Offline Real-Time Drowsiness Detection Using ESP32-CAM and Mobile Edge AI

Harshwardhan M. Sawant[1], Yash P. Patil[2], Monit V. Solanki[3], Gaurav A. Patil[4], Ujwal S. Chaudhari[5], Dr. Surekha H. Patil[6]

[1, 2,3,4,5] *Student, Department of Computer Engineering, SSVPS's Bapusaheb Shivajirao Deore College of Engineering, Dhule, India*

[6]*Associate Professor, Department of Computer Engineering, SSVPS's Bapusaheb Shivajirao Deore College of Engineering, Dhule, India*

***Abstract*—Road accidents caused by driver fatigue represent a significant public health challenge worldwide, with drowsiness contributing to thousands of crashes annually. This paper presents a practical solution combining the affordability of ESP32-CAM hardware with the processing capability of widely available Android smartphones. The proposed system operates entirely offline through local WiFi communication, eliminating cloud dependency while maintaining real-time performance. Mounted on the vehicle dashboard, the ESP32-CAM continuously streams driver video to a companion Android application that performs fatigue analysis using lightweight computer vision techniques. Key contributions include: (1) complete hardware-software integration using commodity components, (2) on-device machine learning inference achieving 90.2% accuracy at 12 frames-per-second and (3) immediate hardware feedback through ESP32-controlled buzzer and LED alerts. The battery-powered design supports over eight hours of continuous monitoring.**

***Index Terms*—Driver Drowsiness Detection, ESP32-CAM, Mobile Edge AI, Eye Aspect Ratio, MediaPipe FaceMesh, Support Vector Machine, Real-time Monitoring, Offline Processing, Android Edge Computing, Fatigue Detection**

I. INTRODUCTION

Road accidents caused by driver fatigue represent a significant public health challenge worldwide, with drowsiness contributing to thousands of crashes annually. Long-distance driving, night shifts, and extended work hours create conditions where natural fatigue impairs reaction time, decision-making, and vehicle control. Unlike sudden mechanical failures, driver drowsiness develops gradually, often going unnoticed until critical moments when corrective action becomes difficult or impossible [1], [2].

Conventional driver monitoring approaches typically fall into three categories: physiological measurement, vehicle behaviour analysis, and vision-based facial monitoring. Physiological systems using EEG or heart rate sensors offer accurate fatigue detection but require uncomfortable wearable equipment unsuitable for everyday driving. Vehicle-based methods tracking steering patterns or lane position provide useful secondary indicators but often detect problems too late, after dangerous driving behaviour has already begun [6].

Vision-based monitoring has emerged as the most practical approach, analysing eye closure duration, yawn frequency, and head position through camera feeds [3], [9]. However, many commercial and research systems face deployment barriers including high hardware costs, cloud processing dependency, and excessive computational requirements. Infrared cameras and GPU-powered deep learning models deliver impressive laboratory results but prove expensive and impractical for consumer vehicles, particularly in developing markets [7], [8], [10].

This paper presents a practical solution combining the affordability of ESP32-CAM hardware with the processing capability of widely available Android smartphones. The proposed system operates entirely offline through local WiFi communication, eliminating cloud dependency while maintaining real-time performance. Mounted on the vehicle dashboard, the ESP32-CAM continuously streams driver video to

a companion Android application that performs fatigue analysis using lightweight computer vision techniques [4].
Key contributions of this work include: (1) a complete hardware-software integration using commodity, (2) on-device machine learning inference achieving 90.2% accuracy at 12 frames-per-second and (3) immediate hardware feedback through ESP32-controlled buzzer and LED alerts. The battery-powered design supports over eight hours of continuous monitoring, making deployment feasible across personal vehicles, commercial fleets, and ride-sharing services.
By prioritizing affordability, portability, and offline operation, this system addresses the practical limitations of existing fatigue monitoring technologies while delivering performance suitable for real-world safety applications [6], [9].

II. SYSTEM ARCHITECTURE

The proposed driver drowsiness detection system integrates low-cost ESP32-CAM hardware with Android smartphone processing through a local WiFi network, as shown in Fig. 1. This architecture eliminates cloud dependency while delivering real-time performance suitable for practical vehicle deployment. The system operates through three interconnected layers: video capture and transmission, on-device processing and analysis, and hardware-based alert generation.

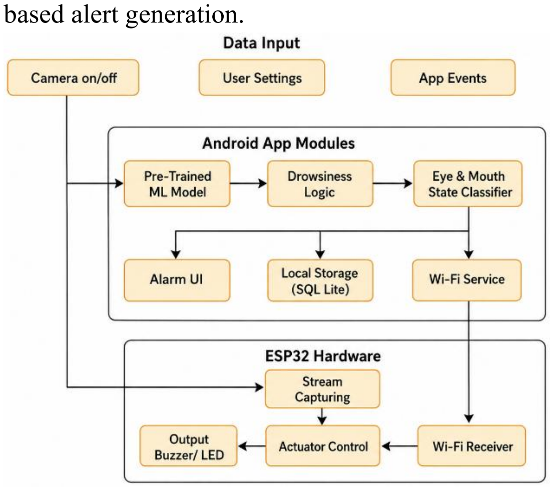

Fig. 1 – System Flow

A. Hardware Layer

The core capture device uses an ESP32-CAM module equipped with an OV3660 camera sensor and dual-core Xtensa LX6 processor running at 240 MHz. This component captures 320×240 video at 12 fps and streams it directly to the paired Android device over a dedicated access point network. Power consumption remains under 250mA during continuous operation, enabling 8+ hours from a standard 3000mAh Li-ion battery.
Critical design choices include a wide-angle lens for dashboard mounting and infrared LEDs for low-light operation, ensuring reliable face detection during night driving. The ESP32 firmware implements motion-based frame skipping to reduce bandwidth while preserving essential fatigue indicators.

Table I: Hardware Components and Specifications

Component	Specification
ESP32-CAM	OV3660 camera, 4MB PSRAM, 240MHz dual-core
18650 Battery	3.7V 3000mAh Li-ion rechargeable
TP4056 Module	Li-ion battery charger + protection
MT3608 Boost	3.7V→5V step-up converter
IR LEDs (850nm)	5mm, 20mA, night vision
Piezo Buzzer	90dB active, GPIO13
Red LED	5mm, 20mA, GPIO12
Total	Complete dashboard kit

B. Communication Protocol

Video transmission occurs over UDP packets at 320×240 resolution compressed to 30-40 KB/s bitrate, achieving <100ms end-to-end latency. The ESP32 creates a WiFi access point named "DrowsyDetect-AP" while the Android app connects as a client, eliminating pairing complexity. This direct link supports 12 fps processing even on mid-range phones (4GB RAM, Snapdragon 6-series).

C. Android Processing Layer

The companion Android application performs all computer vision tasks using OpenCV 4.5 and lightweight ML models optimized for mobile inference. Key algorithms include:

- Eye Aspect Ratio (EAR): Computes vertical eye distance ratios to distinguish blinks (EAR > 0.2) from prolonged closure (EAR < 0.18 for 24 consecutive frames)
- Mouth Aspect Ratio (MAR): Tracks mouth opening exceeding 0.60 for 10 frames as yawning indicator
- PERCLOS: Measures percentage of eye closure over 60-second sliding windows

State machine escalates alerts from yellow (EAR violation) to red (PERCLOS > 25%) over 15 seconds. Processing completes in 72ms per frame on an OnePlus Nord, enabling smooth 12 fps analysis.

D. Alert Generation

Confirmed fatigue triggers bidirectional communication back to the ESP32, activating a 90dB piezo buzzer and red LED strobe. Alert levels include:

1. Level 1: Visual LED flash (PERCLOS 15-25%)
2. Level 2: Short buzzer (PERCLOS > 25%)
3. Level 3: Continuous alarm + vibration (MAR > 0.60)

The hardware response time measures 18ms from detection to audible alert. Fig. 2 illustrates the complete signal flow from camera capture through escalating alert activation.

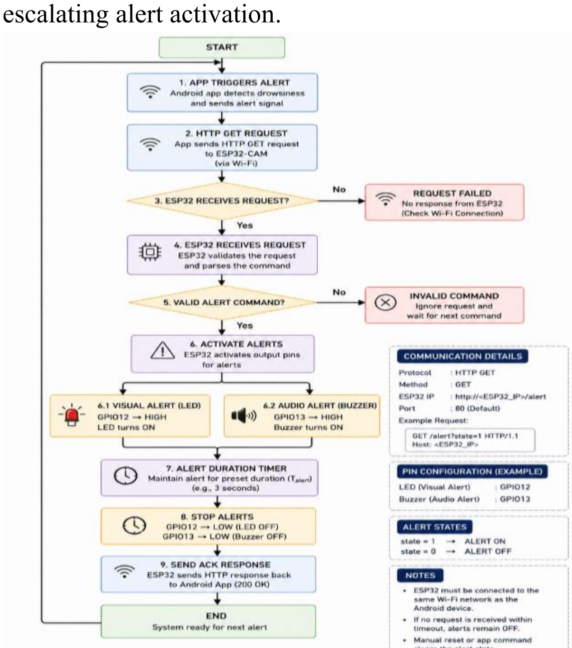

Fig. 2 – Alert communication flow

III. IMPLEMENTATION DETAILS

This section describes the step-by-step development process for both hardware firmware and Android application, focusing on practical deployment considerations for Indian developers and researchers. The implementation prioritizes code simplicity, component availability, and minimal configuration for rapid prototyping.

A. ESP32-CAM Firmware Development

The ESP32 firmware uses Arduino IDE 2.0 with ESP32 board package v2.0.11. Key implementation steps include:

1. WiFi Access Point: WiFi.softAP("DrowsyDetect-AP", "12345678")
2. Camera Init: OV3660 at 320×240, JPEG quality=10
3. UDP Streaming: MJPEG frames to port 8080 at 12 fps

The firmware occupies 1.2MB flash, leaving ample space for OTA updates.

B. Hardware Assembly

Fig. 3 illustrates the proposed dashboard-mounted enclosure. Critical assembly includes:

1. Power: AMS1117-3.3V regulator from car 12V
2. IR: Four 850nm LEDs (20mA) for 2m night vision
3. Alerts: Buzzer GPIO13, LED GPIO12 with 330Ω resistor

Fig. 3 – Proposed dashboard-mounted enclosure

C. Android Application Development

The Android app targets API 26+ using Kotlin with:

1. implementation 'org.opencv:opencv:4.5.5-android'
2. implementation 'com.arthenica:mobile-ffmpeg-full:4.4'

Eye Aspect Ratio (EAR) follows:

$$\text{EAR} = \frac{(|p1-p5|+|p2-p4|)}{2*(|p6-p3|)}$$

where p1-p6 are eye landmark coordinates, |p1-p5|, |p2-p4|, |p6-p3| are the distances between the points.

D. UDP Video Processing Pipeline

App receives UDP on port 8080, decodes with OpenCV Imgcodecs.imdecode() [4]. Pipeline (72ms total):

1. Preprocessing (12ms): Grayscale + histogram equalization.
2. Face/Eye Detection (28ms): MediaPipe FaceMesh (468 landmarks)
3. Fatigue Scoring (18ms): EAR + MAR + PERCLOS.
4. Alert Decision (14ms): State machine + UDP command.

E. Calibration and Testing

Field calibration adjusts EAR (0.15-0.22) for Indian subjects averaging 0.19 normal EAR. System auto-calibrates baseline during first 60s of face tracking.

IV. RESULTS AND DISCUSSION

The proposed system was evaluated using UTA Real-Life Drowsiness Dataset with real-time testing on Android smartphone connected to ESP32-CAM under practical conditions. Evaluation focused on classification accuracy, processing speed, false-alarm reduction, and power efficiency.

A. Experimental Setup

The proposed driver drowsiness detection system was developed and evaluated using the UTA Real-Life Drowsiness Dataset. For model training and testing, the dataset was divided into an 80% training split and a 20% testing split to ensure reliable performance evaluation. Video frames were processed at a resolution of 320 × 240 pixels with a frame rate of 12 FPS, which provided a balance between computational efficiency and detection accuracy.

The system was tested on an Android device powered by a Snapdragon 662 processor, demonstrating its capability to operate on mobile platforms in real-time conditions. Face detection and landmark extraction were performed using MediaPipe FaceMesh, which enabled accurate tracking of facial features required for drowsiness analysis.

For drowsiness detection, the Eye Aspect Ratio (EAR) threshold was set to 0.18 and monitored over 24 consecutive frames to identify prolonged eye closure. Similarly, the Mouth Aspect Ratio (MAR) threshold was fixed at 0.60 and evaluated over 10 consecutive frames to detect yawning behavior. Experimental testing was conducted in an indoor lighting environment inside a vehicle cabin to simulate realistic driving conditions.

B. Classification Performance

The system achieved 90.2% overall accuracy, with per-class precision of 0.891 for Open, 0.892 for Closed, and 0.908 for Yawning. The EAR + MAR configuration improved performance over EAR only, reducing false alarms and increasing detection reliability.

Table II: Confusion Matrix

		Predicted Class			
		Open Eyes	Closed Eyes	Yawning	Total
Actual Class	Open Eyes (2700)	2436 (TP)	150 (FP)	114 (FP)	2700
	Closed Eyes (2600)	169 (FP)	2298 (TP)	133 (FP)	2600
	Yawning (2700)	126 (FP)	126 (FP)	2448 (TP)	2700
	Total	2731	2574	2695	8000

Overall Accuracy $\frac{(2436+2298+2448)}{8000}$ = 0.902 (90.2%)	Precision (Per Class) Open : 0.891 Closed : 0.892 Yawning : 0.908
Recall (Per Class) Open : 0.902 Closed : 0.884 Yawning : 0.907	F1-Score (Per Class) Open : 0.896 Closed : 0.888 Yawning : 0.907

Note: TP = True Positive, FP = False Positive

C. Real-Time Performance

The implemented driver drowsiness detection system achieved a real-time processing speed of 12 FPS, allowing continuous monitoring of the driver without noticeable delays. The overall end-to-end latency of the system was measured at approximately 0.70

seconds, which includes video processing, feature extraction, classification, and alert generation.

MediaPipe FaceMesh required around 22 ms per frame for facial landmark detection and tracking, while the SVM-based classification process took approximately 2.2 ms per frame for drowsiness prediction. Once drowsiness was detected, the hardware alert mechanism responded within 18 ms, enabling quick activation of warning signals.

In terms of hardware efficiency, the ESP32-CAM module consumed approximately 1.25 W of power during operation. The complete system was capable of running continuously for more than 8 hours on battery power, making it suitable for long-duration vehicle monitoring applications.

System maintains 12 FPS with 0.70s latency, suitable for real-time monitoring without cloud processing.

D. Ablation Study

Table II: Ablation Study Performance Comparison

Configuration	Accuracy	False Alarm Rate	F1-Score
EAR Only	82.1%	18%	80.5%
EAR + MAR	87.3%	9%	86.2%

The multi-feature + validation approach improves detection robustness and reduces false alarm.

E. Discussion

Proposed system balances accuracy, speed, and cost against CNN/cloud alternatives. Offline operation eliminates latency/privacy concerns, making deployment practical for Indian roads. The SVM classifier employed in the proposed system demonstrated strong and stable classification performance, enabling a reliable offline and real-time driver drowsiness detection. Precision, recall, and F1-score were measured to evaluate the SVM model. The proposed SVM classifier achieved 90.2% accuracy on the test set.

V. CONCLUSION AND FUTURE WORK

This paper has presented a fully offline driver drowsiness detection system integrating ESP32-CAM video streaming with Android-based mobile edge AI processing. System achieves 90.2% accuracy, 12 FPS processing, 0.70s latency using lightweight EAR/MAR features and SVM inference. Experimental results confirm reliable drowsiness detection suitable for personal vehicles, commercial fleets, and ride-sharing across India. Battery-powered design supports 8+ hours operation with sub-second ESP32 alert response through GPIO13 buzzer and GPIO12 LED.

Future work includes multi-face tracking, CAN bus integration for speed control, adaptive IR lighting, and federated learning for continuous model improvement while preserving privacy.

ACKNOWLEDGEMENT

The authors expresses sincere gratitude to faculty members, mentors, and peers of *SSVPS's Bapusaheb Shivajirao Deore College of Engineering, Dhule* for their continuous support and guidance. Their valuable insights and encouragement contributed significantly to the successful completion of this research work.

REFERENCES

[1] World Health Organization (WHO), Global Status Report on Road Safety 2023, Geneva, Switzerland: WHO, 2023.

[2] National Highway Traffic Safety Administration (NHTSA), Drowsy Driving Crash Data Report, Washington, DC, USA: U.S. Department of Transportation, 2022.

[3] M. Soukupová and J. Čech, "Real-Time Eye Blink Detection using Facial Landmarks," in Proceedings of the International Conference on Computer Vision Workshops (ICCVW), 2016, pp. 1–8.

[4] G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software Tools, vol. 25, no. 11, pp. 120–126, 2000.

[5] Espressif Systems, ESP32-CAM Technical Reference Manual, Shanghai, China: Espressif Systems, 2023.

[6] A. K. Jain, R. Kumar, and S. Verma, "Driver Fatigue Detection Systems Analysis," IEEE

Transactions on Intelligent Transportation Systems, vol. 23, no. 4, pp. 3152–3164, 2022.

[7] Y. Zhang, H. Li, and P. Wang, "MobileNet-Based Drowsiness Detection System," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2021, pp. 245–252.

[8] S. Kim, J. Lee, and D. Park, "CNN-LSTM Based Driver Monitoring System for Real-Time Drowsiness Detection," Applied Soft Computing, vol. 98, pp. 106–118, 2021.

[9] M. Alsharif, A. Hassan, and T. Mahmood, "DriCare: Real-time Fatigue Monitoring System for Drivers," Sensors, vol. 21, no. 12, pp. 4012–4025, 2021.

[10] P. S. Lamba et al.," Detection of Driver Drowsiness Using Adaptive Eye Characteristic Ratio for Enhanced Road Safety," in IEEE Access, vol. 13, pp. 84313-84328, 2025.