: 95.4%, 97.1%, 92.9%	Custom dataset
[37]	Camera	Face feature vectors	SVM	Accuracy: 94.32%	YawDD dataset
[55]	Three respiratory inductive plethysmography sensors	RRV and quality of the respiratory signals	Thoracic effort-derived drowsiness	Sensitivity: 90.3%	Custom dataset
[38]	Infrared sensor	Yawning	Cold and hot voxels	Accuracy: Cold voxels: 71%, Hot voxels: 87%	Custom dataset

3. Materials and methods

The proposed solution of the SleepyWheels system parallelly combines outputs of two state-of-the art techniques: facial landmark detection and convolutional neural network classification, in a simple but effective multithreaded ensemble model. The advantage of this approach is two-fold: firstly, the simplicity in design reduces the computation burden on the deployment platform thus opening doors to embedded systems and mobile platforms in contrast to using numerous sensors and multiple ML models in parallel; secondly, this approach provides better accuracy and reduced false positive alarm-triggers that upset the focus of the driver.

The facial landmark detector used is BlazeFace[56] from Google's Mediapipe[57]. It provides the information required to calculate aspect ratios of the eyes and the mouth which are used to detect sleepiness. The convolutional neural network model used is an EfficientNetV2B0 adapted for this use case via transfer learning. It is trained into a binary classifier that classifies a frame from the camera's feed as sleepy or alert.

3.1. Proposed deep learning architectures

3.1.1. EfficientNetV2B0 as binary classifier

EfficientNet[58] is a convolutional neural network architecture developed by Google. It is the brainchild of a new model scaling methodology whereby a model's depth, width and resolution are scaled uniformly instead of arbitrarily. Version 1 of these networks borrowed MobileNet's inverted bottleneck residual blocks and combined them with the novel squeeze-and-excitation blocks. The EfficientNetV2[59] family of models has some improvements like the use of fused-MBConv blocks along with regular MBConv blocks. These models are known to achieve higher accuracies with fewer parameters.

EfficientNets also adapt/specialize well when trained on new datasets through transfer learning. Fine-tuned EfficientNets often emerge to be among the state-of-the-art models while using a comparatively lower number of parameters when trained on benchmark datasets like CIFAR-100 (91.7%) and Flowers (98.8%).

In the Sleepywheels app, a custom top is added to a headless EfficientNetV2B0 loaded with weights trained on ImageNet. This network is trained using the techniques of transfer-learning and fine-tuning to develop the binary classification model that decides if a face/upper body shows signs of drowsiness or not. Some important architectural components of this network are explained below.

- Swish Activation: Swish is a multiplication of a linear and a sigmoid activation. It is proven to yield higher accuracies than ReLu in models like InceptionResNet and EfficientNet. The activation function is shown in Equation (1).

$$swish(x) = x\ sigmod(x) \tag{1}$$

- Inverted Residual/MBConv Block: Originally introduced in the MobileNetV2[60] architecture, inverted residual blocks start with a widening pointwise convolution, then apply a highly efficient depth-wise convolution which is followed by a compressing pointwise convolution. This approach largely decreases the number of trainable parameters.
- Squeeze and Excitation Block: The Squeeze and Excitation technique grants the network the special ability to treat different channels with different importance. Every channel's importance takes the form of a weight that is treated like a trainable hyperparameter. This new parameter is used to make smarter decisions when judging a multi-channel output, thus leading to higher accuracies.
- Fused-MBConv Block: Performing depth-wise convolutions within MBConv blocks reduces the number of parameters significantly but this does not always translate to improved training speeds because modern accelerators cannot take advantage of such architectural optimizations. Experiments reported an improvement in training speed when MBConv blocks that are part of the first few layers of an EfficientNet model were swapped with fused MBConv blocks that use regular convolution. Hence, researchers recommend the use of a neural architecture search to strike a balance in this efficiency versus training speed trade-off.

3.1.2. Facial landmark detector

MediaPipe Face Mesh[61] is a landmark detector that can output a stream of estimated locations of important landmarks on the human face like border points of the eyes, nose and mouth in real-time. It can infer the three-dimensional surface of a person's face without the need for a depth sensor.

It uses a combination of a detector model and a landmark generator model. The detector is tasked with providing the landmark generator a crop of the face from the original image. The landmark generator is tasked with predicting the facial surface and assigning new positions to the landmarks.

MediaPipe has optimized this pipeline so that the landmark generator can use previous facial crops to continue updating the landmark positions in case the face has not exited the bounds of the crop. This helps reduce unnecessary use of the detector thus saving power, memory and CPU utilization.

The detector uses a model called BlazeFace. It is a lightweight face detector that performs well on mobile GPUs. It is ideal for real-time face detection since it can run at speeds of over 200 frames per second (FPS) on most mobile devices. It performs feature extraction using a MobileNetV2-based model and anchoring using a Single Shot MultiBox Detector (SSD). It is capable of returning 468 3D facial landmarks in microseconds. Nodes representing these landmarks are shown in the depiction of the face mesh in **Figure 1**.

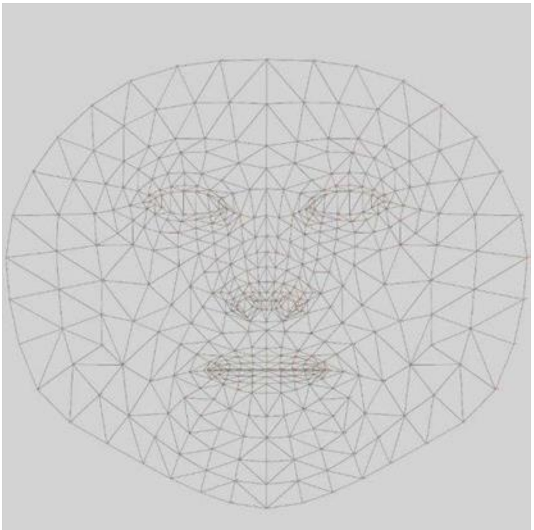

Figure 1. 468 landmarks returned by MediaPipe's Face Mesh[60].

Aspect ratio is typically defined as the proportional relationship between the width of an image and its height. It influences the perceived size and shape of an image. In computer vision, this ratio is often inversed when height is of more significance. Eye Aspect Ratio (EAR)[62] is a metric used to measure how open an eye is and Mouth Aspect Ratio (MAR)[63] is used to measure how open a mouth is EAR and MAR are calculated as the ratio of the average height of the eye or the mouth to its width. This means that a higher value represents the eye or the mouth image having a greater height, i.e., it being more open. There is a concept called Modified Eye Aspect Ratio (mEAR)[64] which is a variation of the EAR metric that may incorporate additional adjustments or features to improve the accuracy of the eye state detection, like PERCLOS[65], Temporal Filtering[64], Hysteresis[62], etc. but it is not used in this study in order to keep the processing of each video frame lean and performant. The face mesh landmarks with node numbers 30, 29, 28, 243, 22, 24, 463, 258, 259, 359, 254 and 252 are used to calculate EAR and the face mesh landmarks with node numbers 61, 39, 0, 269, 287, 405, 17, 181 are used to calculate MAR.

Table 2 gives the node numbers for the left and right eye landmarks from the face mesh and **Figure 2** shows the landmark points for a closed and open eye. Equation (2) calculates the EAR values by dividing the average eye height (average of the lengths of the lines p2–p6 and p3–p5) by the eye width (length of the line p1–p4).

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

Table 2. Node numbers of eye landmarks.

Variable	Left eye landmark	Right eye landmark
p1	30	463
p2	29	258
p3	28	259
p4	243	359
p5	22	254
p6	24	252

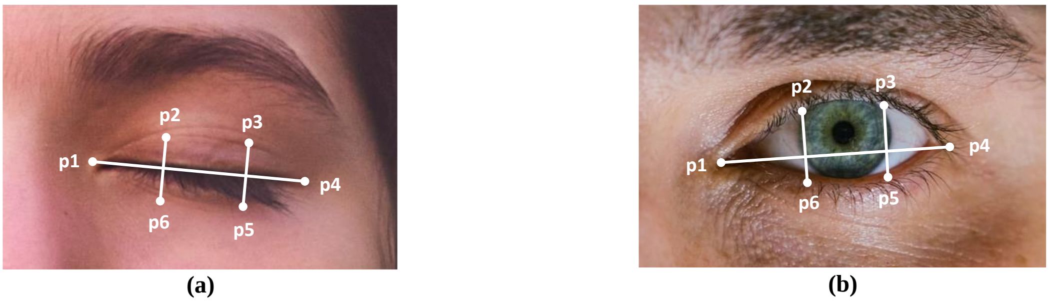

Figure 2. Measurement of eye aspect ratio (EAR): **(a)** Measurement made on a closed eye; **(b)** Measurement made on an open eye.

Table 3 gives the node numbers of the mouth landmarks from the face mesh and **Figure 3** shows the landmark points for a closed and open mouth. Equation (3) calculates the MAR values by dividing the average mouth height (average of the lengths of the lines p2–p8, p3–p7 and p4–p6) by the mouth width (length of the line p1–p5).

$$MAR = \frac{||p2 - p8|| + ||p3 - p7|| + ||p4 - p6||}{3 \times ||p1 - p5||} \tag{3}$$

Table 3. Node numbers of mouth landmarks.

Variable	Mouth landmark
p1	61
p2	39
p3	0
p4	269
p5	287
p6	405
p7	17
p8	181

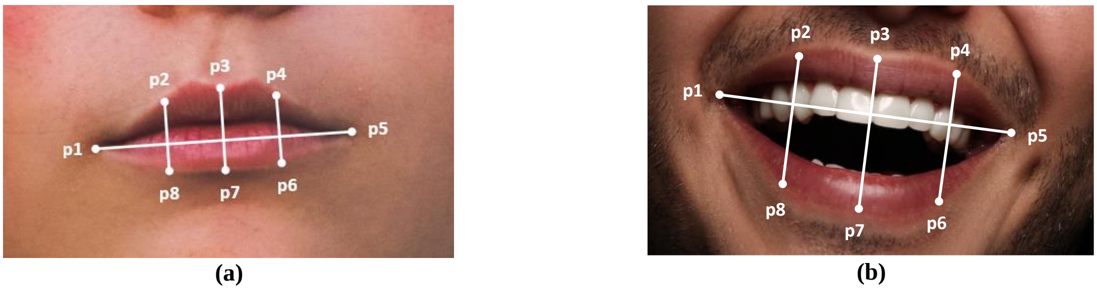

Figure 3. Measurement of mouth aspect ratio (MAR): **(a)** Measurement made on a closed mouth; **(b)** Measurement made on an open mouth.

3.2. Methodology

The architectural diagram of the SleepyWheels system is depicted in **Figure 4**. One video frame of the driver is captured by the camera (smartphone/webcam/independent camera) and sent to OpenCV. Two threads are created and deployed to run parallelly: one for the neural network and the other for the facial landmark detector. In the first thread, a series of functions transform the image into the dimensions expected by the neural network, after which it is fed into the neural network which processes it and returns a real number between 0 and 1. This value represents the probability of the driver being alert. The higher the value, the more probable it is for the driver to be alert. In the second thread, the image is processed by Mediapipe's Face Mesh, which returns a set of landmarks that are used to calculate eye and mouth aspect ratios. The aspect ratio of the mouth is used to determine if the driver is yawning. If so, the database is triggered to register the yawn.

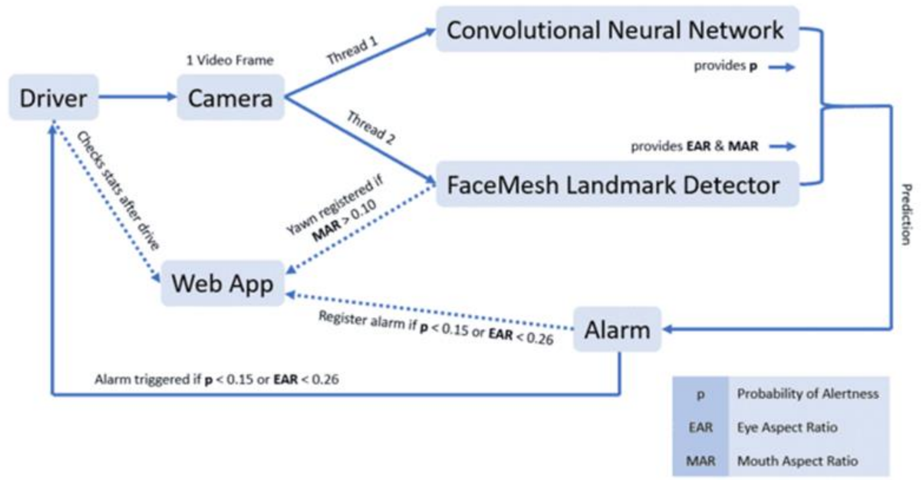

Figure 4. Architectural diagram of the SleepyWheels system.

The eye's bounding points are used to assess how sleepy the driver is, by calculating the eye aspect ratio and comparing it with predefined threshold values. The results of the two threads are taken into consideration together to decide whether to trigger the alarm or not. If the alarm is triggered, the timestamp is recorded in the database. Finally, at a later point of time, the driver can access the web dashboard to understand how sleepy he or she was while driving by viewing the number of alarms triggered and the number of yawns recorded.

3.3. Dataset creation

The image database used for carrying out transfer learning on the EfficientNetV2 convolutional neural network was custom-made by scraping Google Images. Google made changes to its Image Search API to prevent mass image scraping, which makes traditional image scraping difficult. An image scraping program by Jayden Oh Yikong hosted on Github, called Google-Image-Scraper was used to download images of sleepy and alert people. A python script in this scraper takes a list of search keywords such as: drowsy/sleepy boy/girl/woman/man/driver/portrait (with glasses/mask); yawning boy/girl/woman/man (with glasses/mask); clumsy/tired/slumber/doze off; alert/awake boy/girl/woman/man/driver/portrait (with glasses/mask); portrait of a boy/girl/woman/man (with glasses/mask) and the number of images to crawl as input arguments. It uses Selenium and Google Chrome browser and driver to navigate to the Google Images webpage, type in the keywords turn by turn, downloads pictures in PNG and JPEG formats and scrolls down to load more images. The downloads were a mixture of both relevant and irrelevant images, and hence a process of quickly going through each batch of downloads and manually deleting irrelevant images had to be done to ensure the purity of the data being supplied to the neural network for training. The sample data set is illustrated in **Figure 5**. The details of the number of images used for training and validation are is shown in **Table 4**.

Figure 5. Examples of Sleepy (Left two columns) and Awake (Right two columns) images in the dataset (scraped from Google Images).

Table 4. Dataset summary.

Type	Number
Train—Sleepy	4534
Train—Awake	4534
Validation—sleepy	504
Validation—awake	504
Total	10,076

About 16,000 images were downloaded, out of which 10,076 were chosen after a careful manual filtering process. The test-train split was performed manually, and two folders were created named train and validation, each housing its own two folders for sleepy and alert images. The train folder contains 9068 images, 4534 belonging to each class, while the validation folder contains 1008 images, 504 of each class. The Google Drive link to the dataset is provided in the SleepyWheels dataset[66].

3.4. Transfer learning

3.4.1. Need for transfer learning

Initially, a fully unlocked EfficientNetV2 B0 network was used for training from scratch. Even after using many regularization and initialization techniques, the model kept running into local minima. Accuracy was stagnating between 60% and 75%. Moreover, the dataset is too small to train the model from scratch and achieve sufficient generalization. Hence, the transfer learning method was used to take advantage of the pretrained filters in the initial layers that were trained on the generic ImageNet dataset.

3.4.2. Procedure

The machine learning frameworks Tensorflow and Keras were used in this project. Before training, the bottom layers of the EfficientNetV2 B0 network were locked with the pretrained weights from the ImageNet challenge. A max-pooling layer, a flattening layer, a dropout layer, a fully connected layer of 1024 neurons that use ReLU activation and an output layer consisting of one neuron with sigmoid activation were added to the top. At this point, the number of trainable parameters was 64.2 million. **Table 5** shows the optimal values settled on for each hyperparameter for the transfer learning and finetuning processes.

Table 5. Hyperparameter tuning details.

State/Hyperparameter	Original	Before transfer learning	Before fine tuning
Total parameters	7.1 million	70.15 million	70.15 million
Trainable parameters	7.1 million	64.22 million	70.08 million
Non-trainable parameters	0	5.91 million	61 thousand
ReduceLROnPlateau	warmed up from 0 to 0.256 then decayed by 0.97 every 2.4 epochs	Factor = 0.01, patience=3, min_lr = $1e^{-4}$	Factor = 0.01, patience = 3, min_lr = $1e^{-5}$
Initial learning rate		$3e^{-3}$	$3e^{-4}$
Optimizer	Adam	Adam	Nadam
Momentum	0.9	0.9	
Momentum Decay	0.9999	0.999	
Batch size	4096	64	
Epochs	350	50	
Neurons in penultimate layer	-	1024	
Activation at penultimate layer	-	ReLU	
Dropout at penultimate layer	-	0.7	
Image Augmentation	RandAugment[1], Mixup[3]	Rescale = 1.0/255, rotation_range = 180, width_shift_range = 0.1, height_shift_range = 0.1, shear_range = 0.1, zoom_range = [0.9,1.5], brightness_range = [0.5,1.1], horizontal_flip = True, vertical_flip = True, fill_mode = "nearest"	

The CNN's bottom layers were left frozen to avoid changing the core convolutional layers that were responsible to detect features in the images. Keeping them frozen at earlier stages helped to train the top classification layers better. An Nvidia RTX 3060 Ti GPU connected to a computer running the Ubuntu operating system was used for remote training via SSH. This model was trained on the custom 10,000 image SleepyWheels dataset, for 32 epochs with an image size of 224 × 224, learning rate of 0.00002, RMSProp optimizer, using binary_accuracy as metric, binary cross entropy as loss function, batch size 64 and a dropout rate of 0.6 on the flattened layer. Tensorboard was used to monitor performance metrics. At this stage, the model achieved training loss and accuracy of 0.5795 and 85% respectively and a test loss and accuracy of 0.6383 and 84% respectively. The accuracy and loss plots are illustrated in **Figure 6**.

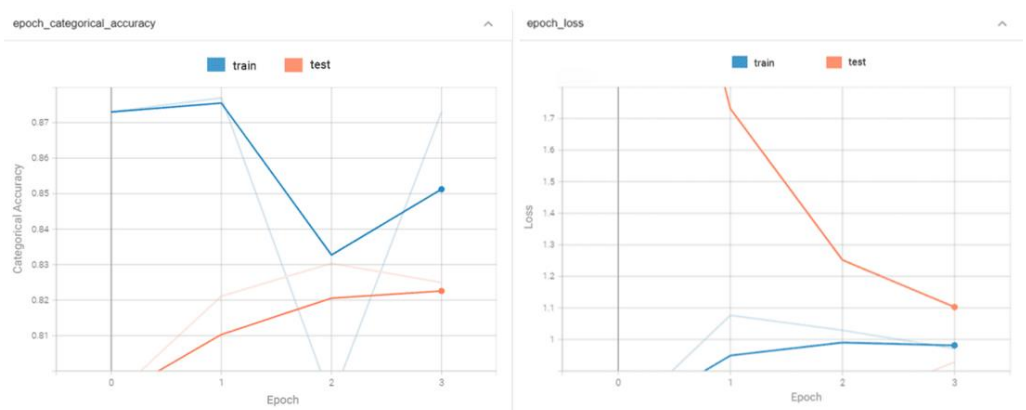

Figure 6. Train vs. Test Accuracy and Loss plot during transfer learning.

3.5. Finetuning the model

The next step was to fine tune the performance of the network by setting all the layers of the EfficientNetV2B0 model as trainable. This unlocked about 5 million previously untrainable parameters so as to try and achieve a bit more specialization to the current classification problem.

The model was trained using the checkpoint callback to make sure that the best version of the model during training was saved to Google Drive. The model was trained on the same train and validation images, with rescaling, rotation, width/height shift, shear, zoom, brightness, horizontal/vertical flip methods used for data augmentation, binary cross entropy as loss function, accuracy as metric, learning rate of 0.0001 with Adam optimizer and batch size of 64 for 50 epochs. Tensorboard was used to monitor performance metrics. Final training and testing accuracies were 97% and 96.92%. The model achieved about 12% more accuracy through fine-tuning. Plots of accuracy, loss and learning rate during fine-tuning are illustrated in **Figures 7** and **8**. The confusion matrix obtained is presented in **Table 6**.

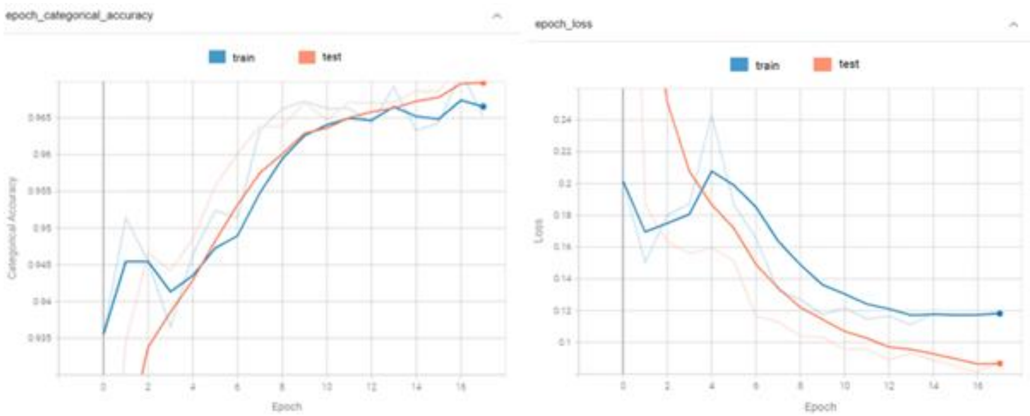

Figure 7. Train vs. Test Accuracy and Loss plot after fine-tuning the model.

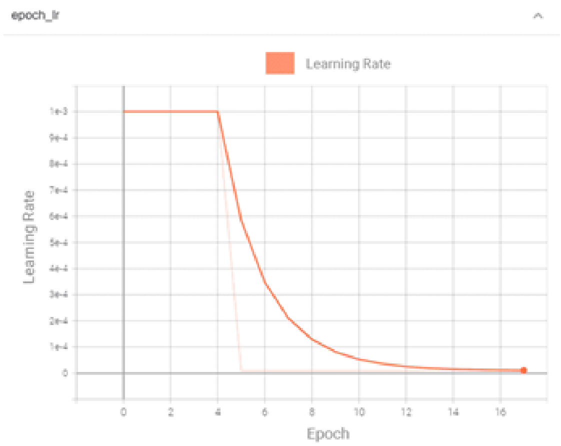

Figure 8. Learning rate plot during fine-tuning.

Table 6. Confusion matrix.

	Predicted: Awake	**Predicted: Sleepy**
Actual: Awake	498	15
Actual: Sleepy	18	481

4. Deployment and testing

4.1. Platform

For testing, the prototype system was deployed on a Personal Computer that runs the Windows 10 operating system. It has built-in stereo speakers, a USB 2.0 UVC WebCam, Python 3.9 and Jupyter Notebook installed. Once the model file model.h5 is placed in the same directory as detector.ipynb, the main program that contains the frame-processing loop is started by executing all cells of the detector.ipynb notebook in order.

For a production deployment setup, for example, a Raspberry Pi having TensorFlow Lite installed and equipped with speakers and a webcam could be used. Model optimizations like quantization (reducing the precision of weights and activations), model pruning (removing less important neurons or layers), etc. would be required to maximize performance. Such a setup can be installed inside a vehicle in the periphery of the driver in a way such that the driver's torso and head are visible to the camera.

4.2. Frame-processing loop

The first step is reading a frame from the camera. Next, this frame is resized and fed to the convolutional network on a separate thread. On the main thread, the frame is resized to a smaller width while maintaining the aspect ratio and converted to grayscale. This is fed to the facial landmark detector, which returns a list of rectangles where faces were found. Each rectangle is processed and landmarks returned are stored in a NumPy array. These landmarks are used to calculate aspect ratios of the eyes and mouth. Landmarks are used to plot the contours around the eyes and mouth. The mouth's aspect ratio is used to detect yawns. The average of left and right eyes' aspect ratios is used to decide if the eyes are closed for long enough to decide that the person is sleepy. This result is coupled with the probability returned by the neural network, and a decision is made whether to increment the frame counter state variable, which counts the number of frames in which the person was detected to be sleepy. When this counter exceeds the threshold (set as 60 by default), the alarm function is called on a separate thread.

4.3. Dashboard's backend

Keeping an XAMPP server running, a PHP file called "index.php" is written into htdocs folder of xampp. This page contains a basic UI which displays the yawn and alarm total counts and two tables for the respective entries accumulated. On PHPMyAdmin, a basic MySQL database called "SleepyWheels" is created, and two tables, yawns and alarms are created in it, each containing two columns: One column to represent the string "yawn" or "alarm" and one timestamp column. Two additional PHP pages are created, to which yawn and alarm data are posted by the frame processing loop. The database and the PHP files are hosted on a service called InfinityFree.

4.4. Testing

The prototype system was tested during two sessions, each in ill-lit conditions for a duration of one hour. During both sessions, the driver emulated drowsy behavior 10 times after random intervals. At the half hour mark, the camera angle and camera position were changed. The first session was conducted with a female person having a dark skin complexion. At the half hour mark, she started wearing a pair of glasses. The second session was conducted with a male person having a fair skin complexion. At the half hour mark, he started wearing a face mask that covers the nose and the mouth. **Figure 9** illustrates the cases when the driver has been sleeping causing the alarm to be triggered (left) and when the driver is awake causing no alarm triggers (right). **Figure 10** shows how the SleepyWheels system is able to detect sleepiness even if facial features are not detected due to a covered or turned face.

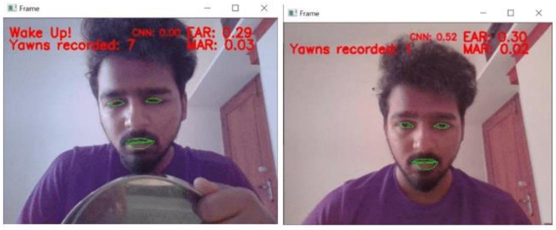

Figure 9. Driver is sleepy and alarm is triggered (left) | Driver is alert and alarm is not triggered (right).

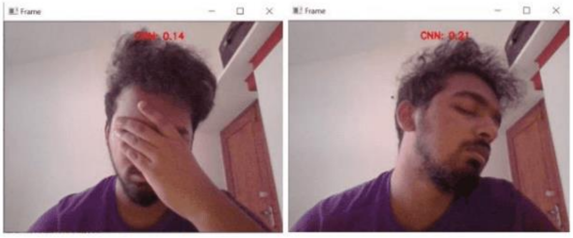

Figure 10. Ensemble model detects sleepiness in spite of missing facial features.

5. Results and discussion

Tables 7 and **8** represent data collected during the two one-hour tests in the form of confusion matrices. The term "Drowsy" represents the situation in which the driver is in a drowsy state and the term "Alert" represents the situation in which the driver is in an alert state. We omit the False Negative (Actual: Alert — Predicted: Alert) term in the accuracy formula since every instant the alarm is silent represents a detection of alertness, hence the correct detection of alertness is not quantifiable in this experiment. Averaging across the two tests, the accuracy of the SleepyWheels system is 95.5% and the precision is 95%. Hence it proves to be a system worth deploying in real-life scenarios. As anticipated, the landmark detector works with the CNN to better predict whether the driver is sleepy or not thus minimizing false alarms which would be counterproductive by being a distraction from driving. Also, in cases where the eyes or the mouth cannot be detected by the landmark detector, the CNN steps in and prevents generation of wrong results, in contrast to models that use the detector alone which cannot function reliably without the availability of facial features in the frame. The backend database updates in real-time and the webapp displays the list of alarm and yawn triggers to the driver. **Figure 11** shows a screenshot of the SleepyWheels web dashboard.

Table 7. Confusion matrix of Test 1.

	Predicted: Drowsy	**Predicted: Alert**
Actual: Drowsy	8	2
Actual: Alert	0	∞

Table 8. Confusion matrix of Test 2.

	Predicted: Drowsy	**Predicted: Alert**
Actual: Drowsy	9	1
Actual: Alert	1	∞

Figure 11. SleepyWheels Web Dashboard.

6. Conclusion

SleepyWheels is a novel approach toward drowsiness detection using a lightweight convolutional neural network in parallel with facial landmark detection, to achieve real-time driver drowsiness detection. The key finding of this paper is that this pair of models mitigate each other's weaknesses to provide high accuracy and low false positives using a technique that does not compromise on the driver's work routine and comfort. Another important result of this work is that our binary classifier proves to be yet another example of the sublime transferability of EfficientNet models for niche tasks. The binary classifier model achieves a validation accuracy of 97% on the custom-made SleepyWheels dataset. As per our test results, the SleepyWheels system has proved to be effective in a variety of test cases, such as absence of facial features while covering the eyes or the mouth, varying skin complexion of drivers, varied positions of the camera and varying angles of observation. It has the potential to be deployed successfully in vehicles. Its minimal computational load on the system unlocks possibilities for deployment on mobile and edge platforms after a few modifications. Potential deployment platforms include Raspberry Pi, Android Auto, Apple CarPlay and embedded devices that are designed to be fitted in the interior of a vehicle. Future directions that can be taken up by researchers include code optimizations in the frame processing loop, modifying the prototype and deploying it in an edge device to gather extensive test data from real drivers, using this test data to continuously improve the system via reinforcement learning, using infrared sensors for drowsiness detection in the absence of light, using more sophisticated modifications of EAR (Eye Aspect Ratio) and MAR (Mouth Aspect Ratio) and finally combining the classifier and landmark detector models into a single end-to-end model for better performance.

Author contributions

Conceptualization, JJ and AJ; methodology, JJ; software, JJ; validation, AJ and JJ; formal analysis, JJ; investigation, JJ; resources, JJ and AJ; data curation, JJ; writing—original draft preparation, JJ; writing—review and editing, AJ, SV and JJ; visualization, JJ; supervision, AJ, KR and JER; project administration, AJ and JER. All authors have read and agreed to the published version of the manuscript.

Acknowledgments

We thank our institutions for all the support provided for this research.

Conflict of interest

The authors declare no conflict of interest.

References

1. Saleem S. Risk assessment of road traffic accidents related to sleepiness during driving: a systematic review. Eastern Mediterranean Health Journal. 2022, 28(9): 695–700. doi: 10.26719/emhj.22.055
2. Drowsy Driving: | NHTSA. Available online: https://www.nhtsa.gov/book/countermeasures/countermeasures-work/drowsy-driving (accessed on 7 September 2023).
3. Sleep testing can reduce sleep disorder related road accidents in India | HT Auto. Available online: https://auto.hindustantimes.com/auto/news/sleep-testing-can-reduce-sleep-disorder-related-road-accidents-in-india-41620010214153.html (accessed on 7 September 2023).
4. SleepHealthFoundation | Drowsy Driving. Available online: https://www.sleephealthfoundation.org.au/drowsy-driving.html (accessed on 7 September 2023).
5. PIARC Road Safety Knowledge 2023 Vulnerable Road Users Factsheets. Available online: https://www.piarc.org/ressources/documents/source/Road-Safety-Knowledge-2023/eef5dda-41087-PIARC-Road-Safety-Knowledge-2023-Vulnerable-Road-Users-Factsheets.pdf (accessed on 7 September 2023).
6. Sewalkar P, Seitz J. Vehicle-to-Pedestrian Communication for Vulnerable Road Users: Survey, Design Considerations, and Challenges. Sensors. 2019, 19(2): 358. doi: 10.3390/s19020358
7. Sun Coast Traffic School—VRU Handout. Available online: https://www.suncoasttrafficschool.com/wp-content/uploads/2015/11/VRU-Handout_%C2%A9.pdf (accessed on 7 September 2023).
8. Distracted Driving | Transportation Safety | Injury Center | CDC. Available online: https://www.cdc.gov/transportationsafety/Distracted_Driving/index.html (accessed on 7 September 2023).
9. Harith SH, Mahmud N, Doulatabadi M. Environmental factor and road accident: A review paper. In: Proceedings of the International Conference on Industrial Engineering and Operations Management, 2019(MAR), 3409–3418.
10. Hammad HM, Ashraf M, Abbas F, et al. Environmental factors affecting the frequency of road traffic accidents: a case study of sub-urban area of Pakistan. Environmental Science and Pollution Research. 2019, 26(12): 11674–11685. doi: 10.1007/s11356-019-04752-8
11. Topolšek D, Babić D, Fiolić M. The effect of road safety education on the relationship between Driver's errors, violations and accidents: Slovenian case study. European Transport Research Review. 2019, 11(1). doi: 10.1186/s12544-019-0351-y
12. Wicaksana A, Rachman T. On the redesign of accident liability for the world of autonomous vehicles. Angewandte Chemie International Edition, 2018, 6(11), 951–952.
13. Drowsy Driving—Facts, Causes and Effects. Available online: https://www.medindia.net/patientinfo/drowsy-driving.htm (accessed on 8 August 2022).
14. Wang J, Chai W, Venkatachalapathy A, et al. A Survey on Driver Behavior Analysis from In-Vehicle Cameras. IEEE Transactions on Intelligent Transportation Systems. 2022, 23(8): 10186–10209. doi: 10.1109/tits.2021.3126231
15. Chui KT, Tsang KF, Chi HR, et al. An Accurate ECG-Based Transportation Safety Drowsiness Detection Scheme. IEEE Transactions on Industrial Informatics. 2016, 12(4): 1438–1452. doi: 10.1109/tii.2016.2573259
16. Chai M, Li S, Sun W, et al. Drowsiness monitoring based on steering wheel status. Transportation Research Part D: Transport and Environment. 2019, 66: 95–103. doi: 10.1016/j.trd.2018.07.007
17. Altameem A, Kumar A, Poonia RC, et al. Early Identification and Detection of Driver Drowsiness by Hybrid Machine Learning. IEEE Access. 2021, 9: 162805-162819. doi: 10.1109/access.2021.3131601
18. Paulo JR, Pires G, Nunes UJ. Cross-Subject Zero Calibration Driver's Drowsiness Detection: Exploring Spatiotemporal Image Encoding of EEG Signals for Convolutional Neural Network Classification. IEEE Transactions on Neural Systems and Rehabilitation Engineering. 2021, 29: 905–915. doi: 10.1109/tnsre.2021.3079505
19. Reddy TK, Arora V, Gupta V, et al. EEG-Based Drowsiness Detection with Fuzzy Independent Phase-Locking Value Representations Using Lagrangian-Based Deep Neural Networks. IEEE Transactions on Systems, Man, and Cybernetics: Systems. 2022, 52(1): 101–111. doi: 10.1109/tsmc.2021.3113823
20. Schmidt J, Laarousi R, Stolzmann W, et al. Eye blink detection for different driver states in conditionally automated driving and manual driving using EOG and a driver camera. Behavior Research Methods. 2017, 50(3): 1088–1101. doi: 10.3758/s13428-017-0928-0
21. Jiao Y, Deng Y, Luo Y, et al. Driver sleepiness detection from EEG and EOG signals using GAN and LSTM networks. Neurocomputing. 2020, 408: 100–111. doi: 10.1016/j.neucom.2019.05.108
22. Satti AT, Kim J, Yi E, et al. Microneedle Array Electrode-Based Wearable EMG System for Detection of Driver Drowsiness through Steering Wheel Grip. Sensors. 2021, 21(15): 5091. doi: 10.3390/s21155091
23. Deng W, Wu R. Real-Time Driver-Drowsiness Detection System Using Facial Features. IEEE Access. 2019, 7: 118727–118738. doi: 10.1109/access.2019.2936663
24. Verma B, Choudhary A. Deep Learning Based Real-Time Driver Emotion Monitoring. 2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES); September 2018. doi: 10.1109/icves.2018.8519595

25. Xing Y, Lv C, Zhang Z, et al. Identification and Analysis of Driver Postures for In-Vehicle Driving Activities and Secondary Tasks Recognition. IEEE Transactions on Computational Social Systems. 2018, 5(1): 95–108. doi: 10.1109/tcss.2017.2766884
26. Abtahi S, Hariri B, Shirmohammadi S. Driver drowsiness monitoring based on yawning detection. 2011 IEEE International Instrumentation and Measurement Technology Conference; May 2011. doi: 10.1109/imtc.2011.5944101
27. Albadawi Y, Takruri M, Awad M. A Review of Recent Developments in Driver Drowsiness Detection Systems. Sensors. 2022, 22(5): 2069. doi: 10.3390/s22052069
28. Arakawa T. Trends and Future Prospects of the Drowsiness Detection and Estimation Technology. Sensors. 2021, 21(23): 7921. doi: 10.3390/s21237921
29. Krajewski J, Sommer D, Trutschel U, et al. Steering Wheel Behavior Based Estimation of Fatigue. Proceedings of the 5th International Driving Symposium on Human Factors in Driver Assessment, Training, and Vehicle Design: Driving Assessment 2009. 2009. doi: 10.17077/drivingassessment.1311
30. Ma J, Murphey YL, Zhao H. Real Time Drowsiness Detection Based on Lateral Distance Using Wavelet Transform and Neural Network. 2015 IEEE Symposium Series on Computational Intelligence; December 2015. doi: 10.1109/ssci.2015.68
31. Arefnezhad S, Samiee S, Eichberger A, et al. Driver Drowsiness Detection Based on Steering Wheel Data Applying Adaptive Neuro-Fuzzy Feature Selection. Sensors. 2019, 19(4): 943. doi: 10.3390/s19040943
32. Jeon Y, Kim B, Baek Y. Ensemble CNN to Detect Drowsy Driving with In-Vehicle Sensor Data. Sensors. 2021, 21(7): 2372. doi: 10.3390/s21072372
33. Li G, Lee BL, Chung WY. Smartwatch-Based Wearable EEG System for Driver Drowsiness Detection. IEEE Sensors Journal. 2015, 15(12): 7169–7180. doi: 10.1109/jsen.2015.2473679
34. Taran S, Bajaj V. Drowsiness Detection Using Adaptive Hermite Decomposition and Extreme Learning Machine for Electroencephalogram Signals. IEEE Sensors Journal. 2018, 18(21): 8855–8862. doi: 10.1109/jsen.2018.2869775
35. Koh S, Cho BR, Lee J il, et al. Driver drowsiness detection via PPG biosignals by using multimodal head support. 2017 4th International Conference on Control, Decision and Information Technologies (CoDIT); April 2017. doi: 10.1109/codit.2017.8102622
36. Danisman T, Bilasco IM, Djeraba C, et al. Drowsy driver detection system using eye blink patterns. 2010 International Conference on Machine and Web Intelligence; October 2010. doi: 10.1109/icmwi.2010.5648121
37. You F, Gong Y, Tu H, et al. A Fatigue Driving Detection Algorithm Based on Facial Motion Information Entropy. Journal of Advanced Transportation. 2020, 2020: 1–17. doi: 10.1155/2020/8851485
38. Knapik M, Cyganek B. Driver's fatigue recognition based on yawn detection in thermal images. Neurocomputing. 2019, 338: 274–292. doi: 10.1016/j.neucom.2019.02.014
39. Dua M, Shakshi, Singla R, et al. Deep CNN models-based ensemble approach to driver drowsiness detection. Neural Computing and Applications. 2020, 33(8): 3155–3168. doi: 10.1007/s00521-020-05209-7
40. Dwivedi K, Biswaranjan K, Sethi A. Drowsy driver detection using representation learning. 2014 IEEE International Advance Computing Conference (IACC); February 2014. doi: 10.1109/iadcc.2014.6779459
41. Park S, Pan F, Kang S, Yoo CD. Driver drowsiness detection system based on feature representation learning using various deep networks. In: Proceedings of Computer vision—ACCV 2016 workshops; 20–24 November 2016; Taipei, Taiwan.
42. Iandola N, Han S, Moskewicz MW, et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. Feb. 2016, doi: 10.48550/arxiv.1602.07360.
43. Parkhi OM, Vedaldi A, Zisserman A. Deep Face Recognition. Procedings of the British Machine Vision Conference 2015. 2015. doi: 10.5244/c.29.41
44. Donahue J, Hendricks LA, Guadarrama S, et al. Long-term recurrent convolutional networks for visual recognition and description. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); June 2015. doi: 10.1109/cvpr.2015.7298878
45. Saito Y, Itoh M, Inagaki T. Driver Assistance System with a Dual Control Scheme: Effectiveness of Identifying Driver Drowsiness and Preventing Lane Departure Accidents. IEEE Transactions on Human-Machine Systems. 2016, 46(5): 660–671. doi: 10.1109/thms.2016.2549032
46. Mehreen A, Anwar SM, Haseeb M, et al. A Hybrid Scheme for Drowsiness Detection Using Wearable Sensors. IEEE Sensors Journal. 2019, 19(13): 5119–5126. doi: 10.1109/jsen.2019.2904222
47. Lee Boon Leng, Lee Boon Giin, Wan-Young Chung. Wearable driver drowsiness detection system based on biomedical and motion sensors. 2015 IEEE SENSORS; November 2015. doi: 10.1109/icsens.2015.7370355
48. Rajkumarsingh B, Totah D. Drowsiness Detection using Android Application and Mobile Vision Face API. R&D Journal. 2021, 37. doi: 10.17159/2309-8988/2021/v37a4
49. Zhang C, Wu X, Zheng X, et al. Driver Drowsiness Detection Using Multi-Channel Second Order Blind Identifications. IEEE Access. 2019, 7: 11829–11843. doi: 10.1109/access.2019.2891971

50. Dasgupta A, Rahman D, Routray A. A Smartphone-Based Drowsiness Detection and Warning System for Automotive Drivers. IEEE Transactions on Intelligent Transportation Systems. 2019, 20(11): 4045–4054. doi: 10.1109/tits.2018.2879609
51. Nguyen T, Ahn S, Jang H, et al. Utilization of a combined EEG/NIRS system to predict driver drowsiness. Scientific Reports. 2017, 7(1). doi: 10.1038/srep43933
52. Ji Q, Zhu Z, Lan P. Real-Time Nonintrusive Monitoring and Prediction of Driver Fatigue. IEEE Transactions on Vehicular Technology. 2004, 53(4): 1052–1068. doi: 10.1109/tvt.2004.830974
53. Liang Y, Reyes ML, Lee JD. Real-Time Detection of Driver Cognitive Distraction Using Support Vector Machines. IEEE Transactions on Intelligent Transportation Systems. 2007, 8(2): 340–350. doi: 10.1109/tits.2007.895298
54. Gwak J, Hirao A, Shino M. An Investigation of Early Detection of Driver Drowsiness Using Ensemble Machine Learning Based on Hybrid Sensing. Applied Sciences. 2020, 10(8): 2890. doi: 10.3390/app10082890
55. Guede-Fernandez F, Fernandez-Chimeno M, Ramos-Castro J, et al. Driver Drowsiness Detection Based on Respiratory Signal Analysis. IEEE Access. 2019, 7: 81826–81838. doi: 10.1109/access.2019.2924481
56. Bazarevsky V, Kartynnik Y, Vakunov A, et al. BlazeFace: Sub-millisecond Neural Face Detection on Mobile GPUs. Jul. 2019, doi: 10.48550/arxiv.1907.05047.
57. MediaPipe. Available online: https://mediapipe.dev/ (accessed on 8 August 2022).
58. Tan M, Le Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In: Proceedings of the International conference on machine learning, 2019. pp. 6105–6114.
59. Tan M, Le Q. Efficientnetv2: Smaller models and faster training. In: Proceedings of the International Conference on Machine Learning; 2021. pp. 10096–10106.
60. Sandler M, Howard A, Zhu M, et al. MobileNetV2: Inverted Residuals and Linear Bottlenecks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; June 2018. doi: 10.1109/cvpr.2018.00474
61. Kartynnik Y, Ablavatski A, Grishchenko I, Grundmann M. Real-time Facial Surface Geometry from Monocular Video on Mobile GPUs. Jul. 2019, doi: 10.48550/arxiv.1907.06724.
62. Soukupova T, Cech J. Real-Time Eye Blink Detection using Facial Landmarks. Center for Machine Perception, Department of Cybernetics Faculty of Electrical Engineering, Czech Technical University in Prague; 2016. pp. 1–8.
63. Gaur A, Kinage A, Rekhawar N, et al. Cursor Control Using Face Gestures. Proceedings of the 11th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2019). Published online August 1, 2020: 31–40. doi: 10.1007/978-3-030-49345-5_4
64. Dewi C, Chen RC, Jiang X, et al. Adjusting eye aspect ratio for strong eye blink detection based on facial landmarks. PeerJ Computer Science. 2022, 8: e943. doi: 10.7717/peerj-cs.943
65. George A, Routray A. Design and implementation of real-time algorithms for eye tracking and perclos measurement for on board estimation of alertness of drivers. Available online: http://arxiv.org/abs/1505.06162 (accessed on 7 September 2023).
66. SleepyWheels dataset. Available online: https://drive.google.com/drive/folders/1bhrgY8RcUFuD675oxcSLJkmtxY3Wxfg9 (accessed on 8 August 2022).