IoT-Based Real-Time Drowsiness Detection and Alert System for Driver Safety

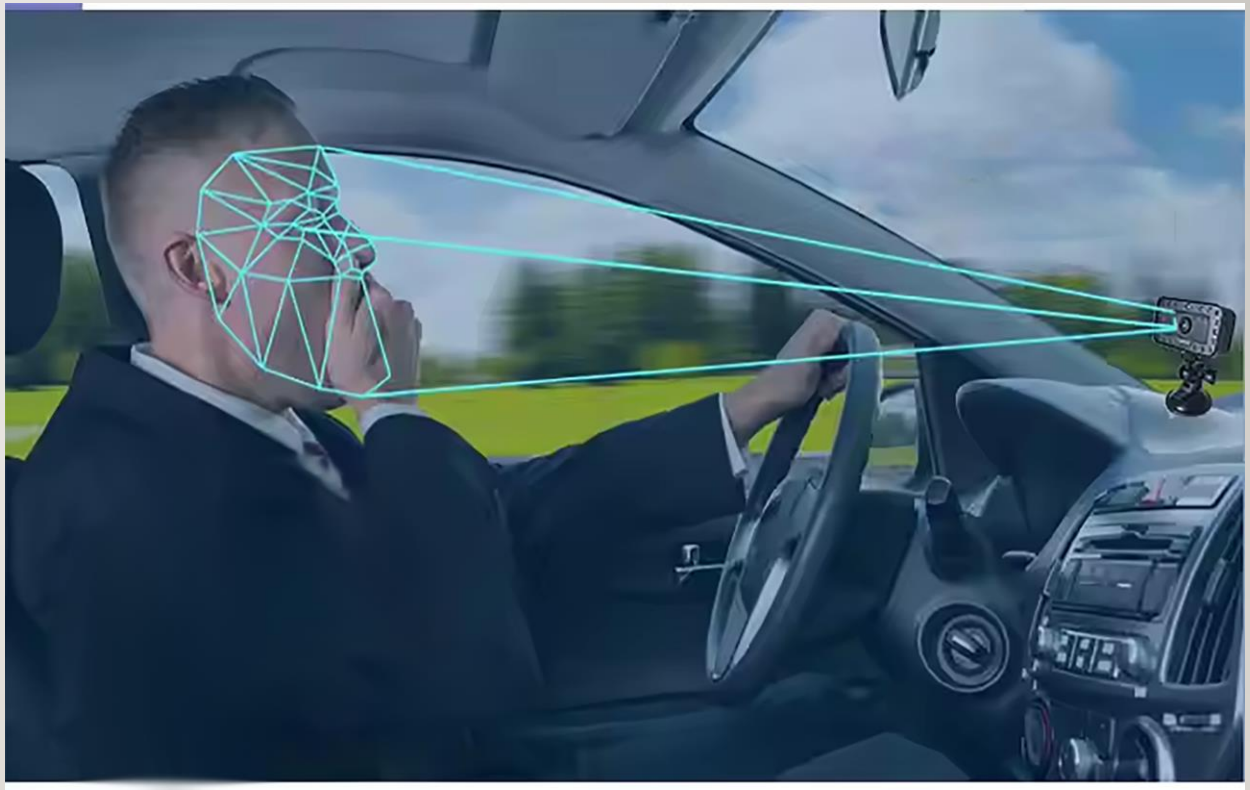
Name : **Ashish Goyal**
Roll No : **22PHC1R13**

Supervisor: **Prof. Aalu Boda**
Assistant Professor
Department of Physics, NIT Warangal

Content

Introduction

- Drowsy driving is a major cause of road accidents.
- Over 100,000 crashes per year are caused by sleepy drivers.
- Fatigue reduces reaction time and decision-making ability.
- Drivers often don’t realize they’re tired until it’s too late.



Introduction and Motivation

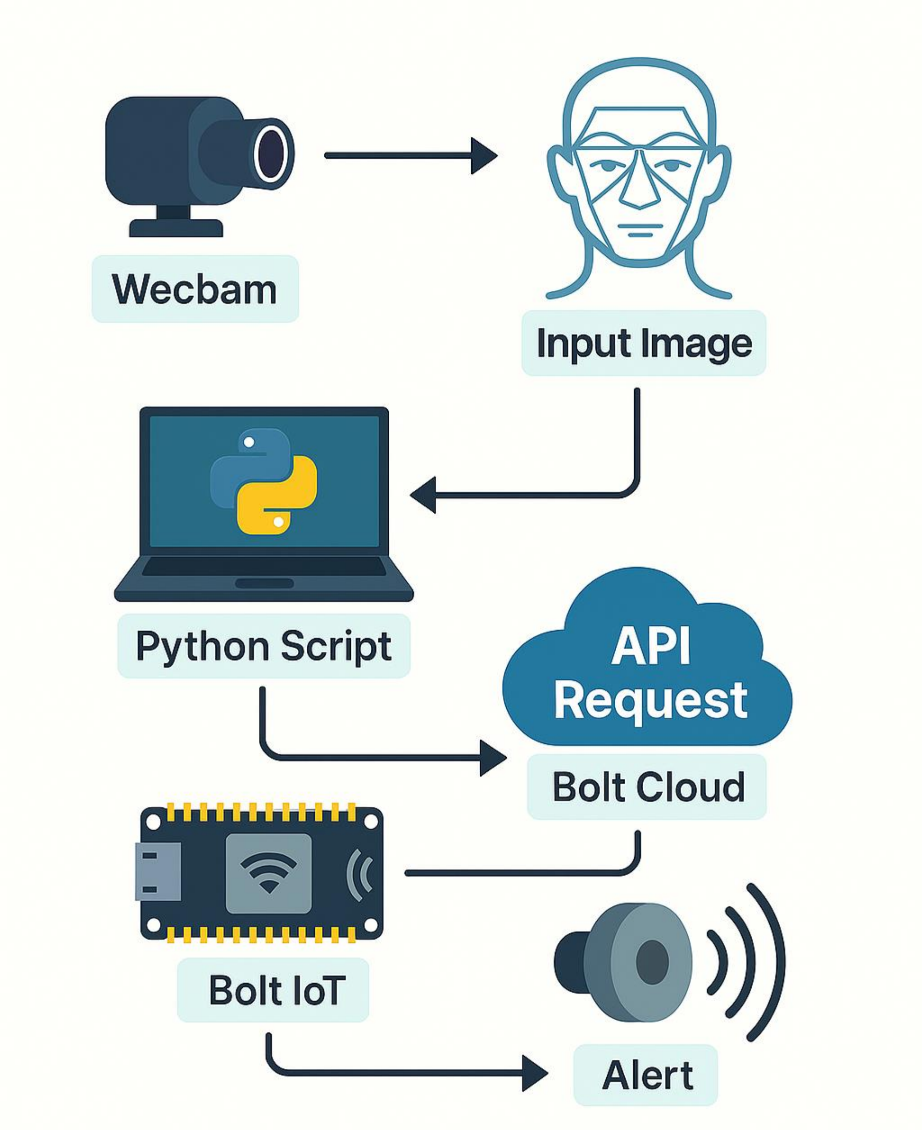
- Most drowsiness detection systems are costly or limited to luxury cars.
- There is no affordable, real-time solution for common vehicles.
- Using computer vision + IoT offers a low-cost and smart alternative.
- Our system detects eye closure and alerts the driver instantly.

Objectives

- To detect driver drowsiness in real time using a webcam.
- To use Eye Aspect Ratio (EAR) for identifying eye closure.
- To send a alert signal using the Bolt IoT module.
- To ensure reliable performance under different lighting conditions.
- To build a low-cost, non-intrusive, and scalable system.

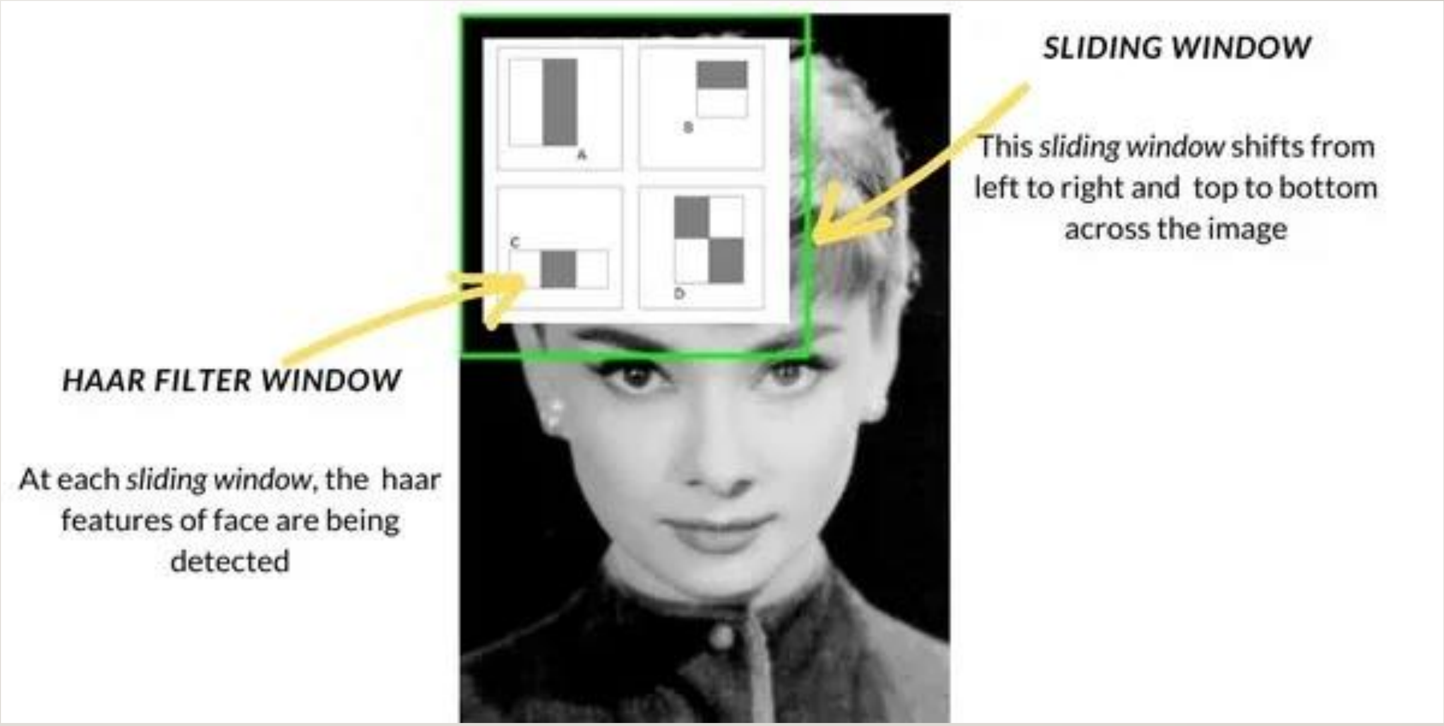
System Workflow

- Webcam captures live video of the driver's face.
- MediaPipe detects facial landmarks and locates the eyes.
- Python script calculates Eye Aspect Ratio (EAR).
- If EAR is below threshold for a set time → drowsiness detected.
- Python sends API request to Bolt Cloud.
- Bolt IoT triggers a buzzer alert to wake the driver.



How the System Detects Eyes from an Image

- **Image Processing**: The captured image is flipped horizontally for selfie-view display and converted from **BGR to RGB** format for compatibility with MediaPipe.
- **Face Detection using Haar Cascade:** A **sliding window** scans the image from left to right, top to bottom, checking small sections.
- **Pattern Matching**: The algorithm uses **Haar-like features**, simple patterns of light and dark to identify regions that look like human faces.



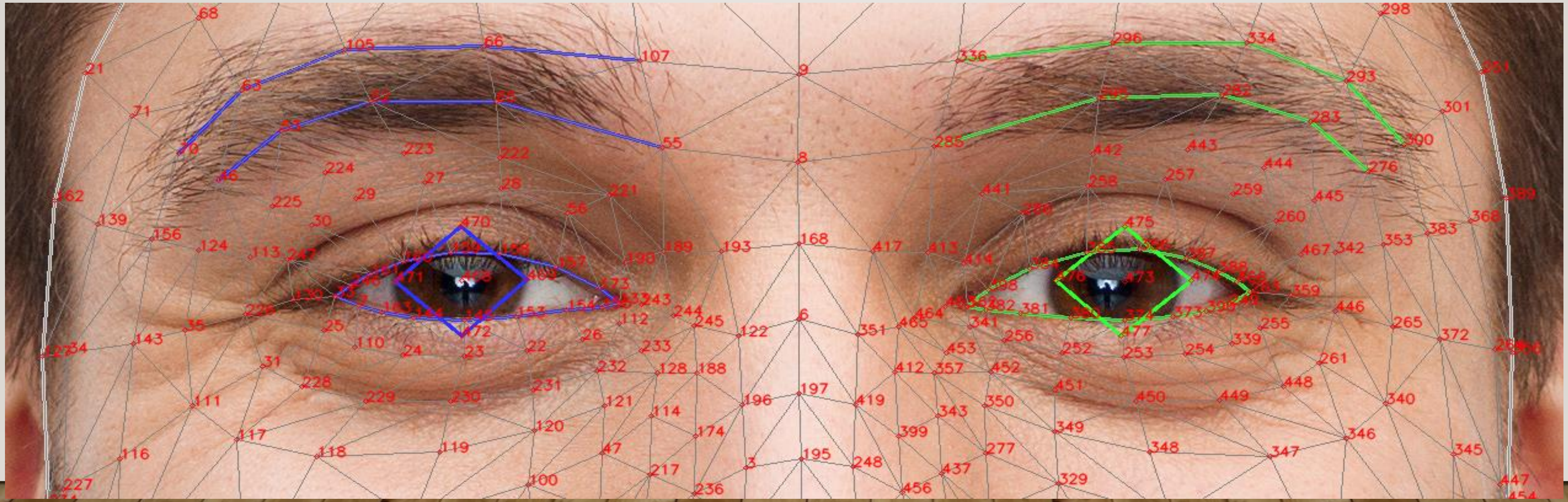
How the System Detects Eyes from an Image

Eye Localization: Once a face is detected, the algorithm focuses on the **eye regions**, extracting landmarks for both eyes using **MediaPipe FaceMesh**.

Landmark Points

- For the **left eye**: [362, 385, 387, 263, 373, 380]
- For the **Right eye**: [33, 160, 158, 133, 153, 144]

The chosen landmark points are in order: $\boldsymbol{P_1, P_2, P_3, P_4, P_5, P_6}$

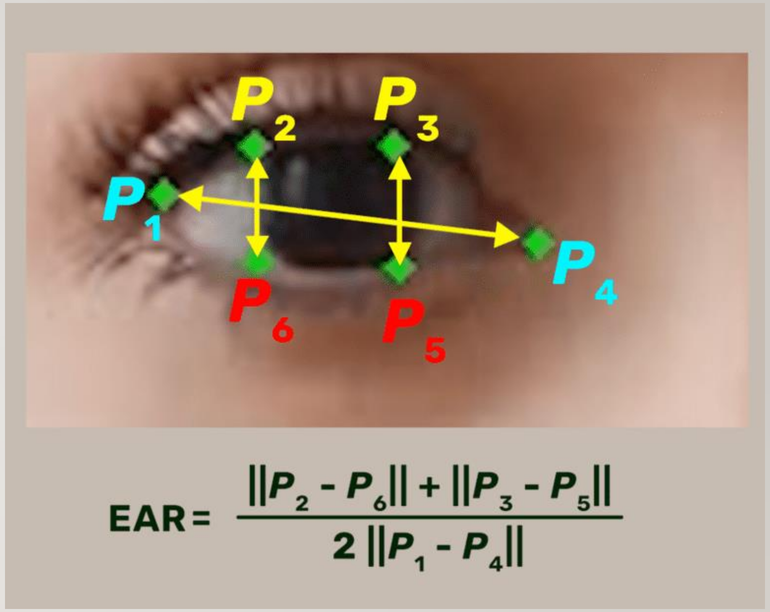


EAR-Based Drowsiness Detection

- **EAR** stands for **Eye Aspect Ratio** a measure of eye openness.
- It is calculated using 6 key landmark points around each eye.
- **Formula:**

  EAR = (‖p2 - p6‖ + ‖p3 - p5‖) / (2 × ‖p1 - p4‖)

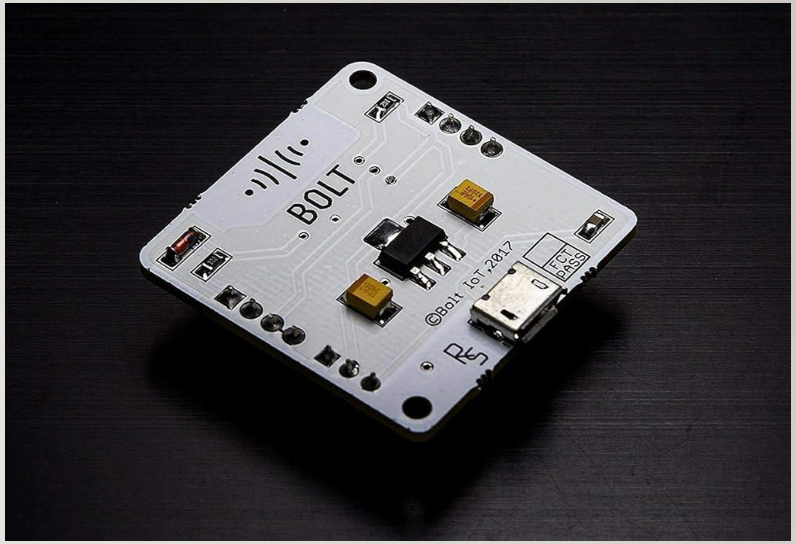
- It is **resolution-independent** and works consistently across different lighting and camera quality.



Hardware Overview – Bolt IoT Device

- The **Bolt IoT module** is a **Wi-Fi-enabled microcontroller** that helps build smart devices by connecting hardware sensors or components (like buzzers, LEDs, etc.) to the **internet or cloud applications**.

- In our **drowsiness detection system**, the Bolt IoT module acts as the **hardware controller** that receives a signal (from a Python script) and triggers a **buzzer or alert** if the driver is detected to be drowsy.



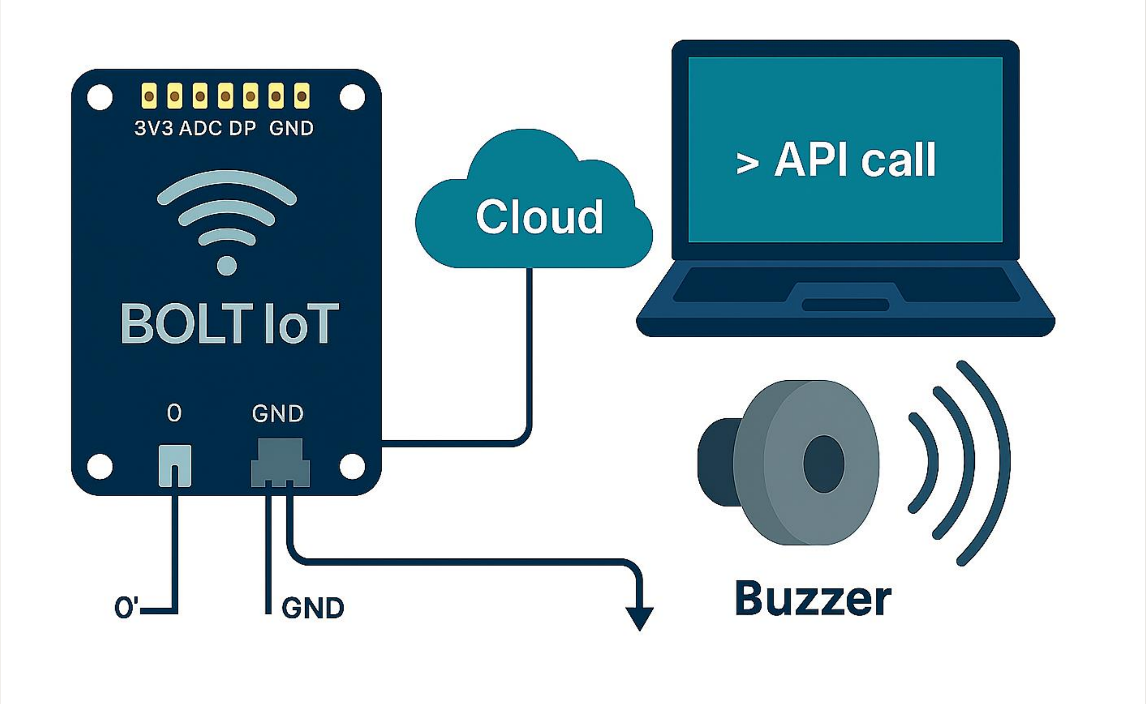
Hardware Overview – Bolt IoT Device

- **Wi-Fi Enabled**: Built on the ESP8266 chip for easy internet connectivity.
- **Cloud Connected**: Integrated with Bolt Cloud for remote monitoring and control.
- **API Support**: Allows RESTful API calls to send signals and control hardware.
- **Real-Time Alerts**: Triggers buzzer alerts instantly during drowsiness detection.
- **Easy Integration**: Works with Python, JavaScript, Arduino for fast development.

Microcontroller	Operating Voltage	Connectivity	GPIO Pins	Programming Interface	Compatible Languages	Cloud Platform
ESP8266	3.3V – 5V DC	802.11 b/g/n Wi-Fi	5 Digital I/O Pins	UART (via USB or TX/RX pins)	Python, JavaScript, Arduino	Bolt Cloud (API-based Control)

Circuit Design – Buzzer Alert System

- Buzzer is connected to Bolt IoT GPIO pin (Digital Pin 0).
- Powered through Bolt's **micro-USB power source**.
- Python script triggers a **digital HIGH signal** via Bolt Cloud API.
- Buzzer activates for few seconds to alert the driver.
- Safe, compact, and works wirelessly through Wi-Fi.



EAR Threshold Tuning – Experimental Evaluation

- Different EAR values were tested to find the **most reliable threshold**.
- Higher thresholds (e.g. 0.25) triggered too many **false alerts** (e.g. blinking).
- Lower thresholds (e.g. 0.16) caused **missed detections** (driver's eyes closed but not detected).

EAR Threshold	False Positives	Missed Detections	Accuracy (%)	Remarks
0.25	5	0	80%	Over-sensitive to blinking
0.23	3	1	85%	Moderately stable
0.20	**1**	**0**	**95%**	✅ **Optimal threshold**
0.18	0	2	80%	Tends to miss subtle closures
0.16	0	4	70%	Too strict, lowers sensitivity

- EAR = 0.20 is selected as the **optimal value** for real-time eye closure detection due to its balance between accuracy and stability.

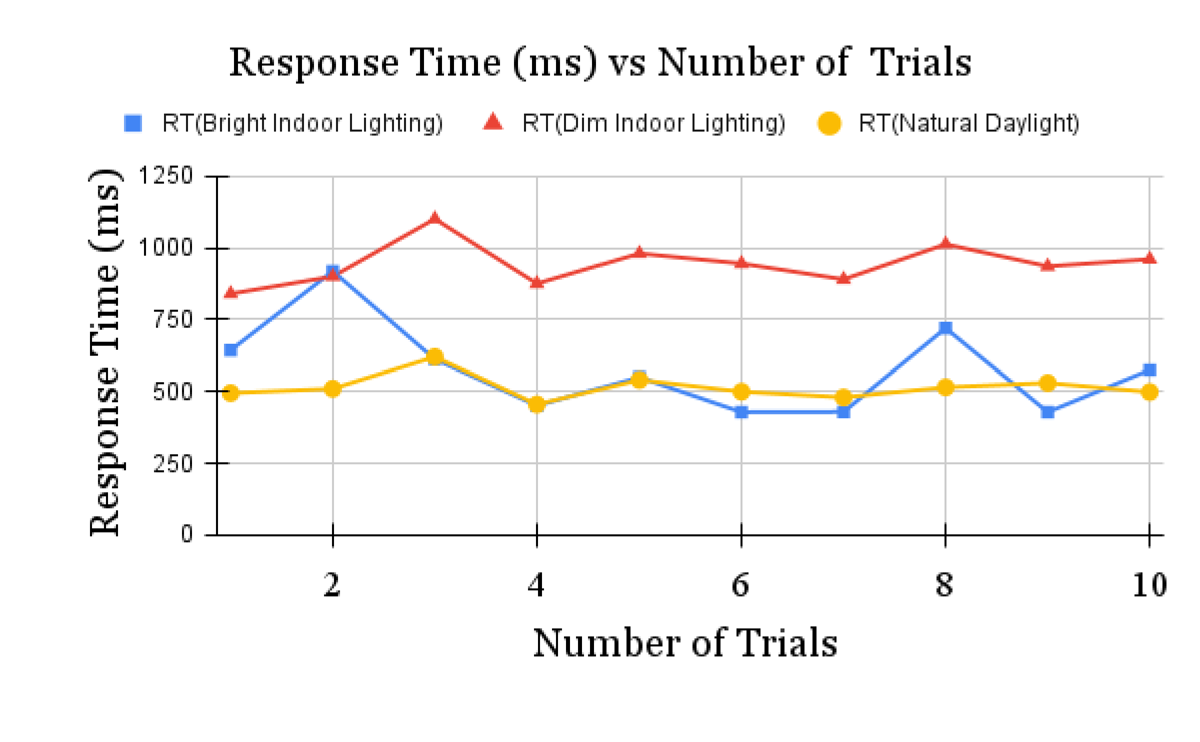
Lighting & Performance Analysis – Experimental Results

- System was tested under **three lighting conditions** to verify reliability.
- **Response time** measures how fast the buzzer activates after drowsiness is detected.
- Accuracy was calculated based on correctly identified drowsy vs. non-drowsy frames.

Lighting	Avg Time (ms)	Min	Max	Accuracy	Remarks
Bright Indoor	573	429	918	95%	Reliable
Dim Light	945	840	1100	88%	Slight delay
Daylight	517	455	620	93%	Best Stable

Lighting & Performance Analysis – Experimental Results

- For each, 10 trials were recorded measuring **response time and accuracy**.
- **Average, minimum, and maximum response times** were calculated.
- Natural Daylight lighting provided **fastest and most accurate** results.



Conclusions

- The proposed system successfully detects **driver drowsiness** using the **Eye Aspect Ratio (EAR)** method.
- Real-time alerts are triggered using the **Bolt IoT platform**, ensuring immediate response.
- Testing under different lighting conditions proved the system to be **accurate and consistent**, with up to **95% accuracy** in bright environments.
- The setup is **cost-effective**, **scalable**, and suitable for real-time applications in daily-use vehicles.
- This solution can be further enhanced by adding detection of **eye closure duration tracking**.
- Future work can involve developing a **mobile monitoring app** and connecting with **cloud dashboards** for advanced alert systems.

References

- T. Soukupová and J. Čech, *“Real-Time Eye Blink Detection using Facial Landmarks,”* Proc. of the 21st Computer Vision Winter Workshop, 2016.

- Google Developers, *MediaPipe Face Mesh Documentation*. Available at: https://google.github.io/mediapipe/solutions/face_mesh

- Bolt IoT Documentation, *Getting Started with Bolt Cloud and API*.
  Available at: https://docs.boltiot.com

- A. Bhandari and A. Rane, *“Drowsiness Detection Using Computer Vision Techniques,”* IJSREM, vol. 6, no. 1, 2022.

- National Highway Traffic Safety Administration (NHTSA), *“Drowsy Driving Statistics,”* https://www.nhtsa.gov

Acknowledgement

- I am deeply thankful to my **guide, Prof. Aalu Boda**, for his expert advice, technical support, and for providing the opportunity to work on this advanced topic.

- Special thanks to the faculty and staff of the **Department of Physics** for their support, and to all those who contributed directly or indirectly to the successful completion of this work.

- Lastly, I would like to thank my **family and friends** for their motivation and continuous support throughout the journey.

Thank You!