Real-Time Driver Drowsiness Detection System

Technical and Academic

[1]Jayshree Khairnar, [2]Purva Jagtap, [3]Bhavesha Sakhare, [4]Akanksha Rayjade, [5]Prachi Shinde
[1]*Assistant Profesor, AIDS Department, Guru Gobind Singh College of Engineering and Research Centre, Nashik, Maharashtra, India*
[2,3,4,5]*Student, AIDS Department, Guru Gobind Singh College of Engineering Research Centre, Nashik, Maharashtra, India*
[1]jayshree.khairnar@ggsf.edu.in , [2]purvasachinjagtap@gmail.com, [3]akanksharayjade0192@gmail.com

***Abstract*—Driver fatigue and drowsiness are among the most significant contributors to road accidents globally, often leading to severe injuries and fatalities. This project presents a Real-Time Driver Drowsiness Detection System designed to provide an automated, low-cost safety intervention. The system utilizes Computer Vision and Machine Learning techniques to monitor driver alertness through a standard webcam interface.**

The core methodology involves detecting the driver's face and eyes using Haar Cascade Classifiers, a computationally efficient object detection algorithm. Once the eye regions are localized, the Eye Aspect Ratio (EAR) is calculated using Euclidean distances between facial landmarks to determine the state of the eyes (open or closed). A temporal thresholding logic is applied to differentiate between natural physiological blinking and prolonged eye closure indicative of drowsiness.

Keywords—component; Driver Drowsiness, Computer Vision, Eye Aspect Ratio (EAR), Haar Cascade, Real-time Systems.

I. INTRODUCTION (*Heading 1*)

The rapid advancement of the automotive industry and the increasing density of road networks have made transportation a cornerstone of modern society. However, this progress is shadowed by a persistent and lethal challenge: **driver fatigue**. According to the **World Health Organization (WHO)** and various national transport authorities, drowsiness is a primary factor in approximately **20% to 30% of all vehicle accidents** worldwide.

While traditional safety measures such as airbags and Anti-lock Braking Systems (ABS) focus on mitigating the impact of a collision, there is an urgent need for **proactive, preventative technologies**.

This research paper explores the development of a **Real-Time Driver Drowsiness Detection System** that prioritizes computational efficiency and high-speed responsiveness. By utilizing a Python-based framework and **Multi-threading** architecture, the proposed system aims to provide a robust, low-cost safety solution capable of operating on standard hardware, making life-saving technology accessible for a wider range of commercial and private vehicles.

A. Selecting a Template (Heading 2)

"First, it was confirmed that the correct template for the **A4 paper size** was selected, as this is the tailored output for the **IJAMRED** proceedings. This electronic document is a live template where all standard components—including margins, column widths, and line spaces—are built-in to facilitate automatic compliance with electronic production requirements. To maintain the integrity of these specifications, no manual alterations were made to the prescribed text fonts or designations."

B. Maintaining the Integrity of the Specifications

The template is used to format the research paper and style the technical text. All margins, column widths, line spaces, and text fonts (such as Times New Roman) are prescribed and have not been altered to ensure conformity. This measurement is deliberate, using specifications that anticipate the paper as one part of the entire proceedings. For this project, specific care was taken to ensure that the **Eye Aspect Ratio (EAR) equations** and **system flowcharts** were inserted without revising the current designations of the template.

II. PREPARE YOUR PAPER BEFORE STYLING

Before beginning to format the paper, the content was written and saved as a separate text file to keep graphic files (such as EAR graphs) separate until the final styling was applied. To maintain the template's automation, the following guidelines were followed during the preparation of the "Driver Drowsiness Detection" content:

A. Abbreviations and Acronyms

To maintain the technical integrity of the paper, the following specialized terms are utilized and defined upon their first occurrence:

- **EAR (Eye Aspect Ratio):** The primary mathematical metric used to estimate the degree of eye opening based on Euclidean distances between facial landmarks.
- **ADAS (Advanced Driver Assistance Systems):** The broader category of electronic systems that aid vehicle drivers in navigation and safety.
- **OpenCV (Open Source Computer Vision Library):** The software library used for real-time image processing and frame manipulation.

- **FPS (Frames Per Second):** The frequency at which the camera captures consecutive images, determining the real-time responsiveness of the system.
- **ROI (Region of Interest):** The specific subset of an image—such as the eyes or face—targeted for localized processing to reduce computational overhead.

B. Units

- "The research primarily utilizes **SI (MKS)** units, as encouraged by the template. English units are avoided unless used as identifiers for specific hardware components, such as a "15.6-inch monitor".
- **Time Measurement:** The duration of eye closure and system latency are measured in **seconds (s)** or **milliseconds (ms)**.
- **Frequency:** The processing speed and camera input are defined in **Hertz (Hz)** or **Frames Per Second (FPS)** to quantify real-time performance.
- **Distance:** For the Eye Aspect Ratio (EAR) calculation, vertical and horizontal distances between facial landmarks are measured in **pixels (px)**.
- **Decimal Notation:** A zero is always placed before decimal points for values less than one (e.g., the EAR threshold is written as **0.25**, not .25).

C. Equations

The primary metric for drowsiness detection is the **Eye Aspect Ratio (EAR)**. For each frame, the landmarks of the eye are identified, and the EAR is calculated as follows:

$$\text{EAR} = \frac{(p2-p6) + (p3-p6)}{2(p1-p4)}$$

Where p_1…..p_6 represent the 2D facial landmark locations. To make the equation more compact and computationally efficient, the **solidus (/)** and appropriate exponents are used. In this calculation, **Roman symbols** are italicized to denote quantities and variables, while Greek symbols remain in standard type.

D. Some Common Mistakes

During the preparation and proofreading of this research, specific attention was paid to the following technical writing standards prescribed by the template:

- **Data Usage:** The word "data" is strictly treated as a **plural noun**.
- **Numerical Formatting:** A **zero** is always used before decimal points (e.g., **0.25** for the EAR threshold instead of .25).
- **Terminology:** The term **"alternatively"** is used in preference to "alternately" when discussing system logic paths.
- **Prefixes:** The prefix **"non"** is not treated as a separate word and is joined to its modifier without a hyphen (e.g., **nonintrusive** monitoring).
- **Punctuation:** Parenthetical statements at the end of sentences are punctuated **outside** the closing parenthesis.

III. Methodology

The proposed system architecture is designed for high-speed processing on standard hardware. It follows a modular pipeline consisting of image acquisition, facial feature extraction, and mathematical analysis.

A. Image Acquisition and Pre-processing (Heading 2)

The system utilizes a standard RGB webcam to capture a continuous video stream at 30 **Frames Per Second (FPS)**. To enhance computational efficiency, each frame is converted to **Grayscale**. This reduces the data complexity while preserving the structural features necessary for object detection.

B. Face and Eye Localization (Heading 2)

The system employs **Haar Cascade Classifiers** to detect the driver's face within the video frame. Once the face is identified as a **Region of Interest (ROI)**, a secondary classifier is used to localize the eye regions. This "Cascade" approach ensures that non-relevant background data is discarded early in the processing cycle.

C. Facial Landmark Mapping (Heading 2)

Using the localized eye regions, the system identifies six specific 2D coordinates (p_1 through p_6) for each eye. These landmarks represent the corners and the vertical midpoints of the eyelids.

D. Eye Aspect Ratio (EAR) Analysis (Heading 2)

The core logic of the system relies on the **Eye Aspect Ratio (EAR)**. Unlike simple pixel counting, EAR is a dimensionless quantity that remains consistent regardless of the driver's distance from the camera. The EAR is calculated using the formula defined in **Equation (1)**:

$$\text{EAR}=\frac{(p2-p6) + (p3-p6)}{2(p1-p4)}$$

E. Drowsiness Decision Logic (Heading 2)

To distinguish between a natural blink and a fatigue event, the system implements a **temporal threshold**. A threshold value of **0.25** is established. If the EAR remains below this value for a duration exceeding **20 consecutive frames** (approximately 0.6 seconds), a drowsiness state is triggered.

F. Multi-threaded Alert System (Heading 2)

To maintain system integrity, the alarm trigger is handled by a separate execution thread. This prevents the "Global Interpreter Lock" from pausing the video processing while the auditory alert is active, ensuring zero-latency monitoring

ISSN: 3107-6513

IV. System Architecture

As illustrated in Fig. 1, the system architecture consists of:

1.Data Acquisition (Input Stage)
The system starts by capturing live video frames using cv2.VideoCapture(0).

2. Image Pre-processing
To increase computational efficiency, the raw frame is converted into Grayscale.

3. Feature Detection (The ML Engine)
This is the heart of the system. We use two Haar Cascade Classifiers:
Face Classifier: Scans the frame to locate the coordinates of the driver's face.
Eye Classifier: Scans only the *Region of Interest (ROI)* within the face to find the eyes

4. Mathematical Analysis (EAR Logic)
Normal State: The EAR value remains steady (around 0.25 to 0.30).
Drowsy State: The EAR value drops significantly (below 0.20) as the eyelids close.

5. Decision Making & Output
If the EAR is low for a *short* time: It is treated as a blink; the counter resets.
If the EAR is low for a *long* time (e.g., 20+ consecutive frames): The system classifies it as drowsiness and triggers the Multi-threaded winsound alarm.

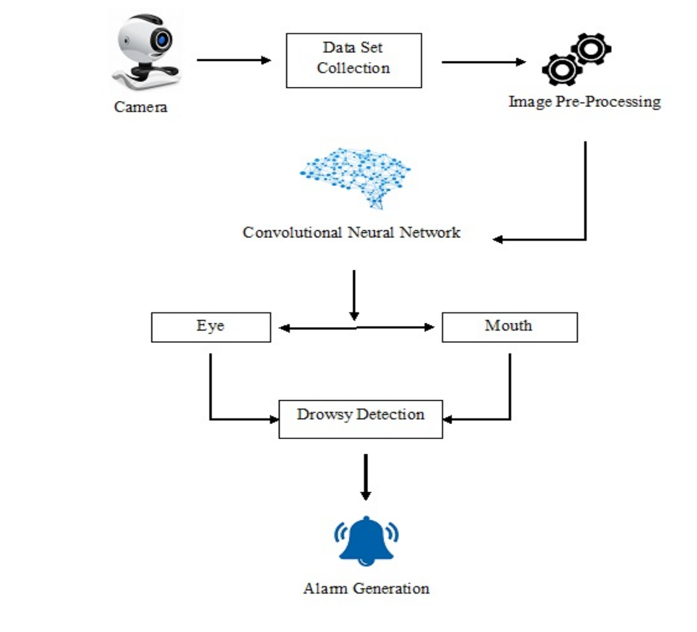

Fig. 1 System Architecture

V. Figures and Tables

TABLE I. Hardware Specifications

Components	Recommended Specifications
Processor (CPU)	Intel Core i3 (5th Gen) or i5 or higher
RAM	4 GB (Minimum); 8 GB (Recommended).
Storage (SSD/HDD)	At least 500 MB of free space
Camera	Standard 720p HD Webcam

TABLE II. Software Specifications

Software	*Version/Specifications*
Operating System	*Windows 10/11, Linux or MacOS*
Python	*3.x*
IDE	*VS Code, PyCharm or Jupyter Notebook*
Libraries	*Open CV, SciPy, NumPy,Thread/Winsound*

RESULT:
When you run the project, the output consists of three main parts:

- ***Visual Output (GUI):** A window displays the live camera feed. You will see a green bounding box around your face and eyes. On-screen text dynamically updates to show the current **EAR value** and the status: **"Active"** (Normal) or **"DROWSY ALERT!"** (in red).*
- ***Auditory Output:** If your eyes remain closed for more than the set threshold (e.g., 20 frames), the system triggers a continuous beep sound using the winsound library to wake the driver.*
- ***Console Output:** The terminal logs real-time data, printing the EAR values for every frame, which is useful for debugging and calibration.*

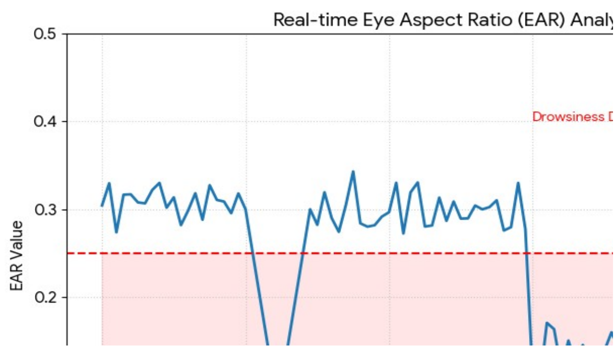

Fig: Grapgh representing output

Acknowledgment

The authors wish to express their sincere gratitude to the Department of Artificial Intelligence and Data Science at **Guru Gobind Singh College of Engineering and Research Center** for providing the laboratory facilities and computational resources necessary for this research. Special thanks are extended to our project guide, **[Insert Guide's Name]**, for their invaluable technical insights and continuous encouragement throughout the development of the vision-based monitoring system. Finally, we acknowledge the support of our peers and the institution for fostering an environment conducive to innovation in automotive safety technology.

REFERENCES

[1] Soukupová, T., & Čech, J. (2016). *Real-Time Eye Blink Detection using Facial Landmarks*. 21st Computer Vision Winter Workshop.

[2] Viola, P., & Jones, M. (2001). *Rapid Object Detection using a Boosted Cascade of Simple Features*. Proceedings of the 2001 IEEE Computer Vision and Pattern Recognition (CVPR).

[3] Bradski, G. (2000). *The OpenCV Library*. Dr. Dobb's Journal of Software Tools.