A Deep Learning Approach to Automated Drowsiness Detection in Drivers Using CNN Models for Enhanced Road Safety

Dr. Harish S Gujjar

Associate Professor and Head
Department of Computer Science,
Government First Grade College, Hosapete, Karnataka, India

Abstract

Traffic accidents are responsible for the majority of fatalities and injuries worldwide. According to the World Health Organization, around one million people die each year due to traffic-related injuries. Drivers who are sleep-deprived, fatigued, or not well-rested may fall asleep behind the wheel, putting both themselves and other road users at risk. Research indicates that drowsiness is a leading cause of major road accidents. In recent times, tired driving has emerged as the primary factor contributing to driver drowsiness, which in turn has led to an increase in road accidents. This has become a critical issue that needs urgent attention. The goal of many technologies is to improve real-time drowsiness detection. Numerous devices have been developed using various artificial intelligence algorithms to address this issue. My research focuses on detecting driver drowsiness by analyzing the driver's face, tracking eye movement, and alerting the driver with an alarm when drowsiness is detected.

In this paper, I utilized various Convolutional Neural Network (CNN) models, including Small CNN, VGG16, VGG19, and Inception, to classify distracted drivers based on the State Farm Distracted Driver Detection challenge on Kaggle. The deep learning framework employed for this task is Keras, which runs on top of TensorFlow. The system compares the extracted eye images with a dataset to detect drowsiness. If the system identifies closed eyes within a certain range, it triggers an alarm to alert the driver. If the driver opens their eyes after the alert, the system resumes tracking. The system adjusts the score based on eye status: the score decreases when eyes are open and increases when eyes are closed. This paper aims to address drowsiness detection with an accuracy of 94%, contributing to the reduction of road accidents.

Keywords: Face Detection, Python, Open CV, Keras, Alarm, Eye blinking and VGG19

1. Introduction

Drowsiness refers to the state of sleepiness, which can last for just a few minutes, yet its impact can be severe. The primary cause of sleepiness is typically exhaustion, which reduces alertness and concentration, though other factors such as lack of focus, medications, sleep disorders, alcohol consumption, or shift work can also contribute. Drivers often cannot predict when sleep might hit. While falling asleep at the wheel is clearly dangerous, even being tired can impair safe driving, as fatigue can hinder focus, reaction times, and decision-making. Studies show that one in twenty drivers has fallen asleep while driving.

Truck and bus drivers, particularly those with long commutes of 10 to 12 hours, are most at risk of drowsy driving. These individuals pose a greater danger to other road users than to themselves. Long-distance driving, especially when sleep-deprived, can lead to drowsiness, as can driving during hours when the body naturally requires rest. In such cases, the driver's fatigue becomes a major contributing factor to accidents on the road.

According to the Ministry of Road Transport and Highways, Government of India [1], police and hospital reports reveal that approximately 150,000 car accidents and over 1,750 deaths occur annually due to driver drowsiness. Drowsy driving is estimated to cause around 1,550 fatalities, 71,000 injuries, and financial losses of ₹1,049,195,562.50 (INR) [1]. In 2022 alone, a sleepy driver was a factor in 697 fatalities. The National Highway Traffic Safety Administration (NHTSA) acknowledges that quantifying the precise number of accidents or fatalities linked to drowsy driving is challenging, and the reported figures likely underestimate the true scope of the problem [5].

Fortunately, technology now allows us to detect driver fatigue and issue warnings before an accident occurs. Common signs of drowsy driving include frequent yawning, prolonged eye closure, and erratic lane changes. Recent research has made significant progress in developing effective methods for detecting driver drowsiness (DDD). Various techniques have been proposed to identify fatigue as early as possible in order to prevent accidents. In my research, drowsiness detection begins by identifying the driver's face, followed by analyzing the position of the eyes and their blinking patterns. A shape predictor with 68 landmarks is employed to examine facial features. Using a camera, typically a webcam, positioned toward the driver, the system captures the facial landmarks and estimates the position of the eyes.

The system processes the images to track the eyes' position, determining whether they are open or closed, as well as the blinking rate (the speed at which the eyes open and close). After a preset duration of closed eyes, an alarm sounds to alert the driver. The system starts with a score of zero for both open and closed eyes. If the eyes are closed, the score increases, and if they are open, the score decreases. If the score surpasses a certain threshold, the alarm will activate to warn the driver.

The remaining sections of this paper are organized as follows: Literature Review, Methodology, Experimental Results, Result Analysis, Conclusion, Authors Biography and References.

2. Literature Review

Various strategies have been implemented to enhance the efficiency and speed of the sleepiness detection process. This section focuses on the methods and approaches previously used to identify drowsiness. The first method relies on driving patterns, which consider vehicle characteristics, road conditions, and driving techniques. By analyzing steering wheel movements or deviations from the lane position, we can assess the driver's behavior and driving style [6][7].

Driving requires constant steering adjustments to maintain the vehicle's position within its lane. Krajewski et al. [6] developed a drowsiness detection method based on the correlation between fatigue and micro-adjustments, achieving an accuracy of 86%. Another method uses lane deviation to detect tiredness by tracking the vehicle's position relative to the lane and analyzing any signs of sleepiness [8]. However, driving pattern-based methods are highly dependent on the vehicle type, the driver's behavior, and the road conditions.

Another category of methods involves using physiological data from sensors, such as electrocardiograms (ECG), electroencephalograms (EEG), and electrooculography (EOG). EEG signals, in particular, provide insights into brain activity. Delta, theta, and alpha waves are the primary signals used to measure driver fatigue. When a driver is tired, theta and delta waves increase, while alpha waves remain mostly unchanged. According to Mardi et al. [9], this approach is one of the most accurate, with a sensitivity rate exceeding 90%. However, the major drawback of this system is its intrusive nature, as it requires multiple sensors attached to the driver, which can be uncomfortable. In contrast, non-intrusive biosignal methods are generally less accurate.

A final approach focuses on detecting facial features such as yawning, head position, and eye blinking [11]. In the eye closure method, driver fatigue is assessed by counting the frequency of eye blinks. A normal blink lasts between 0.1 to 0.4 seconds, meaning the driver typically blinks two to three times per second. If the driver is fatigued, the blink rate decreases. In our project, we use a camera positioned in front of the driver's face to monitor facial position and track eye blinking, helping to identify signs of fatigue.

3. Methodology

The general architecture of the model is straightforward and user-friendly. To operate the system, all that is required is capturing a video of the driver's face through a camera. The system then measures the eye blink score and triggers an alarm based on the results [13][14][15]. The process begins with face detection, followed by tracking the eyes and recording the eye closure events using OpenCV, which detects 68 facial landmarks as shown in **Fig 1**[10]. To determine whether the eyes are open or closed, we use the Euclidean Eye Aspect Ratio (EAR), which is calculated as: **EAR = (||p2 - p6|| + ||p3 - p5||) ÷ 2 * ||p1 - p4||** as shown in **Fig 2.**

This formula helps to accurately assess the eye's condition by comparing the distances between specific facial landmarks.

3.1 Face Calculation Using Euclidean Distance

Once the eyes are detected, the system evaluates whether they are open or closed. If the eyes are closed, the system will trigger an alert and continue to monitor the eye score to check if it exceeds the preset threshold. The alert will sound as long as the eyes remain closed. Once the driver's eyes are open, the system will resume tracking and monitoring the eye status.

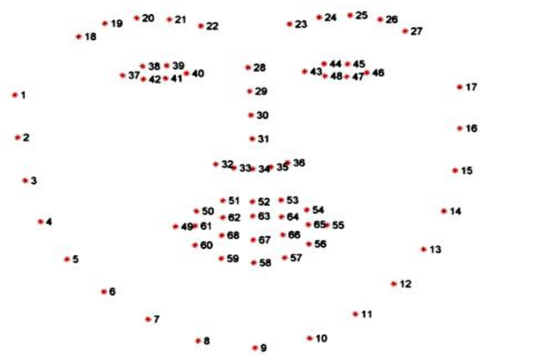
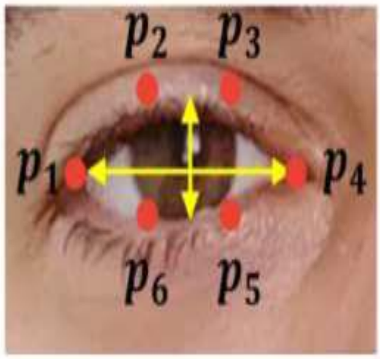
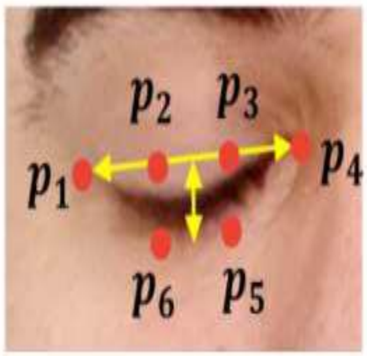


Fig 1: face landmarks detected by OpenCV



Open Eye Coordinates. In this eye coordinates p1, p2, p3, p4, p5, and p6 used for measure eye aspect ratio (EAR) for an open eye is approx. 0.24.



Close Eye Coordinates. Eye aspect ratio (EAR) for a close eye is approx. 0.15.

Fig.2. Six ocular landmarks are present both before and after the eyelids are closed.

4. Experimental Result

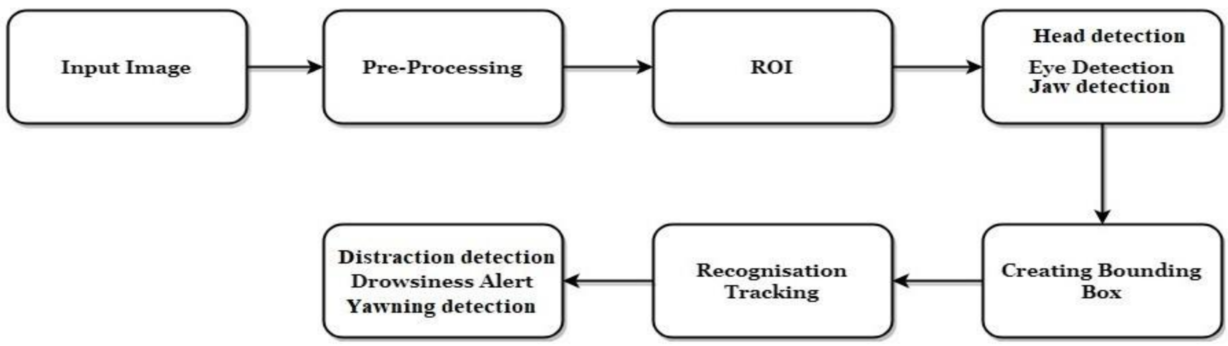


Fig3: General architecute of the model

To apply this generality, we're using following modules-

Video Recording: This module connects to the webcam using the built-in Video Capture function in OpenCV, enabling video stream processing.
Frame Extraction: This module captures frames from the webcam, processing each image frame by frame and converting them into a two-dimensional array.
Face Detection & Facial Landmark Detection: Using the SVM algorithm, this module detects faces in the images and extracts facial landmarks from each frame to analyze facial expressions.
Detection: This module identifies the eyes and mouth within the detected face.
Calculation: Using the Euclidean Distance formula, this module calculates the proximity of the eyes and mouth to detect actions like eye blinking or yawning. If the system detects continuous eye blinking for 20 frames and an open mouth indicative of yawning, it will trigger an alert to warn the driver.

4. 1 Face Detection Using OpenCV

While it may seem complex initially, face detection with OpenCV is actually quite simple.

Step 1: To begin, we first load an image. Next, we need to create a cascade classifier, which will help identify the features of the face.

Step 2: In this step, we use OpenCV to read the image and train it with the relevant feature data. At this point, the data is stored in NumPy arrays at the primary data points. Our task is to find the row and column values that correspond to the face within the NumPy N-dimensional array.

Step 3: The final step involves displaying the image with a bounding box around the detected face. There are various algorithms and techniques for eye tracking and monitoring, most of which focus on detecting specific eye features (often reflections from the eye) within a video stream of the driver.

4.2 Face Detection and Drowsiness Detection Using Retinal Reflection

The primary goal of this paper is to use retinal reflection to detect the eyes on a face, and then identify the absence of this reflection as an indication of closed eyes. By applying this algorithm to consecutive video frames, we can calculate the duration of eye closure, which is crucial for detecting drowsy drivers. The duration of eye closure in drowsy drivers is typically longer than that of normal blinking, and even a slight increase in the time can lead to serious accidents. Therefore, the system will immediately alert the driver when closed eyes are detected.

In addition to eye and head movements, another visual cue for detecting drowsiness is the analysis of facial features. Developing a real-time application using computer vision is a challenging but effective task that requires powerful processing capabilities. OpenCV, an open-source computer vision library, is commonly used for this purpose. OpenCV is available in C, C++, Python, and Java, and can operate on systems with clock speeds up to 1500 MHz, running on lightweight Linux-based Raspbian OS, which comes pre-loaded with programming software and OpenCV.

If the system detects that the driver is drowsy or fatigued, a warning message will be sent via Twilio, and a buzzer will be triggered. The Haar Feature-based Cascade Classifier, a machine learning approach, is employed for face and eye region detection. This method uses a classifier trained on a large set of positive and negative images to detect face and eye regions, updating the region of interest (ROI). OpenCV provides both a trainer and a detector for this task.

OpenCV is also used to create a user-defined object classifier, which is saved as an XML file for use in later stages of the program. Additionally, the Canny edge detection operator is utilized to precisely identify the coordinates of the eye region.

The system employs a camera to capture the driver's face, tracking the eyes and mouth. Eye tracking detects whether the driver's eyes are open or closed, measuring drowsiness levels, while mouth tracking checks for yawning. After detecting and tracking the face, eyes, and mouth, the system continuously monitors any changes in these parameters. Using Visual Studio 2013 and OpenCV with Emgu, facial feature tracking and detection are performed. OpenCV with Emgu includes XML files for eye closure detection and yawn detection, which are carried out using template matching, comparing stored images to determine the driver's state of alertness. If the driver is detected to be drowsy, an alarm is triggered to alert the driver. The alarm continues to sound within a short time interval.

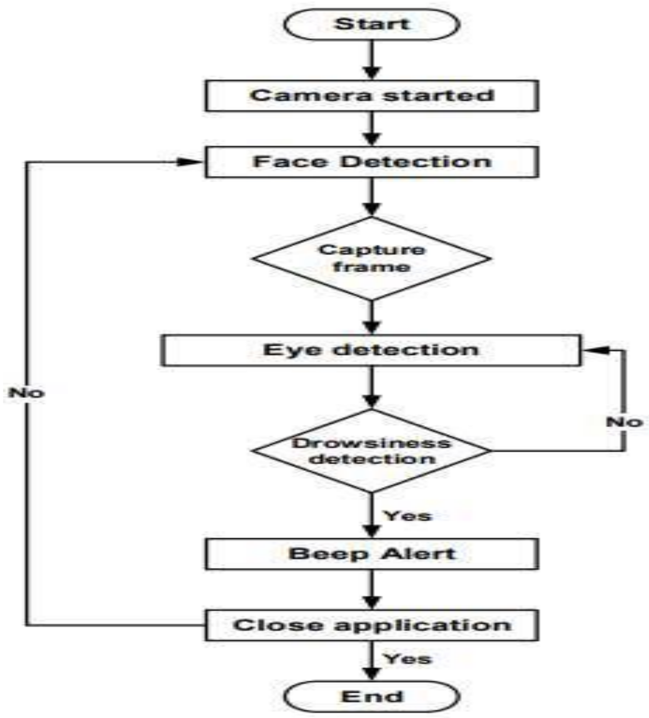


Fig 4: Eye detection process using camera

For sleepiness detection, Python is used in this paper. The system focuses on detecting the face as the key feature. A webcam is positioned in front of the driver's face to capture video input. The algorithm assumes that the driver is sleeping if no face is detected across several frames. OpenCV, using 68 facial landmarks, is employed to identify the face and eyes. The Euclidean Eye Aspect Ratio (EAR) [32][33] is used to determine whether the eyes are open or closed.

The system analyzes the driver's face and eyes, determining if the eyes are open or closed. If the eyes remain closed for a certain period, an alarm will sound to alert the driver. If the eyes open, the system will resume tracking.

Additionally, the system uses PERCLOS (Percentage of Eyelid Closure Over Time), which measures gradual eyelid closures as opposed to brief blinks. The system calculates the PERCLOS score, and based on this score, an alarm is triggered [27][28][29].

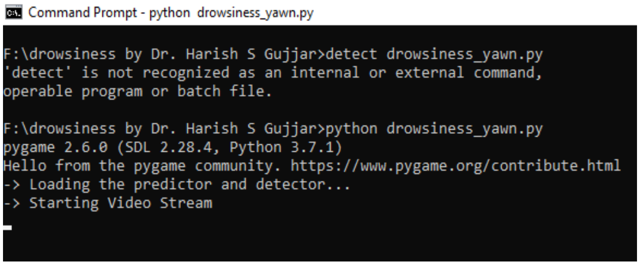
The following libraries are used in this paper:

- **CV2**: OpenCV, an open-source library for computer vision and image processing, is used to analyze images and videos, recognizing faces, objects, etc. [16][17][18].
- **OS**: Python's OS module is used for interacting with the operating system, providing a portable means of accessing system-specific features [19][20][21].

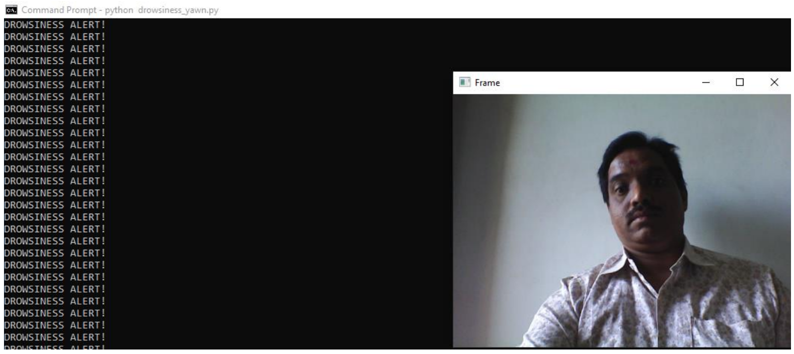
- **Keras**: A high-level Python neural network library that works with TensorFlow, Keras is widely used for its ease of use and compatibility with both CPUs and GPUs [22][23][24].
- **NumPy**: A Python library for working with arrays, matrices, and performing linear algebra operations.
- **Pygame**: A cross-platform Python library for creating video games, which includes graphics and sound libraries [25][26].
- **Matplotlib**: A Python library for creating visual graphs and plots [27][28].

4.3 Outcome Screenshots

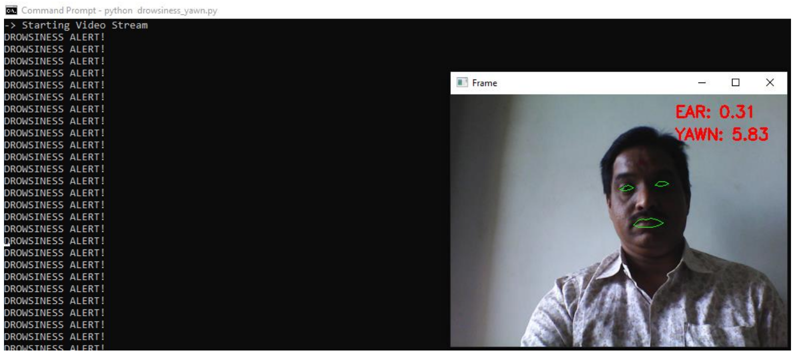
1. Upon running the program, the first screen is displayed.



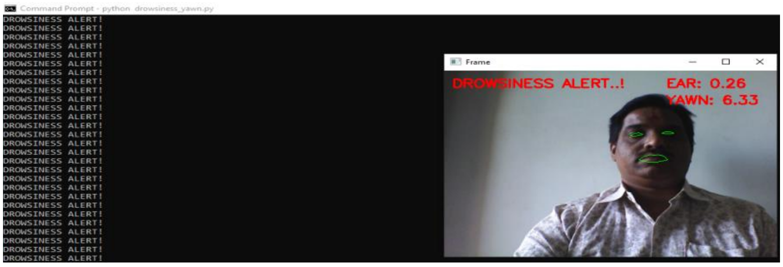
2. The camera opens, detects the face, and since the eyes are open, the score is set to zero. This is an example without glasses.



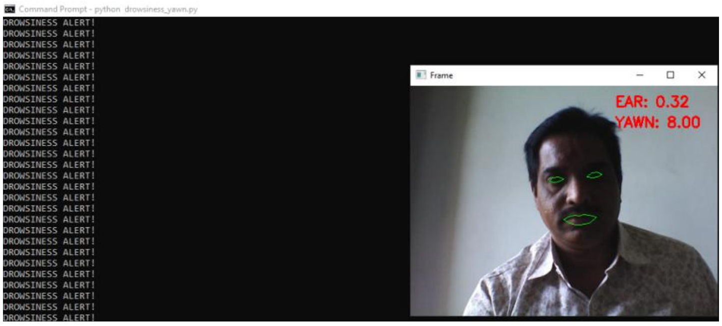
3. The eyes close, causing the score to increase.



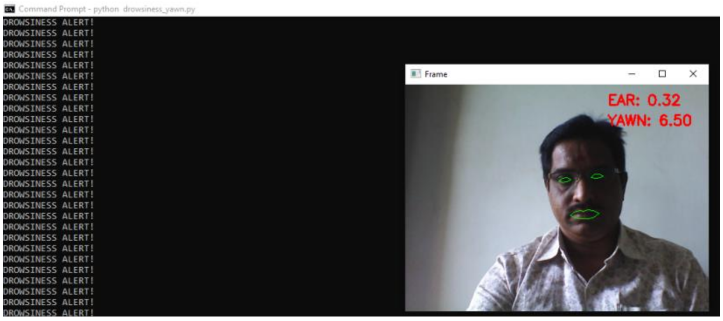
4. With the eyes closed and the score surpassing the set threshold of '10', the alarm beeps to alert the driver.



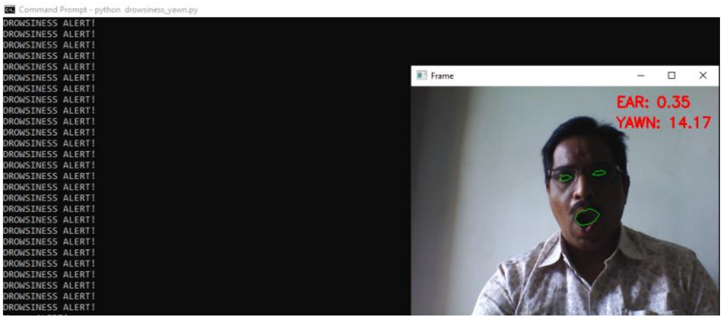
5. Once the eyes open again, the score decreases, and the alarm stops.



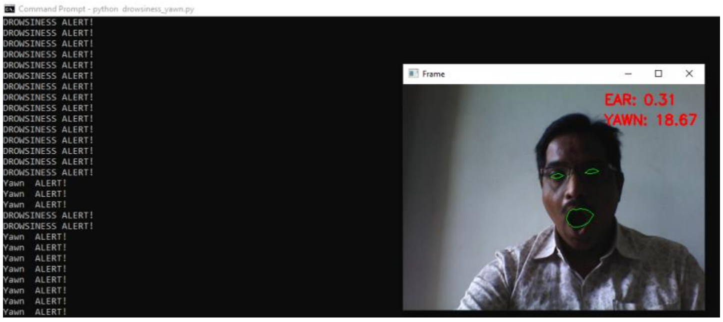
6. In this example, the system detects eyes with glasses. The eyes are open, so the score remains at '0'.



7. The eyes close, and the score increases, but it does not reach the threshold value, so the alarm does not beep.



8. With the score reaching '14', which exceeds the threshold, the alarm beeps to alert the driver [32][33].



9. When a keyboard interrupt occurs, the camera closes, and the program stops [29][30][31].

5. Result Analysis

The primary approach involves extracting image features from facial landmarks. Facial landmarks are typically considered a subset of the shape predictor problem, as they help localize specific areas of interest, such as the eyes, nose, and mouth, along with the overall shape of the face. The dlib library provides a facial landmark detector that identifies 68 (a, b) coordinates for accurate face mapping.

Individual	**Ear Threshold**	**Alarm Sensitivity**	**Light**	**Remarks**	**Drowsiness Detection Alarm**
A	0.2	48	Bright	Normal	3 out of 3
A	0.2	48	Dim	Normal	3 out of 3
A	0.2	48	Bright	Wear sun glasses	3 out of 3
B	0.25	43	Bright	Normal	3 out of 3
B	0.25	43	Dim	Normal	3 out of 3
B	0.25	43	Dim	Rainy Weather	2 out of 3
C	0.22	48	Bright	Wear sun glasses	3 out of 3
C	0.22	48	Dim	Wear sun glasses	3 out of 3
C	0.22	48	Very Dim	Night drive	1 out of 3
C	0.22	48	Very Dim	Normal	3 out of 3

Table 1: The table above outlines the testing parameters used to evaluate accuracy

The entire test was conducted 10 times with varying parameters such as lighting conditions, different drivers, and alarm sensitivity. The table above outlines the testing parameters used to evaluate accuracy. The tests aimed to assess the overall performance of the project, using the accuracy formula: CR = (C/A) × 100%, where CR represents the correct rate, C is the number of successful tests, and A is the total number of tests. Out of 10 tests, 8 were successful, while 2 failed due to poor lighting conditions at night. As a result, the overall accuracy of the project was approximately 84%. The lighting conditions had a significant impact on the accuracy and output of the system, with brightness being a key factor. Despite this, our average experimental accuracy stands at 94%.

6. Conclusion

In conclusion, the driver drowsiness detection system is an important vehicle safety technology designed to help prevent injuries caused by drowsy driving. Early detection and alerting the driver before an accident occurs can significantly reduce the risk of life-threatening incidents. The proposed system detects drowsiness by analyzing the Eye Aspect Ratio (EAR), which measures the size of the driver's eyes. By gathering data on the EAR, a threshold can be established to identify when a driver is showing signs of drowsiness. The system's alert feature, triggered by an alarm, is crucial for reducing the number of accidents and injuries caused by drowsy driving, ultimately decreasing the number of crashes annually.

Currently, the detection system effectively identifies drowsiness in the same driver with minimal limitations. The alarm works properly, triggering a valid alert when necessary. However, the threshold for triggering the alarm may vary due to different EAR values in individuals. Future improvements include enabling the system to automatically determine the threshold for drowsiness detection, eliminating the need for manual calibration for each individual. This enhancement would account for varying levels of alertness and sensitivity, as some drivers may require a more frequent or sensitive alarm system due to their higher awareness and precaution toward road safety.

7. Authors' Biography

Dr. Harish S Gujjar MCA, M.Phil, Ph.D is presently an Associate Professor and Head in department of computer science, SSAS Government First Grade College, Hosapete, Karnataka, India. He did his M.C.A (Master of Computer Application) from Vishveswaraiah Technological University, Belgaum, Karnataka, India. M. Phil (Computer Science) from Allagappa university, Karaikudi, Tamilnadu, India. Ph. D in Computer Science from BU, Coimbatore, Tamil Nadu, India He has published many research papers in national and international journals and presented several research papers in national and international conferences. He has delivered many lectures on current topics at various institutions and UGC HRDCs. He is a editorial board member and reviewer for various international journals and publications. He also a member for various international organizations. He has published many books in peered publications. He is also a BOE and BOS member of many Universities and Autonomous Colleges.

8. References

1. Ministry of Road Transport and Highways, Government of India Report on accidents in India every year. Available at: https://morth.nic.in/road-accident-in-india.
2. Vesselenyi, T., et al. "2017 IOP Conf. Ser.: Mater. Sci. Eng." 252 012097.
3. "Driver Drowsiness Detection and Alert System." ResearchGate. Available at: https://www.researchgate .net/publication/352116070_DriverDrowsinessDetection_and_Alert\System_using_Python_&_OpenCV.
4. "Drowsy Driving Statistics." Available at: https://www.thezebra.com/resources/research/drowsy-driving-statistics/.

5. National Highway Traffic Safety Administration (NHTSA). "Drowsy Driving Data." Available at: https://www.nhtsa.gov/risky-driving/drowsy-driving.
6. Krajewski, J., Sommer, D., Trutschel, U., Edwards, D., & Golz, M. "Steering wheel behavior-based estimation of fatigue." The Fifth International Driving Symposium on Human Factors in Driver Assessment, Training, and Vehicle Design, 2009; pp. 118-124.
7. Fagerberg, K. "Vehicle-based detection of inattentive driving for integration in an adaptive lane departure warning system," M.S. thesis, KTH Signals Sensors and Systems, Stockholm, Sweden, 2004.
8. "Driver Alert Control (DAC)" (2018, January 08). at: http://support.volvocars.com/uk/cars/ Pages/ owners manual.aspx?mc=Y555&my=2015&sw=14w20&article=2e82f6fc0d1139c2c0a801e800329d4e.
9. Mardi, Z., Ashtiani, S. N., & Mikaili, M. "EEG-based drowsiness detection for safe driving using chaotic features and statistical tests." Journal of Medical Signals and Sensors, 2011; 1:130-137.
10. Danisman, T., Bilasco, I. M., Djeraba, C., & Ihaddadene, N. "Drowsy driver detection system using eye blink patterns." Machine and Web Intelligence (ICMWI) IEEE, 2010; pp. 230-233.
11. Hariri, B., Abtahi, S., Shirmohammadi, S., & Martel, L. "A yawning measurement method to detect driver drowsiness." Technical Papers, 2012.
12. Abtahi, S., Hariri, B., & Shirmohammadi, S. "Driver drowsiness monitoring based on yawning detection." 2011 IEEE International Instrumentation and Measurement Technology Conference. doi:10.1109/imtc.2011.5944101.
13. Norhadi, A., Marzuki, A., Wicaksono, L., & Addetya Yacob, R. "Studi Debit Aliran pada Sungai Antasan Kelurahansungai Andai Banjarmasin Utara," J. Poros Tek., vol. 7, no. 1, pp. 1–53, 2015.
14. Sianipar, A. F., & Lutfiani, N. "Sistem Parkir Berbasis QR Code Pada Perguruan Tinggi Raharja." Semin. Nas. Multi Disiplin Ilmu, vol. 1, no. November 2017, pp. 43–54.
15. Narayan, V., & Daniel, A. K. "Multi-tier cluster based smart farming using wireless sensor network." 2020 5th International Conference on Computing, Communication, and Security (ICCCS). IEEE, 2020.
16. Narayan, Vipul, and A. K. Daniel. "Energy Efficient Protocol for Lifetime Prediction of Wireless Sensor Network using Multivariate Polynomial Regression Model." Journal of Scientific & Industrial Research 81.12 (2022): 1297-1309.
17. Tyagi, Lalit Kumar, et al. "Energy Efficient Routing Protocol Using Next Cluster Head Selection Process In Two-Level Hierarchy For Wireless Sensor Network." Journal of Pharmaceutical Negative Results (2023): 665-676.
18. Narayan, Vipul, et al. "E-Commerce recommendation method based on collaborative filtering technology." International Journal of Current Engineering and Technology 7.3 (2017): 974-982.
19. Vipul, Narayan, and A. K. Daniel. "A novel protocol for detection and optimization of overlapping coverage in wireless sensor network." International Journal of Engineering and Advanced Technology (2019): 422-462.
20. Narayan, Vipul, et al. "To Implement a Web Page using Thread in Java." (2017).
21. Srivastava, Swapnita, and P. K. Singh. "HCIP: Hybrid Short Long History Table-based Cache Instruction Prefetcher." International Journal of Next-Generation Computing 13.3 (2022).
22. Srivastava, Swapnita, and P. K. Singh. "Proof of Optimality based on Greedy Algorithm for Offline Cache Replacement Algorithm." International Journal of Next-Generation Computing 13.3 (2022).

23. Srivastava, Swapnita, and P. K. Singh. "Proof of Optimality based on Greedy Algorithm for Offline Cache Replacement Algorithm." International Journal of Next-Generation Computing 13.3 (2022).
24. Smiti, Puja, Swapnita Srivastava, and Nitin Rakesh. "Video and audio streaming issues in multimedia application." 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence). IEEE, 2018.
25. Smriti, Puja, Swapnita Srivastava, and Saurabh Singh. "Keyboard invariant biometric authentication." 2018 4th International Conference on Computational Intelligence & Communication Technology (CICT). IEEE, 2018.
26. Dua, M., Shakshi, Singla, R., Raj, S., & Jangra, A. (2020). Deep CNN models-based ensemble approach to driver drowsiness detection. Neural Computing and Applications. doi:10.1007/s00521-020-052097.
27. P. K. Mall and P. K. Singh, "Explainable Deep Learning approach for Shoulder Abnormality Detection in X-Rays Dataset.," Int. J. Next-Generation Comput., vol. 13, no. 3, 2022.
28. P. K. Mall and P. K. Singh, "BoostNet: a method to enhance the performance of deep learning model on musculoskeletal radiographs X-ray images," Int. J. Syst. Assur. Eng. Manag., pp. 1–15, 2022.
29. P. K. Mall, P. K. Singh, and D. Yadav, "GLCM based feature extraction and medical X-RAY image classification using machine learning techniques," in 2019 IEEE Conference on Information and Communication Technology, 2019, pp. 1–6.
30. P. K. Mall and P. Singh, "Credence-Net: a semi-supervised deep learning approach for medical images," Int. J. Nanotechnol., vol. 20, 2022.
31. Irfan, D., Tang, X., Narayan, V., Mall, P. K., Srivastava, S., & Saravanan, V. (2022). Prediction of Quality Food Sale in Mart Using the AI-Based TOR Method. Journal of Food Quality, 2022.
32. Mall, P. K., Narayan, V., Pramanik, S., Srivastava, S., Faiz, M., Sriramulu, S., & Kumar, M. N. (2023). FuzzyNet-Based Modelling Smart Traffic System in Smart Cities Using Deep Learning Models. In Handbook of Research on Data-Driven Mathematical Modeling in Smart Cities (pp. 76-95). IGI Global.
33. Mall, P. K., Yadav, R. K., Rai, A. K., Narayan, V., & Srivastava, S. (2023). Early Warning Signs of Parkinson's Disease Prediction Using Machine Learning Technique. Journal of Pharmaceutical Negative Results, 2607-2615.
34. Jabbar, R., Al-Khalifa, K., Kharbeche, M., Alhajyaseen, W., Jafari, M., Jiang, S. "Real-time Driver Drowsiness Detection for Android Application Using Deep Neural Networks Techniques." Procedia Computer Science, 130 (2018): 400-407. ISSN 1877-0509. https://doi.org/10.1016/j.procs.2018.04.060.