AJEMA GLOBAL RESEARCH NETWORK

AMERICAN Journal of Engineering, Mechanics and Architecture
Volume 4, Issue 01, 2026 ISSN (E): 2993-2637

A Real-Time Deep Learning–Based Drowsiness Detection and Alert System for Enhanced Automotive Safety

S. Manimaran [1]
[1] Department of Computer Science Engineering, Dhaanish Ahmed College Of Engineering, Padappai, Chennai, Tamil Nadu, India., manimaran@dhaanishcollege.in

S.R. Saranya [2]
[2] Department of Computer Science Engineering, Dhaanish Ahmed College of Engineering, Padappai, Chennai, Tamil Nadu, India

M. Mohamed Thariq [3]
[3] Department of Computer Science and Engineering, Dhaanish Ahmed College of Engineering, Chennai, Tamil Nadu, India.

M. Mohamed Sameer Ali [4]
[4] Dhaanish Ahmed College of Engineering, Chennai, Tamil Nadu, India.

S. Suman Rajest [5]
[5] Dhaanish Ahmed College of Engineering, Chennai, Tamil Nadu, India

Abstract: The Comprehensive Drowsiness Detection and Alert Solution for Automotive Safety addresses the growing concern of road accidents caused by driver weariness, which is a factor in about 20% of incidents worldwide. Keeping drivers awake is very important for reducing dangers and saving lives. This research proposes a robust, real-time monitoring system that leverages computer vision and deep learning to detect early signs of fatigue and prevent accidents. The system uses a live camera feed to monitor the driver's face, paying close attention to how their eyes move and blink. A Convolutional Neural Network (CNN) analyses this data to detect long periods of eye closure, a sign of exhaustion. The system sounds an alarm as soon as it sees that the driver is sleepy. This non-intrusive method ensures continuous monitoring without causing discomfort or disruptions. The project is designed to be flexible and scalable so that it can be used as a standalone device in cars or as a mobile app. It makes sure that a wide variety of users can pay and access it by using inexpensive hardware and open-source software. The system can do more than just the basic tasks. It can also be integrated with additional functions, such as heart rate monitoring, vehicle speed control, and yawning recognition, making it much more useful and reliable. This approach can be used not only in personal cars, but also in commercial fleets, public transportation networks, and heavy industrial operations. The bigger effect on society is that it lowers the emotional and financial costs of car accidents, encourages safer driving, and generally makes traffic safer.

Keywords: Driver Drowsiness Detection; Automotive Safety; Convolutional Neural Network (CNN); Non-Intrusive Approach; Yawning Detection.

Introduction

Driver drowsiness detection is an important safety feature for cars that helps prevent drivers from falling asleep, losing attention, or becoming drowsy. Modern transportation systems have made vehicles faster, more comfortable, and easier to reach, but these improvements have also made long drives and mental fatigue more dangerous [21]. Accidents caused by tiredness are still one of the top causes of serious injuries and deaths around the world. A lot of research shows that about one in five road accidents is caused by drivers falling asleep at the wheel. On some roads and long-distance routes, the number can reach 50%. These scary numbers show how important it is to have smart technologies that can monitor a driver's alertness and step in before a catastrophic situation occurs [31]. Drowsiness is a slow change in the body that makes it harder to react, make decisions, and coordinate movements. Drivers can't tell how vigilant they are, since weariness builds gradually, unlike abrupt distractions [41]. Driver fatigue can be caused by factors such as long work hours, driving at night, insufficient sleep, monotonous road conditions, and health problems [26]. When drivers get sleepy, they may blink a lot, close their eyes for long periods, nod, and have trouble staying in their lane.

In very bad situations, microsleep episodes lasting only a few seconds can cause terrible accidents, especially while going fast [45]. To fix this problem, we need a system that can detect subtle changes in behaviour in real time and respond quickly. Driver drowsiness detection systems look for signs of tiredness by monitoring both physical and behavioural indicators. Researchers have suggested many other approaches, but vision-based solutions have become very popular because they are not obtrusive and highly reliable [30]. Most of the time, these systems use cameras to monitor facial features, including how often the eyes move, how long the eyelids stay closed, where the head is, and how often the person yawns. The system can determine how alert the driver is by analysing these visual clues. It doesn't need to wear sensors or physical contact so it can be used in real cars [50]. The technology in this project is primarily focused on eye detection, which is considered one of the best ways to detect sleepiness. The eyes are very important for gauging a driver's level of awareness, because keeping them closed for a long time is a major symptom of drowsiness [37]. The technology keeps an eye on the driver's eyes to see if they are open or closed, and tracks how long they remain closed. The system calls the state "drowsy" and issues an alert if the eyes remain closed for more than a certain amount of time, usually more than 2 seconds. This quick action can prevent accidents by waking the driver or safely stopping the car. One of the biggest problems in detecting drowsy drivers is ensuring it works well across different real-world situations. Drivers might wear glasses, sunglasses, or contact lenses, and the light inside the car can change significantly depending on the time of day, the weather, or when the car enters a tunnel [22]. Driving at night is harder because of the poor light and glare from headlights. While still being able to accurately identify, the system needs to be strong enough to handle these changes [42]. This project seeks to solve these problems and deliver consistent performance across a variety of situations by leveraging deep learning and powerful computer vision methods [51].

Deep learning has transformed the field of computer vision by enabling machines to learn useful features from raw data on their own. This project uses a deep learning architecture to detect faces and determine whether eyes are open or closed [19]. The model learns from many eye photos, which helps it identify subtle patterns associated with different eye disorders. Deep learning models are better suited for real-time driver monitoring systems because they can adapt to variations in facial features, head positions, and lighting conditions. Traditional rule-based methods couldn't do this [36]. The technology works by capturing live video frames from a webcam inside the car. The system processes each frame to find the driver's face and the areas around their eyes. The deep learning model examines the eyeballs once it finds them and determines whether they are open or closed [29]. The system keeps track of how many frames in a row the eyes are classed as closed. If this count exceeds a limit corresponding to a time period of more than two seconds, the system knows the driver is sleepy and sends a warning [46]. This response includes turning on the parking lights and activating an auditory alarm, which can help slow or stop the car and reduce the risk of accidents.

The impetus for this endeavour stems from the recognition that current driver sleepiness detection methods are not consistently precise or reliable [32]. There are many commercial products and research prototypes, but many of them suffer from false positives and false negatives. Drivers can get annoyed by false positives, which cause unnecessary warnings, and by false negatives, which can miss real tiredness and be dangerous. These constraints show that we need better ways to find problems and new ways of looking at things that can make the system work better [43]. This study adds to this ongoing research effort by implementing and testing a deep learning-based method that prioritises robustness, accuracy, and ease of use. Python is a popular programming language known for its ease of use and extensive library ecosystem [23]. This system is built using Python. Python is a great fit for this project because it offers strong capabilities in computer vision, machine learning, and graphical user interface development. The system has many libraries that each have a distinct role in the overall architecture [40]. These parts work together to record video, interpret photos, perform deep learning, and provide the driver with real-time feedback.

For fast math and array operations, NumPy is employed [47]. It is very important for processing picture data and doing math, such as figuring out how far apart face features are. These calculations are necessary to correctly identify facial features and track eye movements over time. For activities such as video capture and image processing, OpenCV is used. It lets the system use the webcam, read video frames in real time, convert images to black and white, and apply preprocessing methods to improve detection accuracy [25]. The system runs smoothly without major delays thanks to OpenCV's efficient implementation. Dlib is another important library utilised in this research. It has strong features for finding faces and facial landmarks [54]. Dlib's pre-trained models can detect faces and distinguish crucial features such as the eyes, nose, and mouth. The system can accurately detect eye areas in each video frame, even when the driver's head moves slightly [35]. For reliable eye condition classification and drowsiness detection, this level of precision is very important.

Pillow, or PIL, is a Python library that converts image formats and processes images. It lets the system convert OpenCV-processed video frames into formats that can be displayed in a graphical user interface [39]. This integration lets users see the video feed in real time, providing fast feedback and improving the overall experience. When debugging, testing, or demonstrating something, visual information must be easy to read [18]. When tiredness is detected, PlaySound makes sounds that can be heard. Sound-based notifications are quite effective at getting the driver's attention, especially when the driver is in a microsleep state from fatigue [52]. The technology ensures the driver is promptly warned of a risky situation and told to take action by playing a buzzer or alert sound. The alert system is meant to be easy to use but still work, so the driver is less likely to ignore it [27].

We use Tkinter, Python's built-in GUI toolkit, to create a full-screen interface that displays the live video stream and the system's condition [33]. The GUI is easy to use and shows whether the driver's eyes are open or closed and the current alert condition. A simple, easy-to-use interface is necessary for practical use, as it lets drivers and developers quickly see how the system works [53]. Threading allows several tasks to run simultaneously in a system. In one instance, it lets the alarm sound play without the video processing loop. If you don't use threading, playing the sound could cause the main program to stop and prevent the video from updating in real time [24]. The system keeps running smoothly and monitors things, sending timely alerts by running the buzzer in a separate thread. This project is important not only for how it works technically, but also for showing how important it is to use technology to promote safer driving [48]. Driver fatigue is a preventable cause of road accidents that affects millions of families worldwide. You can save lives, reduce injuries, and lower economic losses by developing smart systems that spot drowsiness early and take action [38]. Adding deep learning and computer vision to car safety systems is a major step toward smarter, more responsive transportation infrastructure.

This project is also a great way to learn about how to use artificial intelligence in real-life situations [34]. It shows how theoretical ideas such as neural networks, image processing, and real-time systems can work together to solve important societal problems [28]. The system's modular design means it can be improved in the future by adding more behavioural cues, such as yaw detection, head attitude estimation, or heart rate monitoring. It can also be expanded to handle multiple drivers, connect with vehicle control systems, or run on embedded platforms for commercial use [44]. In conclusion, detecting whether a driver is sleepy is an important area of research and development in modern car safety technology. This research presents a comprehensive approach to using deep learning and computer vision to detect fatigue [20]. The system's goal is to reduce the likelihood of accidents caused by tiredness and encourage safer driving behaviours by monitoring eye behaviour in real time and sending appropriate alerts. As transportation systems evolve, smart monitoring solutions like this will become increasingly crucial for keeping people safe on the road and saving lives [49].

Literature Survey

Introduction

We conducted this poll to learn what the general public needs and wants [10]. To do this, we looked for the basic information on different websites and apps [17]. There is some research on Driver Drowsiness Detection to determine how to use it and achieve the right results. The studies employ several methodologies for the application and requirement processes [3]. System for Detecting Driver Drowsiness and Sending Alerts (5 May 2023, Suhana Nafais A). This study aims to develop a system that uses powerful neural networks to identify sleepy drivers and send alerts to make roads safer. YOLOv3 is a pretrained CNN that performs well at detecting objects, especially facial features such as the eyes and mouth [8]. Driver's Level of Drowsiness (17 February 2021, Kyong Hee Lee, Whui Kim, Hyun Kyun Choi, Byung Tae Jang). This study examines techniques for identifying driver tiredness using facial indicators, including eye closure, yawning, and head orientation [6]. It uses the OpenCV and Dlib libraries to extract extensive information about facial behaviour over time and space to determine how tired a driver is [15]. Drowsiness Detection Based on Driver Temporal Behaviour (31 March 2021, F. Faraji, F. Lotfi, J. Khorramdel, A. Najafi, A. Ghaffari). This study employs YOLOv3 CNN as a pretrained network, demonstrating its efficacy as a robust tool for object detection [12].

A Survey on State-of-the-Art Driver Drowsiness Detection Techniques (December 1, 2020, FHikmat Ullah Khan). The detection method involves steps such as obtaining the face image, checking whether the person is yawning, identifying the area around the eyes, and so on [2]. It looks for signs like eye blinks, head movements, and yawning to determine whether someone is sleepy. A convolutional neural network (2020, Anjith George, Aurobinda Routray) can tell you where your eyes are looking in real time [14]. This research introduces a real-time eye-gaze categorisation system using convolutional neural networks (CNNs). It figures out where someone is looking to figure out their cognitive state and attentiveness, and it works well at 24 frames per second. Machine Learning and Gradient Statistics-Based Real-Time Driver Drowsiness Detection (2022, Cyun-Yi Lin, Paul Chang, Alan Wang, Chih-Peng Fan). This work presents a machine learning-based approach for real-time drowsiness detection [5]. Using gradient statistics and machine learning, it can find faces, eyes, and monitor when eyes close. Camera-based Drowsiness Reference for Driver State Classification under Real-Driving Conditions (1-december 2020, Fabian Friedrichs and Bin Yang). The Intelligent Vehicles Symposium examines how eye-tracking technology can be used to monitor driver fatigue [13]. It uses 90 hours of driving data to examine blink duration, frequency, and eye closure percentage.

Driver Drowsiness Monitoring System Utilising Visual Behaviour and Machine Learning (2021, Ashish Kumar, Rusha Patra). This paper presents a real-time driver sleepiness detection system that utilises a webcam to capture facial landmarks and compute metrics such as the Eye Aspect Ratio (EAR), Mouth Opening Ratio (MOR), and Nose Length Ratio (NLR). Driver Drowsiness Monitoring System Utilising Visual Behaviour and Machine Learning (2021, Ashish Kumar, Rusha Patra). This paper presents a real-time driver sleepiness detection system that utilises a webcam to capture facial landmarks and calculate metrics such as Eye Aspect Ratio (EAR), Mouth Opening Ratio (MOR), and Nose Length Ratio (NLR) [1]. A Survey on Driver Drowsiness Detection Techniques (December 1, 2020, by Reshma, Ishwarya, and Sai Vennala). The Drowsiness Detection System incorporates techniques such as face extraction and yawning propensity assessment [9]. Driver Drowsiness Detection Utilising Behavioural Metrics and Machine Learning Approaches: An Examination of Cutting-Edge Methodologies (2022, Mkhuseli Ngxande, Jules-Raymond Tapamo, Michael Burke). Approaches for determining whether a driver is sleepy use behavioural measures and machine learning techniques. It looks at things like yawning, blinking, and head movements to figure out whether you're sleepy [7].

Summary

In this Python project, we built a system that warns you if you're sleepy while driving [16]. You can use it in many different ways. We used a Haar cascade classifier in OpenCV to detect faces and eyes, and then a CNN to predict the state [4]. The system uses a mix of deep learning models and computer vision techniques to monitor the driver's face and eyes in real time [11]. If it sees indicators that the driver is getting sleepy, it will send an alert.

Project Description

Vision-Based Techniques

No eye detection is the most important indicator of sleepiness [60]. Yawning and nodding don't always make sense. It depends on the person; some people may not yawn when they are tired [69].

Physiological Sensors

- Better answers need to be connected to the body
- What if the motorist forgets to wear it?
- People may be hesitant to use easy-to-use biosensors.
- Headbands and headphones (Signal-Channel EEG)
- Portal glasses for EOG
- Hand bands for PPG.

Proposed System and Advantages: The proposed system would use a live camera to continuously monitor the driver's eye movement. All monitored signals will be preprocessed [59]. To get around problems, Python is used because it already has a training system installed, which saves time spent starting from scratch [73]. To improve performance and eliminate common problems with these kinds of systems, such as delays, it creates a very efficient and flexible environment [64].

System Design: We built the model using Keras and Convolutional Neural Networks (CNNs) [76]. A convolutional neural network is a sort of deep neural network that works very well at classifying images [61]. A CNN has an input layer, an output layer, and a hidden layer. The hidden layer might have multiple layers [68].

System Architecture

The CNN model architecture consists of the following layers:

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

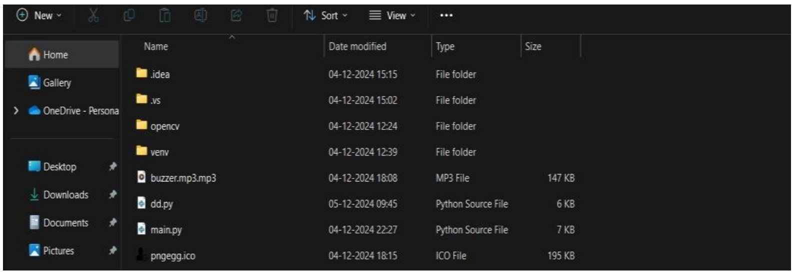

Figure 1: The contents of the zip

The haar cascade files folder contains the XML files needed to detect objects in the picture [58]. In our example, we are finding the person's face and eyes. The model file cnnCat2.h5, which we trained using convolutional neural networks, is in the models folder (Figure 1) [72].

Block diagram

The block diagram shows how a system for detecting tiredness works [67]. At first, the system uses computer vision to detect the driver's face by capturing live video. After finding the face, the algorithm searches for the ocular region to obtain additional information about the driver's eyes [55]. It also looks at the pupil area and estimates where the eyes are, so it can tell for sure whether the eyes are open or closed. We use this information to assess the driver's condition by examining their eye state [63]. If the system detects that the driver is sleepy, it emits a warning, such as a buzzer, to alert the driver [70].

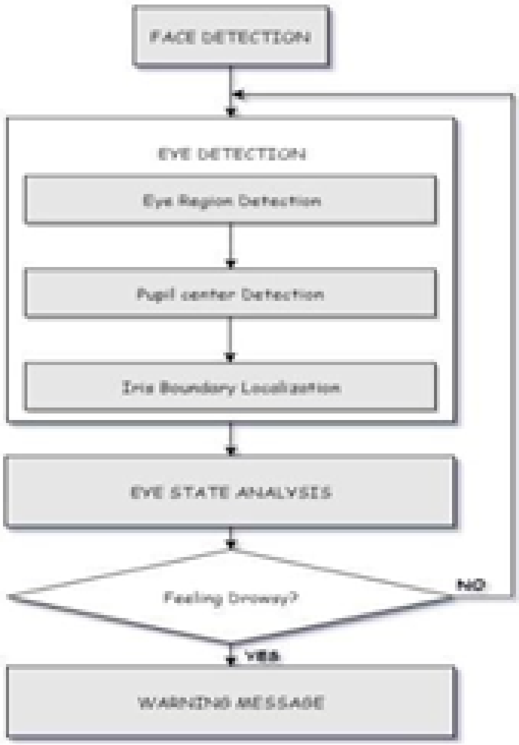

Figure 2: Drowsiness Detection System Workflow

The system keeps an eye on things if it doesn't see any drowsiness [75]. A black box with software controls this procedure in real time. It keeps an eye on things all the time and helps stop mishaps caused by tiredness [62]. Then, the system uses this information about the driver's eye state to determine whether they are sleepy (Figure 2).

Use Case Diagram

The use case diagram gives a clear picture of how the sleepiness detection system works and how it interacts with the driver [65]. The use case diagram shows in great detail how the sleepiness detection system operates and interacts with the driver. The driver starts the process by turning on the camera [71]. The camera records video in real time, and advanced facial recognition techniques are used to detect the driver's face and eyes [56]. After this, the system looks at the eyes and determines whether they are open or closed over specific periods of time.

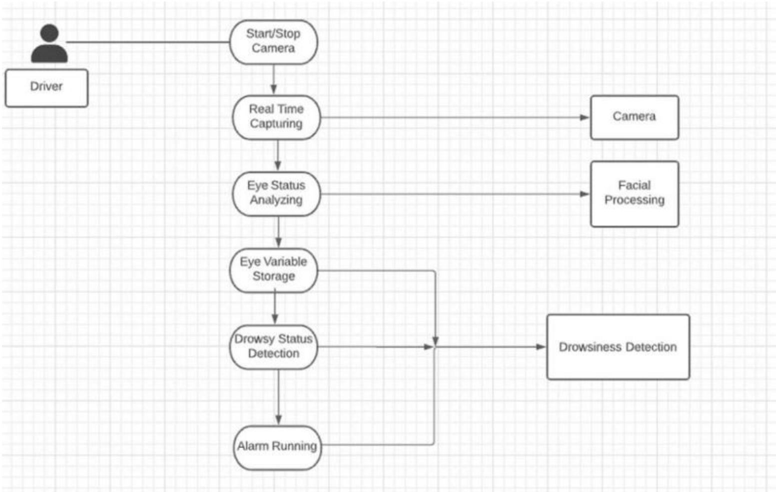

Figure 3: Use Case Diagram for Drowsiness Detection

These observations are kept as eye variables, which are very important for figuring out how aware the driver is [66]. Then the saved information is used to determine how sleepy the driver is [74]. The technology classifies the driver as tired and sets off an alarm to notify them right away if the analysis detects prolonged eye closure or other signs of drowsiness [57]. This warning is a safety feature that tells the driver to pay attention and stay safe (Figure 3).

Project Requirements

Software Requirements

The whole application is developed on Python, a powerful and flexible programming language [100]. It uses a few important libraries to get its work done. NumPy is used to perform mathematical operations and work with data arrays quickly and easily [91]. You need OpenCV, an open-source computer vision library, to process images and videos, such as finding faces and eyes. Dlib has powerful machine learning techniques for recognising facial features that help more accurately detect signs of tiredness [105]. Pillow (PIP) is used to change images, while Imutils makes hard image processing chores easier.

Hardware Requirements

The app needs a laptop with basic hardware specs to run well [95]. A webcam is also necessary to get a live video of the driver. The webcam is the input device that sends live video to the program, which then uses it to find face and eye movements [89]. These hardware components work together to enable the system and ensure that drowsiness can be detected in real time.

Modules Description

There are two main parts to the sleepiness detection system: Webcam Capturing with Face and Eye Detection and sleepiness Detection with Alert Mechanisms [81]. The camera is the main input device; it records live video of the driver.

Eye Detection

We will use a webcam to take pictures. We developed an infinite loop to capture each frame so we could use the webcam [82]. We use the cv2 module that OpenCV provides. To get to the camera and set the capture object (cap), use VideoCapture(0). We put the image in a frame variable, and cap.read() reads each frame [106]. The camera comes on and starts taking the picture. The report includes an image of a closed eye to illustrate how this process works. This is an important state for determining whether someone is sleepy [76]. Researchers examine this closed-eye picture to determine whether the driver's eyes remain closed for a long time. The technology can gauge the driver's level of awareness by tracking whether their eyes are open or closed [88]. This picture shows how important eye detection is for detecting early indicators of tiredness, which is why it is a key part of the drowsiness detection system.

Haar Cascade Approach

Face and eye detection

- The "haar cascade files" folder contains the XML files that machine learning needs to detect objects in the picture. We are finding the person's face and eyes in this scenario [92]. We need to convert the image to grayscale first so the OpenCV object detection algorithm can work with it.
- We don't need to know the colour of things to find them. We will use the Haar cascade classifier to find faces. We utilise this line to build up our classifier.
- The Haar cascade method
- The cascades are essentially a bunch of XML files that contain OpenCV data used to find things [107]. You set up your code with the cascade you desire, and then it does the rest.
- It loads the face cascade into memory, reads the image, and turns it into grayscale [99]. (Most operations are done in grayscale because we don't need to define the colours green, Red, and Blue here.)

Eye Aspect Ratio

The system's main job is to detect drowsiness. It does this by analysing data collected in previous modules to assess the driver's alertness [93]. This module considers the state of the eyes, whether they are open or closed, as an important factor in determining how sleepy a person is. It figures out things like the Eye Aspect Ratio (EAR) and the length of time the eyes are closed by tracking how the driver looks over time [83]. The system considers this a symptom of tiredness if the eyes remain closed for a long time beyond a certain point. As soon as drowsiness is detected, the module activates its alert systems [98].

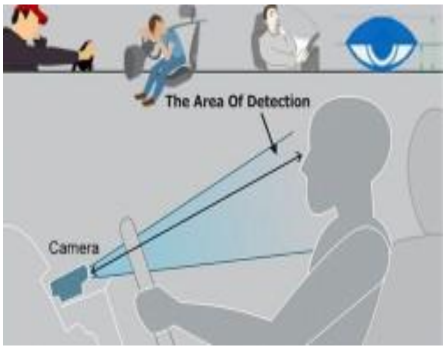

Figure 4: Alert Mechanism

A warning message appears on the screen to let the motorist know, and a loud alarm sounds at the same time [87]. This dual-alert system ensures the driver wakes up quickly and can get back to paying attention to the road. This module's goal is to make roads safer by reducing the risk of accidents caused by fatigue [77]. This module is very important for keeping people safe on the road and preventing harmful situations, as it can detect problems in real time and provide feedback immediately (Figure 4).

Implementation

The sleepiness detection technology is meant to make roads safer by monitoring the driver's behaviour in real time [94]. The main focus is on how the eyes move and how the face is turned, using computer vision and machine learning. The goal of this technology is to reduce accidents caused by tired drivers by sending timely alerts to lower the risk of fatigue-related crashes [84]. The first step in the sleepiness detection system is to use a camera to take pictures in real time. After the image is captured, the algorithm finds the driver's face and isolates a Region of Interest (ROI) around the eyes. The identified eye area is then analysed and sent to a pre-trained classifier, which determines whether the eyes are open or closed [104]. The system uses the classifier's output to determine how vigilant the driver is, assigning them a score. The system recognises the user as tired and sends appropriate alerts when the score exceeds a certain threshold, indicating that their eyes have been closed for a long time [78]. First, we used a camera mounted in a good spot in a car to detect faces. If a face is found, the facial landmark detection task is used to find the area around the eyes. We calculate the Eye Aspect Ratio from the eye area to determine whether the eyelids are down for a long time [96].

Step-By-Step Working Process

- Setting up the camera: A high-resolution camera is placed in the car and aimed at the driver's face. It sends footage to the processing unit in real time.
- Face Detection: The technology uses the video feed to find the driver's face [101]. A Region of Interest (ROI) is set up around the face that was found to reduce.
- Eye Detection: The system's main job is to find the eye area within the ROI. Using strong facial landmarking algorithms from Dlib improves detection accuracy.
- Eye Classification: A classifier looks at the eye area and decides whether the eyes are open or closed. This classification is based on the EAR, or Eye Aspect Ratio.
- Drowsiness Calculation: A score is given based on how long the eyes stay closed [85]. If the score is higher than a certain level, it means the person is sleepy.
- Alert Mechanism: If the driver's eyes remain closed for too long, the system sounds an alarm to wake them and prevent accidents.

The sleepiness detection system uses several key technologies to ensure it works. OpenCV is used to process images and videos in real time, and it is very important for finding faces and eyes [97]. Dlib is used for precise face landmarking, enabling accurate identification of areas of interest. Python is the language that brings together computer vision, machine learning, and all the other parts of the system [103]. The system also has characteristics that make it more useful and easier to use. It changes its sensor settings to work well in different lighting circumstances, from low light to direct sunlight [80]. The camera can be placed in several locations, such as the dashboard, rearview mirror, or steering wheel, to fit different types of cars.

Advantages of the System

- Better safety: Sends alerts on time to stop accidents caused by tiredness.
- Cost-Effective Solution: Uses hardware and software that are easy to find and use.
- Customizable: Can be set up to work with different types of cars and situations.
- Doesn't bother the driver: works without getting in the way of what they're doing.
- The cascades are just sets of XML files that contain OpenCV data to help find things. You set up your code with the cascade you desire, and then it does the rest.

Future Enhancements

Future improvements to the sleepiness detection system will focus on making it more useful, efficient, and flexible [86]. One big improvement might be linking the system to advanced driver-assistance systems (ADAS) so it can automatically respond, such as slowing the car or calling for help if the driver doesn't respond [102]. Adding more factors, such as heart rate and head posture, to a multi-factor analysis can help improve detection accuracy. Edge computing can be used to analyse video feeds on devices, speeding up processing and reducing latency [90]. Using infrared cameras or advanced image enhancement techniques to make the system more adaptable to low-light conditions can further enhance its reliability. The technology can also be used with wearable devices, such as smart glasses or headbands, to track physiological data, including brain activity and eye movements [79]. Adding advanced deep learning architectures, such as transformers, to AI models can make them more accurate and faster to respond [108]. To reduce false positives, you can also create personalised thresholds for each driver based on how they drive and how they sleep.

Conclusion

We built a system in this Python project that alerts sleepy drivers. You can use it in many different ways. We used a Haar cascade classifier in OpenCV to detect faces and eyes, and then a CNN to predict what was happening. The system recognises important features, such as the shape and position of the eyelids, to ensure correct categorisation. Detection of open eyes. Finding open eyes is an important part of the sleepiness detection system, as it provides a baseline for assessing how alert the driver is. The system focuses on the eye area using the Region of Interest (ROI) identified during the facial detection procedure. Advanced computer vision algorithms analyse the eye's features and determine whether it is open or closed. Closed-eye detection is an important part of the sleepiness detection system, as it helps identify signals that indicate the driver is tired. The system measures how long the eyes are closed when it detects that they are closed. If the driver's eyelids stay closed for longer than a set amount of time, which could mean they are becoming asleep or drowsy, the system sends a warning. This signal usually sounds like a buzzer or siren to let the motorist know right away when they are sleepy. The technology can detect even the smallest indicators of tiredness and provide early feedback to prevent potential problems by continuously monitoring eye closure.

References

1. A. George and A. Routray, "Real-time eye gaze direction classification using convolutional neural network," in Proc. 2016 Int. Conf. Signal Process. Commun. (SPCOM), Bangalore, India, 2016.
2. G. Chandan, A. Jain, H. Jain, and Mohana, "Real-time object detection and tracking using

deep learning and OpenCV," in Proc. 2018 Int. Conf. Inventive Res. Comput. Appl. (ICIRCA), Coimbatore, India, 2018.

3. P. Jothilingam, “Digital twin technologies for ICS: Leveraging virtualization and sensor data for FAT/SAT, commissioning and predictive risk detection,” International IT Journal of Research, vol. 1, no. 1, pp. 45–49, Oct. 2023.
4. P. Jothilingam, “Modernization of industrial legacy control systems: Migration strategies, risk mitigation, and operational continuity in critical infrastructures,” International Journal of Unique and New Updates, vol. 5, no. 2, pp. 39–44, Aug. 2023.
5. D. R. Janardhana, K. Shivanna, M. Ghouse Shukur, C. P. Vijay, H. R. Mahalingegowda, and H. V. Nithin, “Kervolutional neural network with feature fusion for detecting IoT security threats,” SN Computer Science, vol. 6, no. 8, p. 979, 2025.
6. J. D. Rangappa, A. P. Manu, S. Kariyappa, S. K. Chinnababu, G. H. Lokesh, and F. Flammini, “A lightweight blockchain to secure data communication in IoT network on healthcare system,” International Journal of Safety & Security Engineering, vol. 13, no. 6, pp. 1015–1024, 2023.
7. K. Shivanna, H. P. Hema, D. R. Janardhana, and P. M. Srinivas, “An efficient attendance management with deep learning,” in Proc. 2024 Int. Conf. on Computing, Semiconductor, Mechatronics, Intelligent Systems and Communications (COSMIC), Nov. 2024, pp. 18–23.
8. S. J. Niranjan, R. M. Lingaraju, B. V. B. Prabhu, and V. N. Vinaykumar, “Coconut disease prediction using neural attention mechanism with Adamax optimizer-based convolutional neural network,” International Journal of Environmental Sciences, vol. 11, no. 5, pp. 2352–2360, 2025.
9. S. J. Niranjan, R. M. Lingaraju, and J. A. Babu, “Spiralizer strategy with wolf pack algorithm and temporal convolution network–gated recurrent unit for coconut crop yield prediction,” SN Computer Science, vol. 5, Art. no. 1108, 2024.
10. D. Narasappa, "Integrating Zero Trust Architecture with Automation and Analytics for Resilient Cybersecurity," 2025 3rd International Conference on Data Science and Network Security (ICDSNS), Tiptur, India, 2025, pp. 1-6.
11. D. Narasappa, "AI-Driven Security Measures for IoT Networks Utilizing Machine Learning for Anomaly Detection," 2025 IEEE 4th World Conference on Applied Intelligence and Computing (AIC), GB Nagar, Gwalior, India, 2025, pp. 134-139.
12. I. Ganie and S. Jagannathan, “Adaptive control of robotic manipulators using deep neural networks,” Proc. 6th IFAC Conf. Intell. Control Autom. Sci. (ICONS), vol. 55, no. 15, pp. 148–153, 2022.
13. I. Ganie and S. Jagannathan, “Continual online learning-based optimal tracking control of nonlinear strict-feedback systems: Application to unmanned aerial vehicles,” Complex Eng. Syst., vol. 4, no. 1, 2024.
14. T. Jesudas, S. Ramesh, K. Senthilkumar, and R. M. Arunachalam, "Study and optimization of overcut on most influencing process parameters of micro EDM," International Journal of Applied Engineering Research, vol. 10, no. 50, pp. 13678–13683, 2015.
15. L. Selvarajan, K. Venkataramanan, S. Devaraj, and T. Jesudas, "The effect of rotary EDM parameters on the surface integrity and geometric parameters of holes in Si3N4-TiN electro conductive ceramics using DFA-RSM-ANN method," Tribology International, vol. 200, p. 110050, 2024.
16. T. Jesudas, R. M. Arunachalam, K. S. Jayakumar, and R. Thiraviam, "Parametric optimization of wire EDM – A Taguchi approach," Manufacturing Technology Today, vol. 9, pp. 9–12, 2007.

17. T. Jesudas and R. M. Arunachalam, "Development and performance study of sparkgap controller for table top micro-EDM," Asian Journal of Research in Social Sciences and Humanities, vol. 6, no. 3, pp. 145–155, 2016.

18. A. Dhanai, H. S. Bagde, R. Gera, K. Mukherjee, C. Ghildiyal, and H. Yadav, "Case report on irritational fibroma," Journal of Pharmacy and Bioallied Sciences, vol. 16, suppl. 1, pp. S960–S962, Feb. 2024.

19. H. Bagde, A. Dhopte, F. Bukhary, N. Momenah, F. Akhter, O. Mahmoud, K. P. Shetty, M. A. Shayeb, H. Abutayyem, and M. K. Alam, "Monkeypox and oral lesions associated with its occurrence: a systematic review and meta-analysis," F1000Research, vol. 12, p. 964, Mar. 2024.

20. H. Bagde, R. S. Karki, S. Husain, S. Khan, V. Haripriya, and P. Purwar, "Evaluation of microbiological flora in endo-perio lesions before and after treatment," Journal of Pharmacy and Bioallied Sciences, vol. 17, suppl. 2, pp. S1707–S1709, Jun. 2025.

21. B. Shyamsukha, H. Bagde, A. Sharan, M. Choudhary, A. Duble, and A. V. Dhan, "Evaluating the potential of ChatGPT as a supplementary intelligent virtual assistant in periodontology," Journal of Pharmacy and Bioallied Sciences, vol. 17, suppl. 2, pp. S1415–S1417, Jun. 2025.

22. H. S. Bagde, M. K. Alam, A. K. A. Alhamwan, H. M. H. Aljubab, F. F. A. Alrashedi, D. H. M. Aljameeli, and M. G. Sghaireen, "The effect of a low-carbohydrate diet on periodontal health and inflammation in patients with type 2 diabetes," Journal of Pharmacy and Bioallied Sciences, vol. 16, suppl. 1, pp. S641–S643, Feb. 2024.

23. H. S. Bagde, M. K. Alam, Y. E. M. Almohammed, S. M. M. Almaqawid, A. W. N. Alanazi, F. T. F. Alanazi, and M. G. Sghaireen, "The efficacy of platelet-rich plasma as an adjunct to bone grafting in alveolar ridge preservation following tooth extraction," Journal of Pharmacy and Bioallied Sciences, vol. 16, suppl. 1, pp. S564–S566, Feb. 2024.

24. S. B. Mangalekar, H. S. Bagde, M. Sale, S. V. Jambhekar, C. Patil, and C. V. Deshmukh, "Comparing laser-assisted and conventional excision in the management of oral soft lesions: a prospective clinical study," Journal of Pharmacy and Bioallied Sciences, vol. 16, suppl. 1, pp. S859–S861, Feb. 2024.

25. M. K. Alam, H. S. Bagde, A. K. A. Alhamwan, H. M. H. Aljubab, F. F. A. Alrashedi, D. H. M. Aljameeli, and M. G. Sghaireen, "Comparing the long-term success rates of immediate implant placement vs delayed implant placement in patients with periodontally compromised teeth," Journal of Pharmacy and Bioallied Sciences, vol. 16, suppl. 1, pp. S626–S628, Feb. 2024.

26. H. S. Bagde, M. K. Alam, Y. E. M. Almohammed, S. M. M. Almaqawid, K. K. Ganji, and M. G. Sghaireen, "Comparing the clinical and radiographic outcomes of two different surgical approaches for treating infrabony defects in chronic periodontitis patients," Journal of Pharmacy and Bioallied Sciences, vol. 16, suppl. 1, pp. S567–S569, Feb. 2024.

27. A. Sharan, B. Pawar, H. Bagde, T. K. Chawla, A. V. Dhan, B. Shyamsukha, and S. Sharma, "Comparative evaluation of dentin hypersensitivity reduction over one month after a single topical application of three different materials: a prospective experimental study," Journal of Pharmacy and Bioallied Sciences, vol. 16, suppl. 4, pp. S3405–S3407, Dec. 2024.

28. J. Prakash, T. Sinha, H. Bagde, N. Rajegowda, S. Bhat, A. Dhopte, M. Cicciù, and G. Minervini, "Evidence-based assessment of temporomandibular disorders in complete denture versus partial denture users: a systematic review," Minerva Dental and Oral Science, Sep. 2025.

29. J. R. Rogers, Y. Wang, N. F. Khan, K. Mott, V. K. Nomula, D. Wang, P. C. Fiduccia, M. Burcu, and X. Liu, "Landscape assessment of clone-censor-weight methodology application

in real-world data studies: A scoping review," in Proceedings of the Pharmacoepidemiology and Drug Safety Conference, vol. 33, pp. 424–424, Nov. 1, 2024.

30. S. K. R. Padur, "Empowering Developer & Operations Self-Service: Oracle APEX + ORDS as an Enterprise Platform for Productivity and Agility," Int. J. Sci. Res. Sci. Eng. Technol., vol. 4, no. 11, pp. 364–372, 2018.
31. S. K. R. Padur, "The Future of Enterprise ERP Modernization with AI: From Monolithic Systems to Generative, Composable, and Autonomous Platforms," J. Artif. Intell. Mach. Learn. & Data Sci., vol. 3, no. 1, pp. 2958–2961, 2025.
32. S. K. R. Padur, "Automation-First Post-Merger IT Integration: From ERP Migration Challenges to AI-Driven Governance and Multi-Cloud Orchestration," Int. J. Sci. Res. Sci. Eng. Technol., vol. 12, no. 5, pp. 270–280, 2025.
33. S. K. R. Padur, "Network Modernization in Large Enterprises: Firewall Transformation, Subnet Re-Architecture, and Cross-Platform Virtualization," Int. J. Sci. Res. Sci. Eng. Technol., vol. 2, no. 5, 2016, Zenodo.
34. S. K. Somayajula, "Enterprise Data Migration Success Patterns: Lessons from Large-Scale Transformations," International Journal of Research in Computer Applications and Information Technology (IJRCAIT), vol. 8, no. 1, pp. 757-776, Jan.-Feb. 2025.
35. S. K. Somayajula, "Demystifying Modern Data Warehousing: From Traditional to Cloud-Native Solutions," International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 2025.
36. S. K. Somayajula, "Building a Career in Enterprise Data Architecture: A Practical Guide," International Research Journal of Modernization in Engineering Technology and Science (IRJMETS), vol. 7, no. 1, Jan. 2025.
37. S. K. Somayajula, "Advanced ETL Optimization: A Framework for Next-Generation Data Integration," International Journal of Computer Engineering and Technology (IJCET), vol. 16, no. 1, pp. 381-406, Jan.-Feb. 2025.
38. S. Somayajula and A. Orlovsky, "Proof, Truth and Contradiction in the System and Meta-System: Comprehensive Mathematical Solutions and Implementation Framework," 2025.
39. P. Nutalapati, "Automated Incident Response Using AI in Cloud Security," Journal of Artificial Intelligence, Machine Learning and Data Science, vol. 2, no. 1, pp. 1301–1311, 2024
40. P. Nutalapati, "Data Leakage Prevention Strategies in Cloud Computing," European Journal of Advances in Engineering and Technology, vol. 8, no. 9, pp. 118–123, 2025.
41. P. Nutalapati, "Security Considerations for Hybrid Cloud Deployments in Fintech Using Blockchain," Journal of Artificial Intelligence, Machine Learning and Data Science, vol. 1, no. 1, URF Publishers, 2025.
42. P. Nutalapati, "Zero Trust Architecture in Cloud-Based Fintech Applications," Journal of Artificial Intelligence & Cloud Computing, vol. 2, no. 1, pp. 1–8, 2023.
43. P. Nutalapati, J. R. Vummadi, S. Dodda and N. Kamuni, "Advancing Network Intrusion Detection: A Comparative Study of Clustering and Classification on NSL-KDD Data," 2025 International Conference on Data Science and Its Applications (ICoDSA), Jakarta, Indonesia, 2025, pp. 880-885.
44. P. Nutalapati, S. M. Dhavale, A. Shrivastava, R. V. S. Praveen, H. K. Vemuri and R. RiadHwsein, "IoT and Machine Learning-Enhanced Energy Management in Enabled Smart Grids for Predictive Load Balancing," 2025 World Skills Conference on Universal Data Analytics and Sciences (WorldSUAS), Indore, India, 2025, pp. 1-6.

45. P. Nutralapati, “Disaster Recovery and Business Continuity Planning in Cloud-Blockchain Infrastructures,” SSRN Electron. J., Jun. 2020. Available: https://ssrn.com/abstract=5219038. doi:10.2139/ssrn.5219038.

46. V. P. Ramesh Kumaar and B. Balaji, "Asian Research Consortium," Asian Journal of Research in Social Sciences and Humanities, vol. 6, no. 8, pp. 1474–1491, Aug. 2016.

47. V. P. Rameshkumaar, "An Analytical Study on the Usage and Effects of Transportation Management of Top 3 MNC Companies During and Before Pandemic Crisis," International Journal of Aquatic Science, vol. 12, no. 3, pp. 2619–2629, Jun. 1, 2021.

48. V. P. Rameshkumaar, D. Ganesan, S. Revathy, R. Karthikeyan, and P. P. Devi, "Evaluating the impact of artificial intelligence on logistics and supply chain efficiency," Indian Journal of Natural Sciences, vol. 15, no. 88, pp. 88823–88828, Feb. 2025.

49. V. P. Rameshkumaar, R. A. Arul Raja, S. M. Shavalliuddin, and S. J. Fakhriddin Ugli, "Phyto assisted fabrication of highly stable titania nanoparticles for environmental purification applications," AIP Conf. Proc., vol. 3306, p. 040004, 2025.

50. V. Kumar, P. P. Devi, T. N. Babu, A. S. Nader, A. A. S. Mohammed and R. Saravanakumar, "AI-Powered Recruitment Marketing Enhancing Candidate Experience and Employer Branding," 2025 IEEE International Conference on Emerging Technologies and Applications (MPSec ICETA), Gwalior, India, 2025, pp. 1-6.

51. V. P. Rameshkumaar, D. Ganesan, S. Revathy, R. Karthikeyan, and P. P. Devi, "Evaluating the impact of artificial intelligence on logistics and supply chain efficiency," Indian Journal of Natural Sciences, vol. 15, no. 88, pp. 88823–88828, Feb. 2025.

52. Y. A. Abduvakhob Kizi, P. Praba Devi, S. Mahadevan, and S. D. S. Ugli, "Words and algorithms: The intersection of linguistic and artificial intelligence," AIP Conference Proceedings, vol. 3306, no. 1, p. 050005, 2025.

53. S. G. K. Peddireddy, “Advancing Threat Detection in Cybersecurity through Deep Learning Algorithms,” FMDB Transactions on Sustainable Intelligent Networks., vol.1, no. 4, pp. 190–200, 2024.

54. S. G. K. Peddireddy, “Integrating AI for Proactive Network Defense against Emerging Security Vulnerabilities,” FMDB Transactions on Sustainable Computer Letters., vol. 2, no. 4, pp. 232–241, 2024.

55. S. G. K. Peddireddy, “Optimizing Resource Allocation in Multi-Cloud Environments for Cost Efficiency and Scalability,” FMDB Transactions on Sustainable Computing Systems., vol. 2, no. 4, pp. 167–177, 2024.

56. V. Hiremath, “Quantum Networking: Strategic Imperatives for Enterprises and Service Providers in the Emerging Quantum Era,” Journal of Computational Analysis and Applications (JoCAAA), vol. 31, no. 3, pp. 617–631, Dec. 2023.

57. V. Hiremath, "AI-Optimized Adaptive Routing for High-Performance Data Centers: A Predictive Framework for Dynamic Network Optimization," 2025 IEEE 4th World Conference on Applied Intelligence and Computing (AIC), GB Nagar, Gwalior, India, 2025, pp. 152-157.

58. V. Hiremath, "Optimizing SDN Controller Placement for Enhanced Performance and Scalability in Large-Scale Networks," 2025 International Conference on Intelligent Communication Networks and Computational Techniques (ICICNCT), Bidar, India, 2025, pp. 1-6.

59. D. K. Arora et al., “An in vitro assessment of microleakage of pit and fissure sealants and restorative materials using dye penetration method,” Journal of Pharmacy and Bioallied Sciences, Feb. 2025.

60. R. Nagar et al., “In vitro analysis of compressive strength of three different aesthetic restorative materials,” Journal of Pharmacy and Bioallied Sciences, Feb. 2025.

61. N. Maiti et al., “Assessment of the efficacy of photobiomodulation (PBM) therapy in periodontal treatment: a longitudinal study,” Journal of Pharmacy and Bioallied Sciences, vol. 16, no. Suppl 3, pp. S2449–S2451, Jul. 2024.

62. I. Ganie and S. Jagannathan, “Continual optimal adaptive tracking of uncertain nonlinear continuous-time systems using multilayer neural networks,” Proc. 2023 Amer. Control Conf. (ACC), San Diego, CA, USA, 2023, pp. 3395–3400.

63. Kumar, P.R., Mohammad, G.B., Narsimhulu, P., Narasappa, D., Maguluri, L.P. et al. (2025). Computer Modeling Approaches for Blockchain-Driven Supply Chain Intelligence: A Review on Enhancing Transparency, Security, and Efficiency. Computer Modeling in Engineering & Sciences, 144(3), 2779–2818.

64. V. V. N. Nagaraju, A. B. Jayachandra, A. Stateczny, S. H. Yogesh, R. M. Lingaraju, and B. P. B. Veeranna, “Dual strategy based improved swarm intelligence and stacked LSTM with residual connection for land use land cover classification,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 18, pp. 4188–4198, 2025.

65. P. M. Srinivas, K. Shivanna, D. R. Janardhana, and S. Samarth, “SAAI—Smart automated anti-violence intervention,” in Proc. 2024 Int. Conf. on Computing, Semiconductor, Mechatronics, Intelligent Systems and Communications (COSMIC), Nov. 2024, pp. 82–87.

66. D. R. Janardhana, K. Shivanna, and A. P. Manu, “Security and privacy in military application using blockchain,” in Artificial Intelligence for Military Applications with Blockchain. Boca Raton, FL, USA: CRC Press, 2025, pp. 1–18.

67. D. R. Janardhana, A. P. Manu, K. Shivanna, and K. C. Suhas, “Malware analysis and its mitigation tools,” in Malware Analysis and Intrusion Detection in Cyber-Physical Systems. Hershey, PA, USA: IGI Global, 2023, pp. 263–284.

68. P. Jothilingam, “Systems and management innovation in Industry 4.0: Redefining organizational models, human–machine collaboration, and process efficiency,” in Proc. Int. Conf. Innovative Trends in Engineering and Technology, India, Jul. 2022, pp. 699–706.

69. P. Jothilingam, “Industrial Internet of Things (IIoT): AI-driven anomaly detection and multi-protocol communication across Modbus and EtherNet/IP networks,” International Journal of Enhanced Research in Science, Technology & Engineering, vol. 11, no. 3, pp. 138–143, Mar. 2022.

70. A. K. Joshi and S. B. Kulkarni, “Flow analysis of vehicles on a lane using deep learning techniques,” J. Adv. Inf. Technol., vol. 14, no. 6, pp. 1354–1364, 2023.

71. A. K. Joshi, V. Shirol, S. Jogar, P. Naik, and A. Yaligar, “Credit card fraud detection using machine learning techniques,” Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol., vol. 6, no. 3, pp. 436–442, 2020.

72. A. K. Joshi and S. B. Kulkarni, “Multi-modal information fusion for localization of emergency vehicles,” Int. J. Image Graph., vol. 24, no. 1, Art. no. 2550050, 2024.

73. A. K. Joshi and S. B. Kulkarni, “Multimodal deep learning information fusion for fine-grained traffic state estimation and intelligent traffic control,” Int. J. Intell. Syst. Appl. Eng., vol. 11, no. 3, pp. 1020–1029, 2023.

74. V. S. A. Anala, A. R. Pothu, and S. Chintapalli, “Enhancing Preventive Healthcare with Wearable Health Technology for Early Intervention,” FMDB Transactions on Sustainable Health Science Letters., vol.2, no.4, pp. 211–220, 2024.

75. V. S. A. Anala and S. Chintapalli, “Scalable Data Partitioning Strategies for Efficient Query

Optimization in Cloud Data Warehouses," FMDB Transactions on Sustainable Computer Letters., vol. 2, no. 4, pp. 195–206, 2024.

76. S. N. Akhter, R. Kumari, and A. Kumar, "Fertility booster effect of Asparagus recemosus against arsenic induced reproductive toxicity in Charles Foster rats," J. Adv. Zool., vol. 45, no. 5, 2024. [Online]. Available: http://jazindia.com.
77. Z. Hashmi, R. Kumari, and A. Kumar, "Antidote effect of Bacopa Koneru against arsenic induced toxicity in rats," J. Adv. Zool., vol. 45, no. 5, 2024. [Online]. Available: http://jazindia.com.
78. Z. Hashmi, R. Kumari, and A. Kumar, "Phytoremedial effect of Ocimum sanctum against arsenic induced toxicity in Charles Foster rats," J. Adv. Zool., vol. 45, no. 5, 2024. [Online]. Available: http://jazindia.com. ISSN: 0253-7214.
79. B. Kumari, P. Das, and R. Kumari, "Accelerated processing of solitary and clustered abasic site DNA damage lesion by APE1 in the presence of aqueous extract of Ganoderma lucidum," J. Biosci., vol. 41, pp. 265–275, 2016.
80. R. Kumari, R. K. Singh, N. Kumar, and R. Kumari, "Preparation of superfine Bael leaf nanopowder, physical properties measurements and its antimicrobial activities," Egypt. Chem. Bull., vol. 12, no. 4, pp. 284–297, 2023.
81. M. K. Sinha, R. Kumari, and A. Kumar, "Ameliorative effect of Ganoderma lucidum on sodium arsenite induced toxicity in Charles Foster rats," J. Adv. Zool., vol. 45, no. 5, 2024. [Online]. Available: http://jazindia.com.
82. Md S. Miah and Md S. Islam, "Big Data Analytics Architectural Data Cut off Tactics for Cyber Security and Its Implication in Digital forensic," in Proc. 2022 Int. Conf. Futuristic Technol. (INCOFT), Belgaum, India, 2022, pp. 1–6.
83. M. A. Obaida, Md S. Miah, and Md. A. Horaira, "Random Early Discard (RED-AQM) Performance Analysis in Terms of TCP Variants and Network Parameters: Instability in High-Bandwidth-Delay Network," Int. J. Comput. Appl., vol. 27, no. 8, pp. 40–44, Aug. 2011.
84. A. Srivastava, "Use of Python in Data Science, Data Integration and Data Engineer," Int. J. Sci. Res. Eng. Manag., vol. 8, no. 7, 2024.
85. A. Srivastava, "AI in Healthcare and its Future," J. Artif. Intell. Cloud Comput., vol. 1, no. 1, pp. 1–2, Mar. 2022.
86. A. Srivastava, "Cloud Replacing Traditional Database," Int. J. Multidiscip. Res., vol. 7, no. 2, pp. 1–2, Mar.–Apr. 2025.
87. A. Srivastava, "Data Transformation Normalization to Denormalization in Cloud," Int. J. Core Eng. Manag., vol. 6, no. 7, pp. 249–252, 2020.
88. A. Srivastava, "Impact of AI/ML on Job Market and Skills Set and Health Industry," ESP J. Eng. Technol. Adv., vol. 4, no. 3, pp. 122–126, 2024.
89. A. Srivastava, "Integrated Snowflake with AWS and AWS ETL Migration," Int. J. Multidiscip. Res., vol. 7, no. 1, pp. 1–6, Jan.–Feb. 2025.
90. A. Srivastava, "Securing PHI Data in Healthcare," Sci. Data Learn. Mach. Intell. Artif. J., vol. 2, no. 4, pp. 1678–1679, Oct. 2024.
91. A. Srivastava, "Use of AI/ML in Data Security," J. Artif. Intell. Mach. Learn. Data Sci., vol. 2, no. 2, pp. 1–3, 2024.
92. A. Srivastava, "WFH Impact on Work Culture and Future of Work Place," Int. J. Innov. Res. Creat. Technol., vol. 8, no. 6, pp. 1–8, Nov. 2020.

93. A. Srivastava, “Impact of AI on Social Media and its Implication Mental Health,” Int. J. Sci. Res. Eng. Manag., vol. 7, no. 7, 2023.

94. A. Srivastava, “WFH Impact on Work Culture and Future of Work Place,” Int. J. Innov. Res. Creat. Technol., vol. 8, no. 6, pp. 1–8, 2022.

95. G. Kashyap, "Multilingual NLP: Techniques for Creating Models that Understand and Generate Multiple Languages with Minimal Resources," Int. J. Sci. Res. Eng. Manag. Sci., vol. 6, no. 12, pp. 1–5, Dec. 2024.

96. G. Kashyap, "Adversarial Attacks and Defenses: Investigating How AI Systems Can Be Manipulated Through Adversarial Inputs and Methods to Defend Against Them," Int. J. Innov. Res. Creat. Technol., vol. 7, no. 1, pp. 1–6, Feb. 2021.

97. G. Kashyap, "Reinforcement Learning in Dynamic Environments: Developing RL Algorithms That Can Adapt to Continuously Changing Environments," Int. J. Innov. Res. Creat. Technol., vol. 8, no. 4, pp. 1–10, Jul. 2022.

98. G. Kashyap, "Impact of AI on blockchain scalability: Solutions and trade-offs," Int. J. Core Eng. Manag., vol. 7, no. 1, pp. 197–202, 2022.

99. G. Kashyap, "Explainable AI (XAI): Methods and techniques to make deep learning models more interpretable and their real-world implications," Int. J. Innov. Res. Eng. Multidisciplinary Phys. Sci., vol. 13, no. 1, pp. 1–8, Jan. 2025.

100. G. Kashyap, "Knowledge transferring techniques: AI in learning transfer," Science Data & Learning Machine Intelligence Artificial Journal, vol. 4, no. 1, pp. 1898–1902, Dec. 2023.

101. G. Kashyap, "AI in blockchain-enabled healthcare systems: Transforming healthcare delivery and data security," ESP Journal of Engineering & Technology Advancements, vol. 3, no. 8, pp. 109, 2023.

102. G. Kashyap, "Knowledge graphs and NLP: Integrating structured knowledge into NLP systems for better reasoning and contextual understanding," Int. J. Innov. Res. Eng. Multidiscip. Phys. Sci., vol. 13, no. 2, pp. 1–7, Mar.–Apr. 2025.

103. G. Kashyap, "Self-supervised learning: How self-supervised learning approaches can reduce dependence on labeled data," Int. J. Innov. Res. Creat. Technol., vol. 10, no. 4, pp. 1–10, Jul. 2024.

104. G. Kashyap, "Unsupervised learning for high-dimensional data: Advancements in unsupervised learning techniques like clustering, anomaly detection, and dimensionality reduction," Int. J. Leading Res. Publ., vol. 5, no. 12, pp. 1–10, Dec. 2024.

105. G. Kashyap, "Meta-learning: Learning to learn: Investigating how meta-learning algorithms can improve learning efficiency across tasks," Int. J. Sci. Res. Eng. Manag., vol. 8, no. 12, pp. 1–7, Dec. 2024.

106. G. Kashyap, "Infrastructure automation using Ansible and YAML for IT environment build in the health insurance industry," Int. J. Innov. Res. Creat. Technol., vol. 10, no. 6, pp. 1–10, Dec. 2024.

107. G. Kashyap, "Quantum machine learning: Exploring the intersection of quantum computing and AI," Int. J. Innov. Res. Eng. Multidiscip. Phys. Sci., vol. 13, no. 2, pp. 1–8, Mar.–Apr. 2025.

108. G. Kashyap, "AI for robustness and fairness: Addressing bias, fairness, and robustness in machine learning algorithms," Sci. Data Learn. Mach. Intell. Artif. J., vol. 1, no. 3, pp. 1–15, Jan. 2025.