iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** III **Month of publication:** March 2025

DOI: https://doi.org/10.22214/ijraset.2025.67918

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue III Mar 2025- Available at www.ijraset.com

Sleep Sense

Shiv Bhardwaj[1], Swati Pandit[2], Yash Kaundinya[3], Tusha Singh[4], Gauri[5], Shruti Varshney[6]
Department of Computer Science and Engineering /IT, VCTM ALIGARH

Abstract: ***There is a large number of cases of road accidents across the world out of which a significant number of accidents are caused due to driver fatigue and drowsy behavior .A system which is reliable in detection of drowsy behavior and signs of fatigue can help in preventing such road accidents by monitoring driver behavior and generating alerts to them when signs of fatigue detected. This paper presents a machine learning approach that uses real-time image and processing it by taking insights from its processing like closing eyes ,facial expressions, calculating the closing and open eye-aspect ratios using different machine learning algorithms and generating alerts if it detects driver drowsy behavior . Experimental research shows that our system is effective and reliable in detection of drowsy behavior of drivers and it generates timely alerts to drivers if it is found drowsy and makes the driver safe.***

Keywords: Real-time image analysis, neural network algorithms, driver safety, and recognition of drowsiness.

I. INTRODUCTION

Imagine you're driving late at night, struggling to keep your eyes open. Your eyelids feel heavy, your reaction time slows down, and before you know it—you could drift off, even if just for a second. That split second is all it takes for a serious accident to happen. This is where a Drowsiness Detection System comes into play.

Driving with fatigue and drowsy behavior is one of the dangerous causes of road accidents, often as dangerous as drunk driving. When a driver is exhausted, their ability to focus and react quickly is significantly reduced. Traditional ways of staying awake, like rolling down the window or drinking coffee, might not always work. That's why modern technology has stepped in to help.

Sleep sense is designed to monitor driver fatigue and drowsy behavior in real-time using a live camera system. It can track eye movements, head position,eye closing time, and facial expressions to detect manifestation of fatigue. Some systems go one step further by using heart rate sensors or brain activity monitoring. If the Sleep Sense precepts that the driver is getting drowsy, it instantly consigns out an alert—beeping sounds, vibrations in the seat or steering wheel, or even voice warnings—to wake them up or remind them to take a break.

With advancements in AI and machine learning, these systems are becoming smarter and more reliable. Many modern cars, especially high-end models, already come equipped with some form of driver fatigue detection. But this technology isn't just for personal vehicles—it's also being used in trucks, buses, and even workplaces where staying alert is crucial, like construction and factory jobs.

In short, a Drowsiness Detection System acts as a lifesaver on the road. By recognizing fatigue before it leads to disaster, it helps make driving safer for everyone. As technology improves, we can expect these systems to become even more common and effective in preventing accidents caused by drowsy driving

FIGURE- DROWSINESS CAR ACCIDENTS

Technologies that detect or prevent accidents due to the drowsiness of drivers while driving is a major challenge. There are multiple dangers led by drowsiness of drivers on the road, it is important to create methods that can effectively counteract its effects.

The goal of this project is to create a simulation of a system for detecting tiredness. The primary goal is to create a system that reliably detects the driver's drowsy condition using eye's detection, guaranteeing prompt notifications to avert collisions..

Outlining the software requirements for developing the "Drowsy Detector" technology is the primary objective of this paper. The project's goal is to develop an interim device that can identify driver fatigue in order to save lives and stop property damage.

A web-based, immediate time tool called "Sleep Sense" evolved to identify drivers who are drowsy. The device continuously takes pictures, uses preset algorithms to assess the driver's eye condition, and sends out alarms as needed.

The scope of the project includes:

Alerting the driver promptly when drowsiness is detected.

Allowing users to search for specific locations or explore areas on a map.

Providing approximate time and distance estimates for reaching a chosen destination.

Enabling users to input their current and target locations, with the system suggesting the most suitable route.

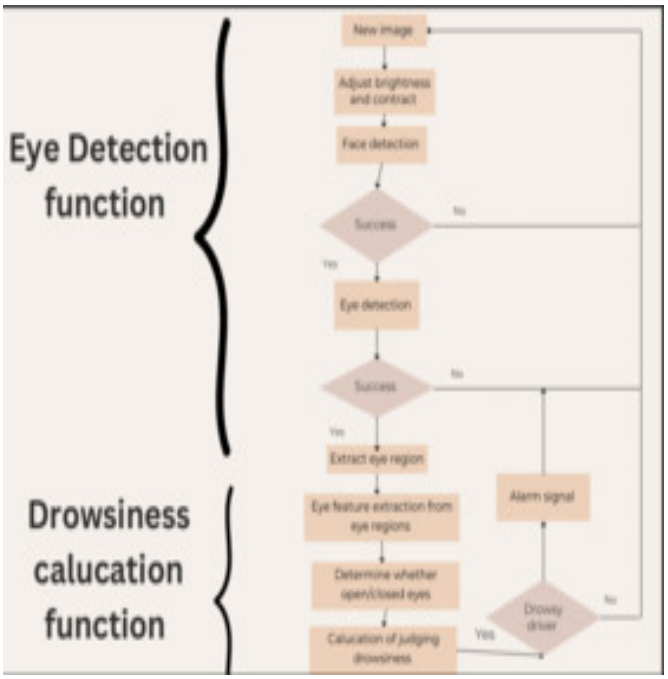


FIGURE- FLOWCHART

This flowchart outlines how a "Sleep Sense" works to monitor a driver's alertness and trigger an alarm if they appear sleepy.

Step-by-Step Process:

1) Capturing an Image – The system takes a picture of the driver.

2) Enhancing Visibility – It adjusts brightness and contrast to improve image quality.

3) Detecting the Face – The system looks for the driver's face in the image.

- If no face is found, it tries again.

4) Detecting the Eyes – Once the face is located, it zooms in to find the eyes.

- If it can't detect the eyes, it retries.

5) Examining the Eyes: The system determines if the eyes are drowsy or not and extracts information about them..

6) Judging Drowsiness– If the eyes are not in active state for too long a system detects it as drowsy .

7) Setting Off an Alarm: An alarm is toggled off to wake the driver if drowsiness is detected.

This process continuously repeats to ensure the driver stays alert and safe while driving.

Figure 1: lan

The system uses an AI-powered camera mounted on the vehicle's dashboard,which regularly looks for indications of tiredness in the driver's eyes and face.

How It Works:

1) Facial and Eye Tracking: To determine if a motorist is aware or sleepy, the system examines their head posture, blinking patterns, and gaze.
2) Real-Time Monitoring: A digital interface displays the driver's status, with indicators showing whether they are "Drowsy" or "Active". Alerts may be triggered by the system to get the driver's attention if they seem sleepy..
3) Temperature and Condition Alerts: The interface includes additional data, such as temperature readings (28° shown in the image) and system status, ensuring continuous monitoring.
4) Vehicle Safety Features: The system likely integrates with the car's safety mechanisms to provide warnings or even take preventive actions, such as slowing down or vibrating the steering wheel.

Why It Matters:

A major contributor to traffic accidents, drowsy driving is frequently just as deadly as driving while intoxicated.By warning the driver before they lose complete control, this device helps avert possible collisions.

Image analysis in computer science analyses and works with images using algorithms, enabling efficient interpretation and manipulation of visual data.

As a branch of digital signal processing, it offers several advantages over analog methods, such as the ability to apply a wider range of algorithms, reduce noise accumulation, and prevent signal distortion. Since images exist in two dimensions, digital image processing is often modeled as a multidimensional system.

A. System Review

To understand the needs and expectations of potential users, we conducted a thorough survey by reviewing various websites and applications. This research provided valuable insights and inspired new ideas for our project. Based on this analysis, we concluded that there is a significant demand for a drowsiness detection application, with ample room for innovation and improvements in this domain.

B. Technology Used

1) Python

Python is a high-level, flexible programming language that is renowned for being easy to understand and use. It is perfect for both small and large-scale projects because of its focus on clean code and support for various programming paradigms, such as procedural, object-oriented, and functional programming.

2) Jupyter Lab

Jupyter Lab is an open-source platform developed by Project Jupyter, providing tools for interactive computing across multiple programming languages.It's commonly used in data science, machine learning, and scientific research.

3) Image Processing

Digital image processing uses algorithms to analyze and work with digital images. It's a key technology for detecting and interpreting visual patterns, making it essential for this project.

4) Machine Learning

Machine learning develops systems that can learn from data and make predictions based on patterns, all without being explicitly programmed.This project analyzes and detects fatigue in drivers by training neural network models on sample data.

5) Face and Eye Detection (OpenCV Algorithm)

OpenCV algorithms are used for detecting faces and eyes within images. These algorithms are highly efficient and reliable, making them. Perfect for drowsiness detection and other real-time applications.

How Face and Eye Detection Works

The process of detecting a driver's face and eyes involves several key steps:

- *Capturing the Driver's Image*

A webcam installed on the dashboard is used by the system to continuously take pictures of the driver. This guarantees that the driver's face and eyes are monitored in real time.

- *Preprocessing the Image*

Before analyzing the image, the system enhances its quality by reducing noise and improving contrast. One common technique used is histogram equalization, which balances the brightness levels in the image, making facial features clearer and easier to detect.

- *Detecting the Face*

The driver's face is then detected by the system after scanning the picture. The face can be located with the aid of OpenCV's facial recognition computations.

- *Focusing on the Eyes*

Once the face is identified, the system defines a Region of Interest (ROI) around the eyes. This step helps in reducing unnecessary computations by focusing only on the part of the face that is crucial for detecting drowsiness.

- *Detecting Eyes in Grayscale Images*

To simplify processing and improve efficiency, the

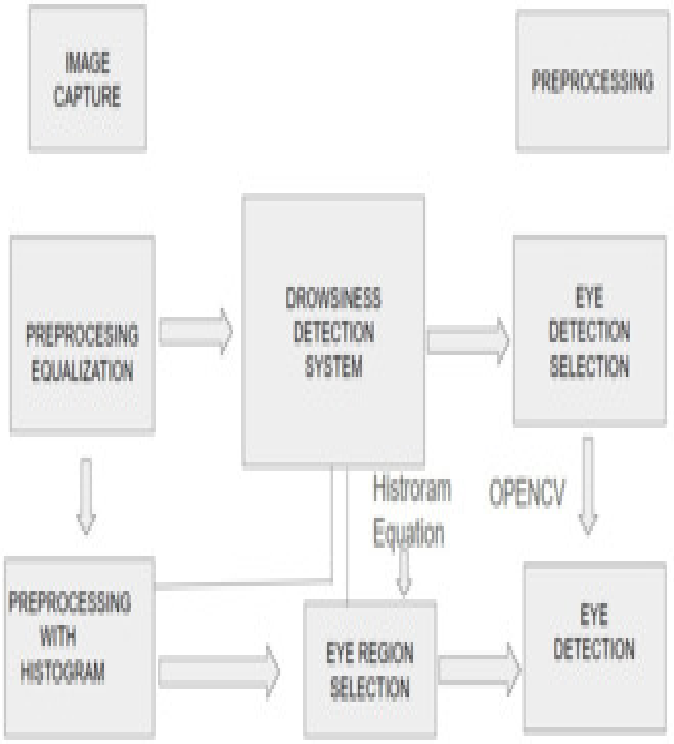


FIGURE- DROWSINESS SYSTEM

Imagine you're sitting in a car, and there's a small camera mounted on the dashboard, quietly watching over you. As you drive, the camera continuously captures images of your face. But it's not just taking pictures—it's analyzing them in real-time.

First, the system cleans up the images, reducing any noise and enhancing the details so it can work more accurately. Then, it searches for your face in each frame. If your face isn't visible—maybe you turned away—it waits for the next frame to try again.

Section	Description
Image Processing	Utilizes algorithms to analyze and manipulate digital images, enabling accurate interpretation of visual data.
System Review	A study of various websites and applications indicates a high demand for drowsiness detection systems, highlighting opportunities for innovation.
Technology Used	- **Python:** A high-level programming language with extensive support for AI and machine learning. - **Jupyter Lab:** An interactive platform for computation and research. - **Image Processing:** Algorithms for analyzing and refining digital images.
Machine Learning	Develops predictive models that learn from data, enabling automated detection and analysis of drowsiness patterns without explicit programming.
Face & Eye Detection	Uses OpenCV-based algorithms to detect faces and eyes in images, ensuring real-time and reliable drowsiness detection.
Detection Process	1. Capture image via dashboard webcam. 2. Preprocess image to enhance quality (e.g., histogram equalization). 3. Detect face in frame. 4. Define Region of Interest (ROI) for eyes. 5. Detect eyes in grayscale using OpenCV. 6. Track eye movements and issue alerts when drowsiness is detected.

II. METHODOLOGY

A functional requirement outlines a particular feature or capability that a system must have in order to satisfy user demands. The functional requirements for this methodology are as follows:

Taking a picture or recording of the driver's face before the journey begins. observing and interpreting the driver's facial expressions at all times while driving. sending out an alert if the driver exhibits any symptoms of weariness or drowsiness.

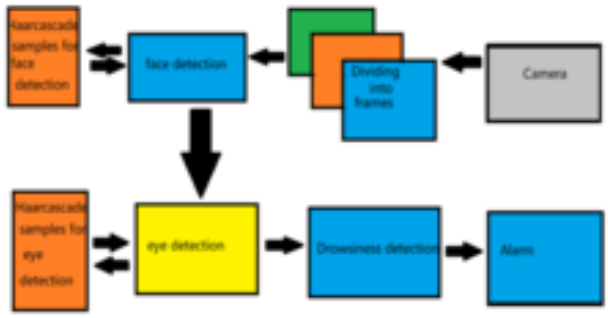


FIGURE- FLOW CHART

A. Non-Functional Needs

Non-Functional Needs

The quality standards and limitations that the system must adhere to in accordance with the project specifications are defined by non-functional requirements. Another name for these is non-behavioral requirements.

The camera used for video capture must have high resolution for clear image processing.

The system should function effectively even in low-light environments.

The alarm must be loud enough to immediately alert and wake the driver if drowsiness is detected.

System Configuration

Software Requirements

Programming Language: Python 3

Development Environment: Visual Studio Code

Hardware Requirements

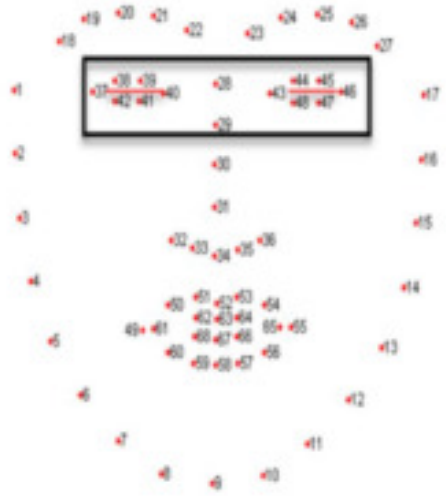
Processor: 64-bit, quad-core with a minimum speed of 2.5 GHz per core

RAM: At least 4 GB
Display: Monitor with a resolution of 1024 x 768 or higher
Camera: A functional webcam for real-time video capture
LIBRARIES:
Libraries Used
NumPy: A Python library that enables efficient handling of multidimensional arrays and provides various functions for performing fast mathematical operations on arrays.
OpenCV:An open-source Python computer vision library for processing webcam images prior to additional analysis.
TensorFlow: A powerful open-source library for machine learning and artificial intelligence, primarily used for training and running deep neural networks. In this project, it is utilized to integrate Keras.
Keras: A user-friendly open-source library that serves as an interface for building and managing artificial neural networks, making deep learning implementation easier.

B. Hardware Interface

Hardware Interface Server Side:
• The specifications are as follows:
RAM: Eight gigabytes
Hard Disk Drive (HDD): Two hundred fifty-six gigabytes or more (excluding the space occupied by data)
Processor: Intel Core i3 (10th Generation)
Client Side:
The specifications are as follows:
RAM: Four gigabytes
Processor: Octa-core, with a maximum clock speed of 2.32 GHz, Snapdragon
Software Interface :
Server Side:
OS: Windows Server 2010 or onwards
Web Server: Xampp
Client Side:
OS: android operating system
Browser: chrome, opera mini
Product Functions:
Calculating Real Time Video
Frame processing: The captured frames will undergo pre-processing, which involves turning them into greyscale. This system will be in charge of identifying the areas of the face and eyes.
Taking pictures of closed-eye frames The system detects whether the driver is sleepy.
Producing Output/Results (Alarm) The system will determine whether the driver is sleepy based on output and sound an alarm.
In order to determine drowsiness, the device camera uses the driver's eye aspect ratio and live video as input. An alert is triggered if the driver's left and right eyes are found to be closed for an extended period of time, indicating drowsiness.
Region of Interest Selection
The Haar Cascade classification algorithm is used for detecting the driver's head and facial features.
It identifies optic landmarks (key points on the face, especially around the eyes).
The algorithm calculates the absolute distance between these key points.
It will check that the targeted person's eyes are open or closed.
Improved accuracy in drowsiness detection by precisely analyzing facial features.
Using the Haar Cascade classification technique, the driver's head and facial traits were identified. Important facial landmarks, especially those surrounding the eyes, can be identified with the use of this algorithm. It establishes whether the driver's eyes are open or closed by measuring the absolute distance between particular spots on the face. This technique guarantees accurate eye condition recognition, which greatly increases the accuracy of drowsiness detection.

The driver's head and important face features were identified using the Haar Cascade classification technique. It estimated the separation between the eyes by identifying particular locations around them. This method increased the precision of determining if the driver's eyes



Eye Aspect Ratio (EAR) Calculation

- Pixel Sensitivity – The algorithm is sensitive to changes in lighting and surroundings, which can impact image processing.
- Glasses Interference – The pixel values of pupils and glasses can be very similar, making it difficult to correctly detect eye shape.

There is a significant difference in the positioning of eye landmarks between **open and closed eyes**, and the EAR formula helps measure these differences accurately.

$$EAR = \frac{||\square\square - \square\square|| + ||\square\square - \square\square||}{\square||\square\square - \square\square||}$$

FIGURE-EYE COORDINATES

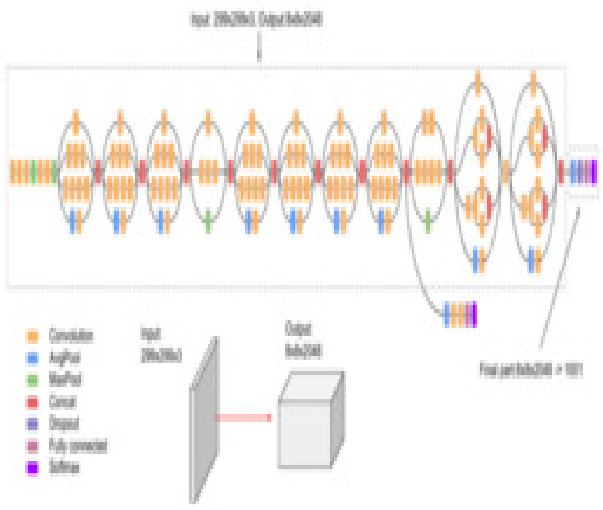
Inception V3

Convolutional Neural Networks (CNNs) were used to create the deep learning model Inception V3, which was mainly created for image classification applications. With 42 layers and a reduced error rate than previous iterations, it was released in 2015.

Key enhancements in **Inception V3** include:

- Breaking Down Large Convolutions – Splitting large convolution operations into smaller ones for better efficiency.
- Asymmetric Convolutions – Using non-square (asymmetric) convolution filters to optimize processing.
- Auxiliary Classifiers – Adding extra classifiers within the network to improve training and prevent overfitting.

- Efficient Grid Size Reduction – Optimizing how image data is processed by reducing grid sizes without losing important features.



The image appears to depict the architecture of a deep neural network, specifically an Inception-based convolutional neural network (CNN). Here's a breakdown of its contents in a humanized description:

1) Network Input & Output

- The input to the model is an image of size **299×299×3** (height × width × RGB color channels).
- The output of the feature extraction layers is a tensor of shape **8×8×2048.**
- The final output is a classification layer mapping to **1001** categories.

2) Layers & Operations Used (Color Coded)

- Orange Blocks: Convolution layers
- Blue Blocks: Average pooling layers
- Green Blocks: Max pooling layers
- Red Blocks: Concatenation operations
- Purple Blocks: Fully connected layers
- Dark Blue Blocks: Softmax layer (final classification)
- Pink Blocks: Dropout layers

3) Structure

The network consists of multiple *Inception modules*, which are represented as circled sections with multiple parallel convolutional filters.

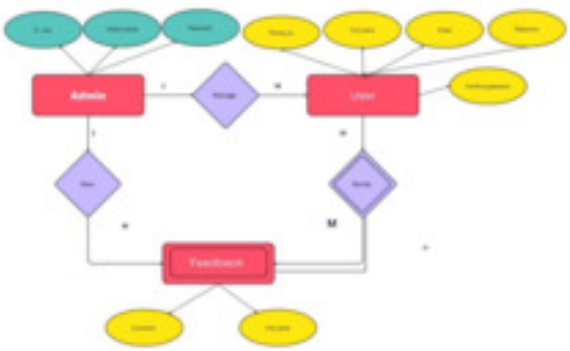
These modules process the image through different filter sizes, capturing features at multiple scales.

After passing through several inception layers, the extracted features are aggregated and transformed into a final classification output.

4) Final Processing

The *8×8×2048* feature map is further processed through *fully connected layers*, dropout, and softmax activation to predict the final class out of 1001 possible categories.

This resembles an *Inception-v3* model, which is a widely used CNN for image classification tasks.



This is an Entity-Relationship (ER) diagram representing an admin-user feedback management system. Here's how it works:

1) *Admin:*

The system has an Admin entity with attributes like E-mail, Admin-name, and Password.

The Admin is responsible for managing users and viewing feedback.

2) User Management:

A single supervisor can supervise and manage a large number of users because they have the ability to manage multiple users.

Every user has important information linked to them, such as their full name, email address, phone number, password, and password confirmation.

3) User Feedback System:

A User can Send multiple pieces of Feedback (many-to-many relationship).

The Feedback entity includes attributes like Comment and Full-Name.

4) Admin Viewing Feedback:

The Admin can View multiple Feedback entries (one-to-many relationship).

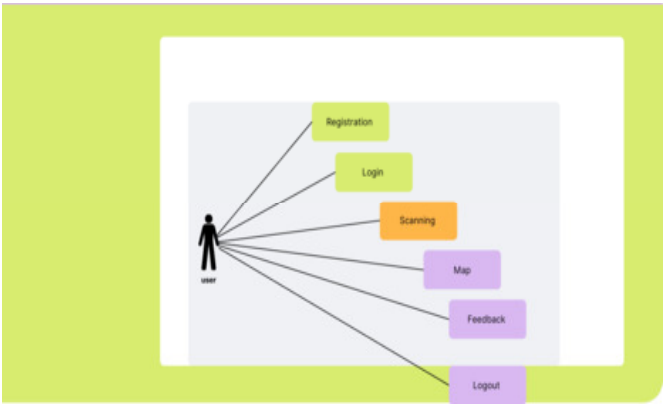


FIGURE - USER CHART

This chart illustrates how a user interacts with the system in a more natural and relatable way and showcasing various functionalities available to them.

Actors and Use Cases:

- User (Actor)

The stick figure on the left represents a User who interacts with the system.

- System Functionalities (Use Cases):

The user can perform six key actions, represented as oval shapes within the system boundary:

Registration: The user can create an account.

Login: The user can log into the system.

Scanning: The system includes a scanning feature, possibly for detecting drowsiness or analyzing input.

Map: The user can access a map feature, which may assist in navigation.
Feedback: The user can submit feedback about their experience.
Logout: The user can exit the system securely.

III. CONCLUSION

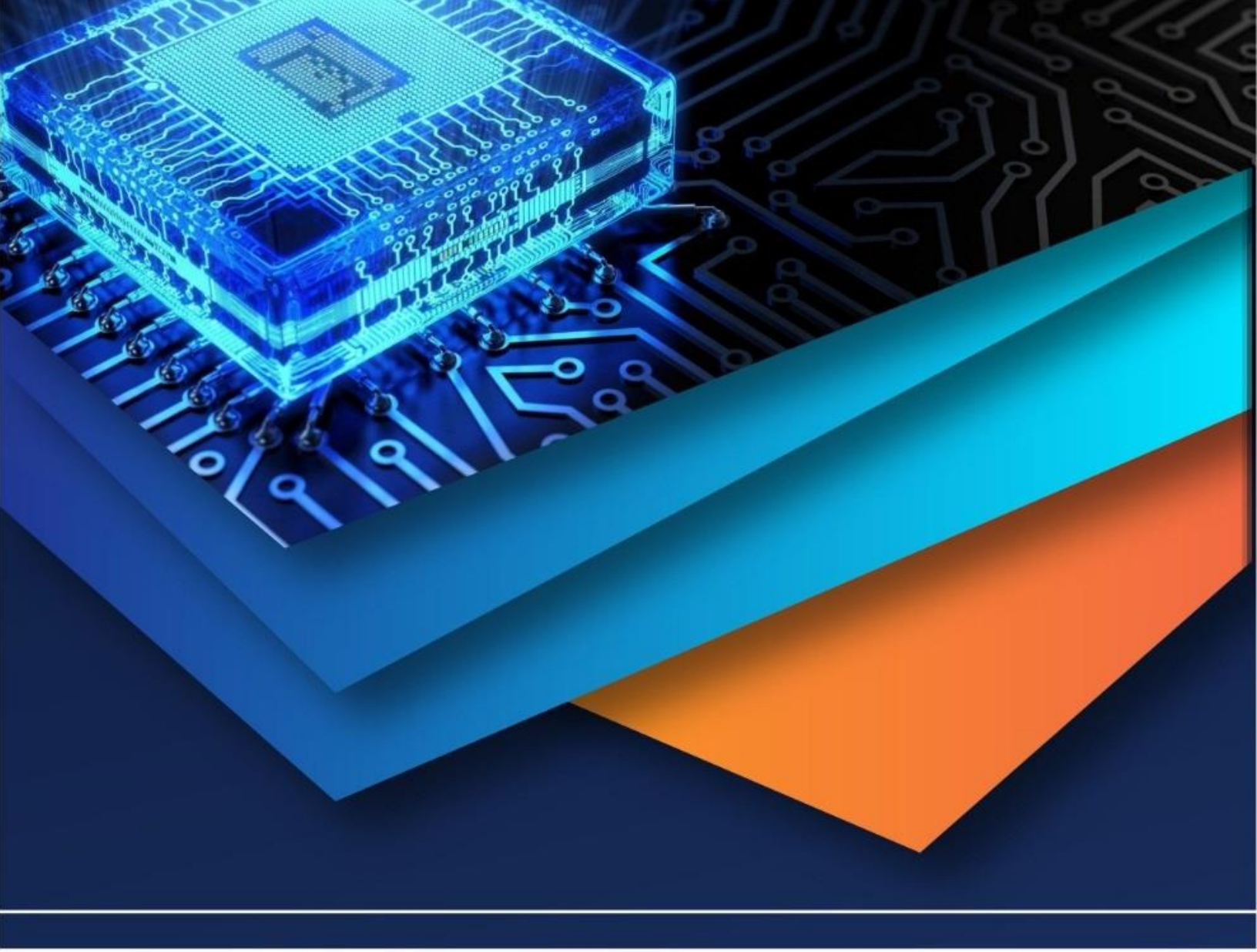
Drowsiness detection is easily helpful to analyse the person's eye while he/she is driving. It can be done when we can train the model that can identify the driver's eye in the current situation.
The model's main motive is to see eye aspect ratio (EAR) for detecting whether the person's eye is closed or open during driving. If we come to know that the EAR value lies below a certain threshold for a period,the detection counter will increase.Once this counter tries to exceeds the predefined limit then the alert is triggered to warm the driver.
The goal of this project is to minimum the number of road accidents caused due to drowsy person,this may make road safe while driving.

IV. FUTURE SCOPE

The accuracy can only be archived if the camera used is of good quality. If the eye's of driver are not detected due to bad quality of camera may effect the ability to identify the drowsiness of driver. Factors like sunglasses, reflective spectacles, poor lighting, or any obstruction between the eyes and the camera can reduce detection accuracy.
Additionally, if the driver is not facing the camera properly, the model may struggle to track eye movements effectively. Future improvements could include infrared-based eye tracking, advanced image processing techniques, and AI-driven enhancements to improve accuracy, even in challenging conditions.

REFERENCES

[1] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.
[2] E. Zhou, H. Fan, Z. Cao, Y. Jiang, and Q. Yin, "Extensive facial landmark localization with coarse-to-fine convolutional network cascade," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2013, pp. 386–391.
[3] R. Grace et al., "A drowsy driver detection system for heavy vehicles," in *Proc. 17th AIAA/IEEE/SAE Digit. Avion. Syst. Conf. (DASC)*, Oct. 1998, vol. 2, pp. I36-1–I36-8.
[4] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3476–3483.
[5] D. F. R. Ferreira, "A system for driver drowsiness detection," *Int. J. Comput. Vis. Intell. Syst.*, vol. 6, no. 4, pp. 203–217, 2018.
[6] M. A. D. Siddik and M. S. Rahman, "Drowsiness and lethargy detection using machine learning techniques," *Int. J. Artif. Intell. Res.*, vol. 8, no. 1, pp. 45–62, 2020.
[7] S. Vitabile, "Bright pupil detection in an embedded, real-time drowsiness detection system," *Sens. Actuators B Chem.*, vol. 254, pp. 210–218, 2017.
[8] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2001, vol. 1, pp. 511–518.
[9] OpenCV Documentation, "Face and eye detection using Haar cascades," *Open Source Comput. Vis. Libr.*, 2023. [Online]. Available: https://docs.opencv.org
[10] D. E. King, "Dlib-ml: A machine learning toolkit," *J. Mach. Learn. Res.*, vol. 10, pp. 1755–1758, 2009.
[11] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.
[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[14] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.
[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.
[17] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1251–1258.
[18] M. Li and Y. Li, "A real-time driver drowsiness detection system using deep learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 1345–1356, 2022.
[19] T. Baltrušaitis, P. Robinson, and L. P. Morency, "OpenFace: An open source facial behavior analysis toolkit," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2016, pp. 1–10.
[20] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Int. Conf. Mach. Learn.*, 2015, pp. 448–456.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY