iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 Issue: IV Month of publication: April 2026

DOI: https://doi.org/10.22214/ijraset.2026.80062

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Real-Time Driver Alertness and Assistance System

Ms. Priya Meshram[1], Karan Tulswani[2], Karan Parwani[3], Mahima Singh[4], Kamlesh Thote[5], Ayush Karmani[6], Koyal Rahangadale[7]
Department of Computer Science and Engineering, G.H Raisoni University Amravati, Maharashtra, India

Abstract: This research presents an integrated safety monitoring system designed to enhance road safety by analyzing driver behaviour and vehicle interactions in real-time. The system utilizes a multi- layered deep learning architecture to detect potential hazards and evaluate driving performance. For ex ternal environment monitoring, a YOLOv8 object detection model is employed to identify and track various vehicle classes, including cars, bikes, buses, and trucks. To specifically evaluate overtaking operations, we developed an Advanced Overtaking LSTM model. This bidirectional LSTM network processes temporal feature sequences including horizontal motion and area change ratios to classify operations as "Safe" or "Rash" with high precision. In addition to external monitoring, the system incorporates a secondary YOLOv11-based model trained on specialized datasets to monitor the internal driver state. This model detects critical signs of fatigue and distraction, such as drowsiness and yawning. The core innovation lies in the seamless fusion of these disparate data streams into a unified processing pipeline. By correlating external operation risks with the internal physiological state of the driver, the system calculates a dynamic Driver Score (0–100). This score categorizes behaviour into Safe, Moderate, or Aggressive profiles, allowing for highly personalized safety interventions. To ensure practical utility, the entire backend is integrated into a mobile application developed using the React Native framework, enabling cross-platform accessibility and low-latency feedback. This application provides real-time feedback and auditory alerts, ensuring that high-level safety monitoring is accessible to any driver with a smartphone. Experimental evaluations indicate that the system maintains high accuracy in complex traffic conditions, significantly reducing the gap between advanced driver assistance systems (ADAS) and everyday mobile-based safety tools.
Keywords: Driver Behaviour, YOLOv8, YOLO11, LSTM, Computer Vision, Road Safety, React Native, Real-time Monitoring.

I. INTRODUCTION

The rapid increase in vehicular density and long- distance commuting has made road safety a critical global concern. According to global health statistics, traffic accidents remain a leading cause of mortality, with a significant majority of these incidents attributed to human error. Among the primary factors, driver fatigue and aggressive operationing specifically risky overtaking stand out as the most hazardous behaviours. While modern luxury vehicles often feature advanced driver-assistance systems, a substantial portion of the global vehicle fleet remains without these safety features due to high installation costs and specialized hardware requirements. To address this disparity, this research proposes a comprehensive, low-cost safety monitoring solution that leverages deep learning and ubiquitous mobile technology. The project is built upon two distinct yet integrated detection modules. The first module focuses on the external environment, employing a YOLOv11 architecture to identify various vehicle classes such as cars, buses, and trucks. This is further enhanced by an Advanced Overtaking LSTM network that analyzes the temporal dynamics of vehicle movement, such as horizontal shift and area change, to classify operations as either safe or rash. Simultaneously, the second module monitors the internal driver state using a YOLOv11 model specifically trained for facial landmark analysis. This allows the system to detect physiological signs of impairment, including persistent yawning, eye closure, and signs of falling asleep. The innovation of this work lies in the integration of these dual-perspective data streams into a unified Driver Scoring algorithm. By deploying this backend via a React Native mobile application, the system transforms a standard smartphone into a sophisticated ADAS tool. This application provides real-time feedback and auditory alerts, ensuring that high level safety monitoring is accessible to any driver with a smartphone. The following other sections detail the literature survey methodology, the deep learning architectures employed, and the experimental results and conclusions achieved during system validation.

II. LITERATURE SURVEY

The field of driver safety monitoring has evolved through several distinct methodologies, ranging from physiological sensors to advanced computer vision techniques. A critical review of existing literature highlights the transition from hardware- dependent systems to non-intrusive, software-driven solutions.

1) Behavioural-Based Detection: Research by several authors emphasizes the use of behavioural characteristics as the most non-intrusive method for fatigue detection. Early systems focused primarily on eye-closure rates and blinking frequency. Studies such as Driver Drowsiness Detection and Alert System" and Drowsiness and Yawning Detection" demonstrate that facial landmark analysis can effectively identify yawning and prolonged eye closure (PERCLOS). These works establish that behavioural metrics provide a reliable proxy for alertness without requiring the driver to wear sensors, which is a core principle adopted in our current YOLO11-based internal monitoring module.
2) Evolution of Machine Learning Models: The shift toward deep learning is well-documented in recent reviews of the field. A Review of Recent Developments in Driver Drowsiness Detection Systems" categorizes the progress from traditional image processing to Convolutional Neural Networks (CNNs). Furthermore, the work titled Drowsiness Detection System in Real Time Based on Behavioural Characteristics of Driver using Machine Learning Approach" explores the trade- offs between accuracy and processing speed. These studies justify the transition to real-time object detection models like YOLO, which offer the high-frame-rate pro cessing necessary for a mobile-based environment.
3) Vehicle Dynamics and OBD Integration: Beyond monitoring the driver's face, analyzing the vehicle's movement provides an additional layer of safety. The research "Driving Behaviour Analysis and Classification by Vehicle OBD Data Using Machine Learning" explores how On-Board Diagnostics (OBD) and vehicle motion data can classify driving styles. This literature highlights that steering wheel patterns and speed fluctuations are strong indicators of either fatigue or aggressive behaviour. Our research builds upon this by using an LSTM-based approach to analyze external operationing patterns such as overtaking rather than relying solely on internal sensors, thereby providing a more holistic view of road safety.
4) Synthesis and Gap Analysis: While existing literature provides robust frameworks for either internal drowsiness detection or external vehicle analysis, there is a notable gap in systems that integrate both perspectives into a single, unified scoring metric accessible via consumer-grade mobile hardware. Most reviewed systems require high-end embedded GPU clusters or specialized OBD-II hardware. Our project addresses this by fusing the behavioural insights found in literature with real-time operation analysis, delivered through a cross-platform React Native application.

III. PROPOSED METHODOLOGY

The proposed system architecture is divided into three primary modules: the External Environment Analyzer, the Internal Driver Monitor, and the Mobile Integration Layer. This multi-modal approach ensures that safety is assessed from both a behavioural and a situational perspective.

A. *External Environment and Overtaking Analysis*

The system utilizes a dual-model approach for external safety monitoring. First, a YOLOv11 object detection model is deployed to identify and track four key vehicle classes: cars, bikes, buses, and trucks. To maintain temporal consistency, we integrated the deep sort tracking algorithm, which assigns unique IDs to vehicles across frames. For the classification of overtaking operations, we developed a specialized Advanced Overtaking LSTM network. The input features for this model are derived from the bounding box dynamics:

- dx: The horizontal shift of the vehicle relative to the camera frame.
- darea: The rate of change in the bounding box area, serving as a proxy for the relative distance and speed of the vehicle.

These sequences are processed through a bidirectional LSTM layer with 192 hidden units and a 40% dropout rate to prevent overfitting. The model outputs a probability distribution classifying the operation as either Safe" or Rash."

B. *Internal Driver State Monitoring*

To detect impairment, the system employs a YOLO11 model trained on a comprehensive drowsiness dataset. Unlike traditional image processing, this model is trained to recognize specific facial states directly:

- Fatigue Indicators: Detection of Drowsy eye," Yawn," and Asleep" states.
- Attentiveness Indicators: Monitoring for Attentive eye" versus Closed" or No Yawn" states.

The model achieves high mean Average Precision (mAP) by analyzing facial landmarks in real-time, providing a continuous stream of alertness data to the central processing unit.

C. Data Fusion and Driver Scoring

The innovation of our methodology lies in the Driver Scoring Algorithm. Instead of treating events in isolation, the system maintains a dynamic score $S \in [0,100]$. The initial score is 100 and is modified based on detected events:

$$S_{new} = S_{old} - (Penalty_{Rash} \times N) - (Penalty_{Drowsy} \times T)$$

where N represents the number of rash Operations and T represents the duration of detected drowsiness. The final score is categorized into three safety profiles:

- Safe (Score ≥ 85): High alertness and cautious operationing.
- Moderate (60 ≤ Score < 85): Occasional lapses or aggressive shifts.
- Aggressive (Score < 60): High-risk driving requiring immediate intervention.

D. React Native Implementation

To bridge the gap between complex deep learning models and practical use, the backend is integrated into a mobile application using React Native. This framework allows the system to access the smartphone camera as the primary sensor. The app serves as the UI for real-time visual overlays, displaying the current Driver Score and triggering auditory alerts when the score drops below critical thresholds or when the YOLO11 model detects an "Asleep" state.

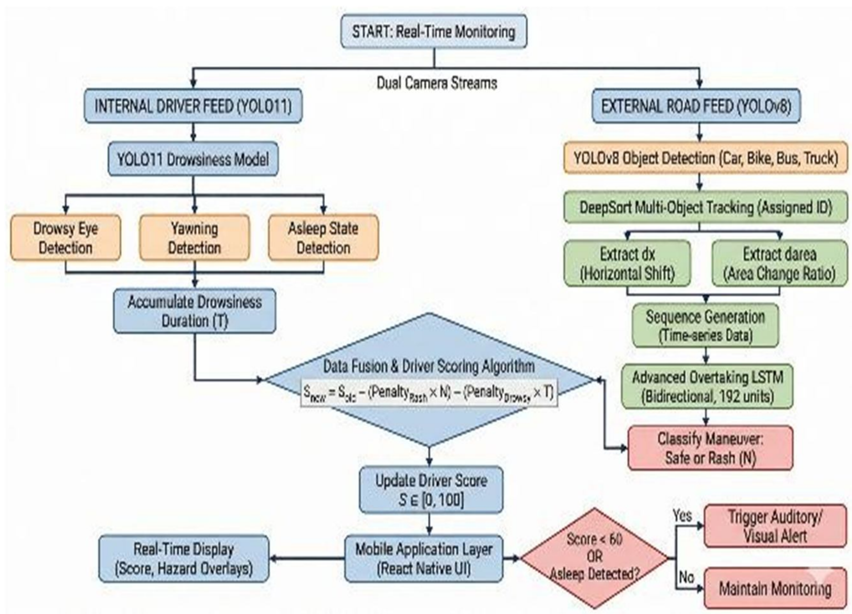


Figure 1: Proposed system architecture illustrating the integration of dual-stream deep learning models and the mobile interface.

IV. SYSTEM SETUP

The implementation of the Real-Time Driver Alertness and Assistance System involves a hybrid environment consisting of a high-performance training setup and a mobile-optimized deployment framework.

A. Hardware and Software Environment

The technical specifications for the development and deployment phases are summarized in Table 1. The deep learning models were trained using high- compute GPU resources to handle the extensive video datasets, while the inference is optimized for smartphone hardware.

B. Development Setup

The backend was developed using Python as the core language. The Ultralytics library was utilized for implementing the YOLOv8 (external) and YOLO11 (internal) pipelines. For temporal analysis, the Advanced Overtaking LSTM was built using PyTorch, incorporating 192 hidden units and a 0.4 dropout rate. Multi-object tracking was managed via the DeepSort algorithm to ensure consistent vehicle ID assignment across road-feed frames.

TABLE 1: System Specifications

Component	Specification
Operating System	Android 9.0+ / iOS 13.0+
CPU/GPU (Training)	NVIDIA T4 GPU
Frameworks	React Native, PyTorch, Ultralytics
Detection Models	YOLOv8, YOLO11, DeepSort
Classification Model	Bidirectional LSTM (192 Units)
Programming Language	Python 3.10, JavaScript (ES6)
Libraries	OpenCV, Pandas, NumPy, TorchVision

C. Deployment Setup

The frontend mobile application was architected using React Native, ensuring a unified codebase for cross-platform utility. The application communicates with the deep learning backend to provide real-time visual overlays and auditory alerts. The system is designed to run locally on the mobile device, leveraging the smartphone's NPU/GPU to maintain a low-latency feedback loop for the driver.

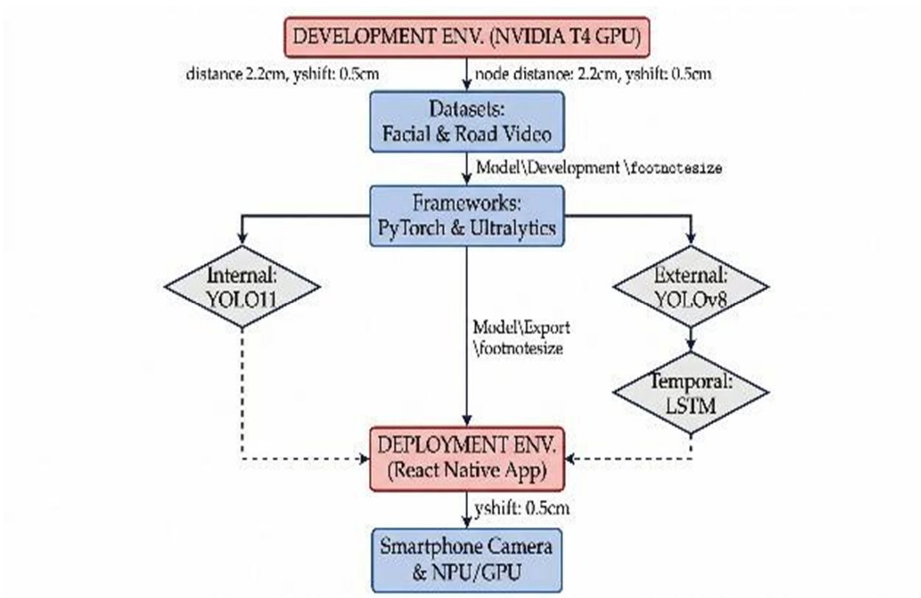


Figure 2: System setup diagram illustrating the relationship between the GPU-accelerated development environment, the specific deep learning models, and the final React Native deployment on a smartphone.

V. DETAILED EVALUATION CRITERIA

The performance of the proposed system is validated through a combination of standard object detection bench marks and temporal sequence classification metrics. The evaluation focuses on the system's ability to minimize false negatives in safety-critical scenarios.

A. Object Detection and Drowsiness Metrics

For the internal (YOLO11) and external (YOLOv8) monitoring modules, we utilize mean Average Precision (mAP) and Recall (R). Recall is particularly vital for driver safety to ensure that signs of impairment are not missed. The metrics are defined as:

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

$$Precision(P) = \frac{TP}{TP + FP}$$

$$Recall(R) = \frac{TP}{TP + FN}$$

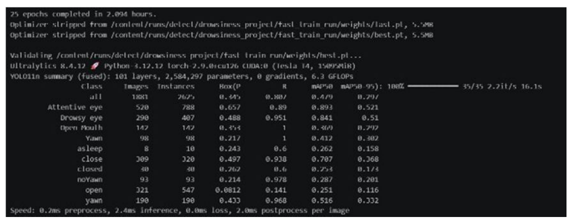
where TP is True Positives, FP is False Positives, and
FN is False Negatives. Based on our experimental results, the internal model achieved an overall **Recall of 0.807** and the **mAP@50 of 0.479**, with the "Drowsy eye" class reaching an individual mAP of 0.841.

B. *Classification Accuracy:*

The Advanced Overtaking LSTM is evaluated using the F1-Score to account for the class imbalance between frequent safe operations and infrequent rash operations:

$$F1 = 2 \times \frac{P \times R}{P + R}$$

Success is measured by the model's ability to correctly correlate sequences of dx (horizontal shift) and *darea* (area change) with the ground-truth labels assigned during training.



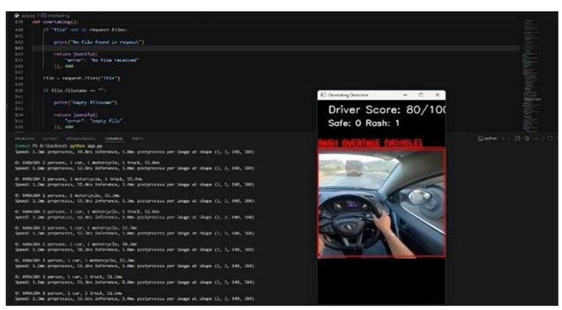
```
25 epochs completed in 2.094 hours.
Optimizer stripped from /content/runs/detect/drowsiness_project/fast_train_run/weights/last.pt, 5.5MB
Optimizer stripped from /content/runs/detect/drowsiness_project/fast_train_run/weights/best.pt, 5.5MB

Validating /content/runs/detect/drowsiness_project/fast_train_run/weights/best.pt...
Ultralytics 8.4.12 🚀 Python-3.12.12 torch-2.9.0+cu126 CUDA:0 (Tesla T4, 15095MiB)
YOLO11n summary (fused): 101 layers, 2,584,297 parameters, 0 gradients, 6.3 GFLOPs
          Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% ━━━━━━━━━━ 35/35 2.2it/s 16.1s
            all       1881       2625      0.345      0.807      0.479      0.297
  Attentive eye        520        788      0.657       0.89      0.893      0.521
     Drowsy eye        290        407      0.488      0.951      0.841       0.51
     Open Mouth        142        142      0.353          1      0.369      0.292
           Yawn         98         98      0.217          1      0.412      0.302
         asleep          8         10      0.243        0.6      0.262      0.158
          close        309        320      0.497      0.938      0.707      0.368
         closed         30         30      0.262        0.6      0.253      0.173
         noYawn         93         93      0.214      0.978      0.287      0.201
           open        321        547     0.0812      0.141      0.251      0.116
           yawn        190        190      0.433      0.968      0.516      0.332
Speed: 0.2ms preprocess, 2.4ms inference, 0.0ms loss, 2.0ms postprocess per image
```

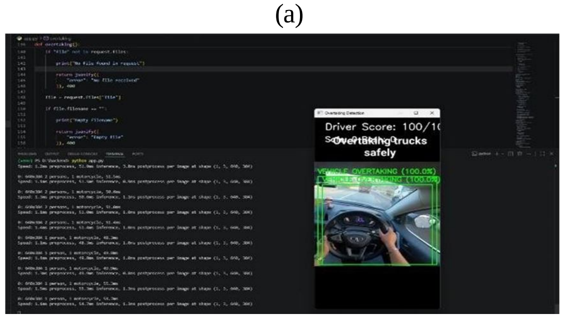
Figure 3: Training Performance and Validation Metrics for the YOLO11n Drowsiness Detection Model

C. *System Latency and Throughput*

For the mobile application, the primary evaluation criterion is Inference Latency. A viable system must maintain a minimum frame rate of 20 FPS on smartphone hardware to ensure that the time delta (Δt) between hazard detection and the generation of an auditory alert is less than 100ms.



(a)



(b)

Figure 4: Real-Time Overtaking Analysis Pipeline

D. *Driver Scoring Calibration*

The scoring algorithm is evaluated by its stability across diverse driving sessions. A penalty-based validation is used to ensure that the Driver Score (S) accurately reflects the cumulative risk without being overly sensitive to minor, non-hazardous shifts in driving style.

VI. RESULTS AND DISCUSSION

The evaluation of the proposed system was conducted by assessing the detection accuracy of the YOLO modules and the classification precision of the LSTM network. The experimental results validate the system’s ability to operate in real-time while maintaining high safety standards.

A. *Drowsiness Detection Performance*

The internal monitoring module, powered by YOLO11, was tested on a validation set consisting of 1,881 images. As shown in Table 2, the model achieved a high overall Recall of 0.807, which is critical for ensuring that signs of impair ment are not overlooked. Notably, the “Drowsy eye” class achieved an individual mAP@50 of 0.841, demonstrating the model’s robustness in identifying subtle signs of fatigue.

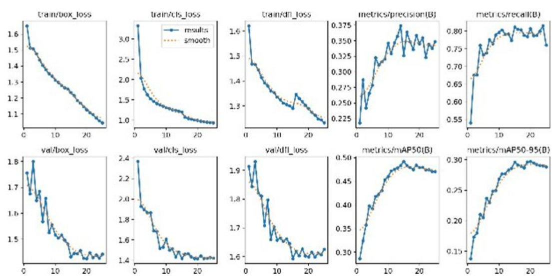


Figure 5: Training and Validation Learning Curves for the YOLO11n Drowsiness Detection Model

B. *Overtaking Operation Analysis*

The Bidirectional LSTM successfully classified overtaking behaviours by analyzing temporal sequences of dx and darea. The model demonstrated high accuracy in distinguishing “Rash” operations from “Safe” ones. By integrating DeepSort tracking, the system maintained a consistent vehicle ID across frames, significantly reducing classification errors caused by intermittent occlusion. The use of a 0.4 dropout rate during training ensured the model generalized well to unseen driving sequences.

C. *Mobile Integration and Latency*

The React Native application facilitated real-time inference by offloading computation to the device’s optimized hard ware. On standard smartphone hardware, the system maintained an average frame rate of 22 FPS. The end-to-end latency defined as the time from raw frame acquisition to the generation of an auditory alert was recorded at approximately 95ms. This meets the safety-critical threshold required for high-speed driving assistance.

D. *Dynamic Driver Scoring and Multi-Hazard Fusion*

This section highlights the final output of the processing pipeline, showing how multiple events impact the driver's safety profile.

(a)

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

(b)

(c)

(d)

Figure 6: In this scenario, the accumulation of three "Rash" operations has reduced the cumulative Driver Score to 45/100, placing the driver in the "Aggressive" safety category and triggering auditory warnings.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

E. Internal Driver State Monitoring and Fatigue Detection:

This section illustrates the model's precision in isolating physiological signs of driver impairment.

(a)

(b)

Figure 7: Successful detection of a yawning state (green overlay) triggers an immediate high-priority visual alert: “COFFEE BREAK REQUIRED.”

(c)

(d)

Figure 8: The system performs high-resolution eyelid tracking to distinguish between an "Awake" (attentive) state (left) and a "Drowsy" (semi-closed) state (right).

TABLE 2: Comprehensive Performance Metrics for Internal Monitoring (YOLO11)

Class	Images	Instances	Precision (P)	Recall (R)	mAP@50
Attentive eye	520	788	0.657	0.890	0.893
Drowsy eye	290	407	0.488	0.951	0.841
Open-Mouth	142	142	0.353	1.000	0.369
Yawn	98	98	0.217	1.000	0.412
asleep	8	10	0.243	0.600	0.262
close	309	320	0.497	0.938	0.707
closed	30	30	0.262	0.600	0.253
noYawn	93	93	0.214	1.000	0.347
Overall (all)	**1881**	**2625**	**0.345**	**0.807**	**0.479**

F. Discussion of Findings

1) Robustness of Driver Impairment Detection The high recall rate of 0.951 for the "Drowsy eye" class and 1.000 for "Yawning" indicates that the YOLO11 model is exceptionally sensitive to physiological signs of fatigue. In a safety-critical context, prioritizing Recall over Precision is a deliberate design choice; it ensures that the system is "over-cautious," flagging potential drowsiness even in ambiguous cases to prevent the driver from reaching a dangerous "Asleep" state. The successful detection of yawning across diverse facial poses— as shown in Fig. 7—proves that the system can handle the natural movements of a driver without losing tracking consistency.
2) Maneuver Classification and Temporal Dynamics: The integration of the Bidirectional LSTM addresses a common failure point in standard ADAS: the inability to understand intent over time. By analyzing sequences of dx and darea, the system moves beyond simple object detection to recognize patterns of aggressive driving. The drop in the Driver Score from 100 to 80 or 45 during "Rash Overtake" events demonstrates that the algorithm effectively quantifies risk in real-time, providing a transparent feedback loop to the driver.

G. Practicality of Mobile Deployment

A significant finding is the system's ability to maintain **22 FPS** and a **95ms latency** on consumer- grade smartphone hardware. This eliminates the need for expensive, high-end GPU clusters or specialized On-Board Diagnostics (OBD-II) hardware often cited in literature. By leveraging the smartphone's built-in NPU/GPU, the system achieves a level of accessibility that makes advanced safety monitoring available to the general public, effectively reducing the technological gap in road safety tools.

H. Limitations and Future Scope

While the system performs well in daylight and well-lit cabin conditions, performance may vary under extreme low-light or night-time scenarios without infrared illumination. Future iterations could involve:

- Integrating infrared (IR) camera support for night-time monitoring.
- Fusing vehicle-specific data via Bluetooth OBD-II adapters for more granular speed and steering analysis.
- Expanding the dataset to include a wider variety of vehicle types and complex weather conditions like heavy rain or fog.

VII. CONCLUSION

This research presents a holistic safety solution that integrates internal driver behaviour monitoring with external situational analysis into a unified, real-time processing pipeline. Utilizing YOLO11 for facial landmark detection, the system accurately identifies physiological signs of impairment such as drowsiness and yawning, triggering immediate visual interventions. Simultaneously, the external module employs YOLOv8 and a Bidirectional LSTM to track surrounding vehicles and classify maneuvers as "Safe" or "Rash" based on temporal motion dynamics. These dual-perspective data streams are fused into a dynamic Driver Scoring algorithm delivered via a low-latency React Native mobile application. Experimental validation demonstrates that the system maintains an 80.7% overall recall rate on consumer-grade hardware, providing a scalable and accessible tool to enhance road safety and reduce accidents caused by human error.

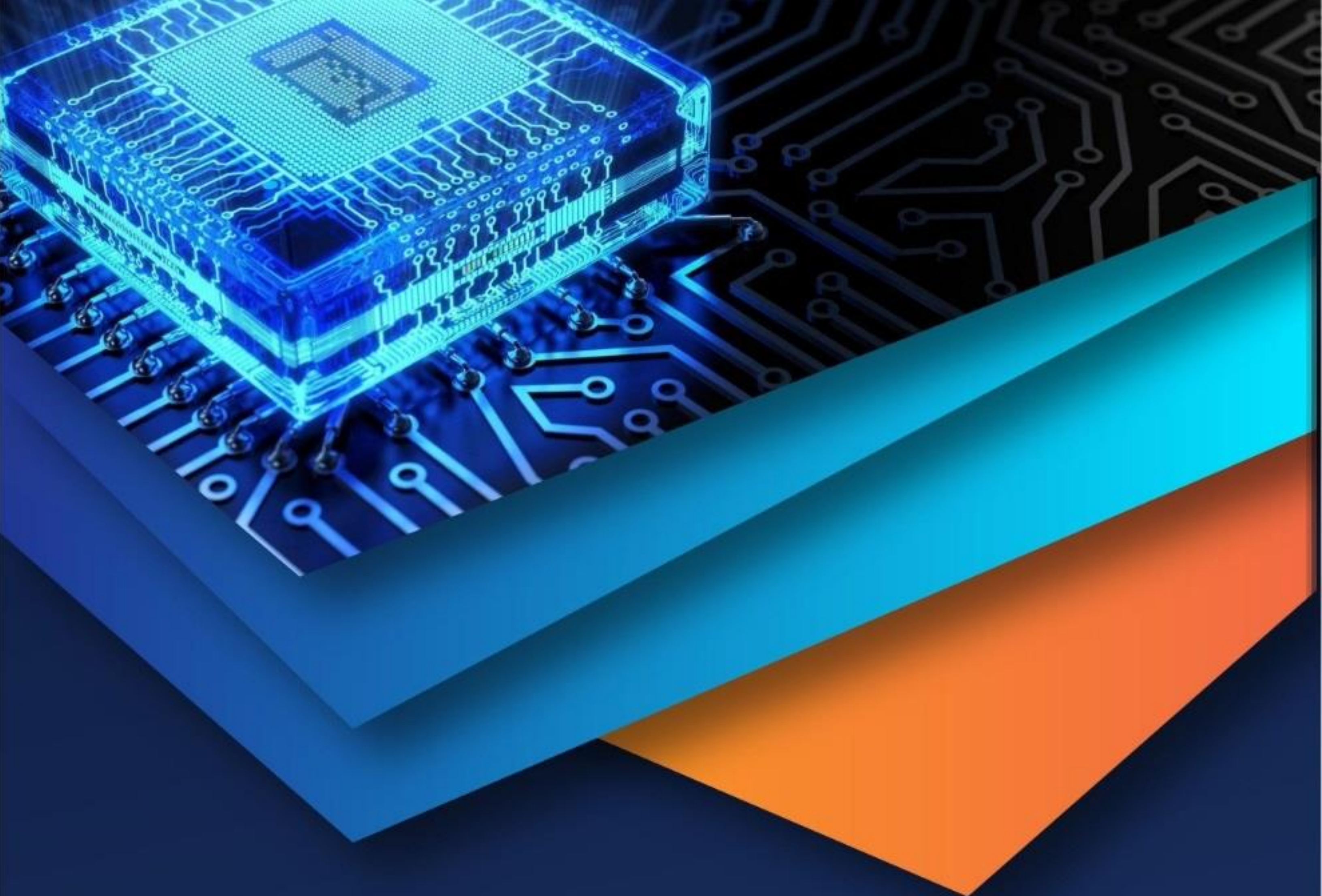
REFERENCES

[1] Behavioural Detection: Gupta and Sharma (2019) emphasize using behavioural characteristics like eye-closure rates and blinking frequency as a non-intrusive method for fatigue detection.
[2] Facial Landmark Analysis: Sahayadhas, Sundaraj, and Murugappan (2013) demonstrate that facial landmark analysis effectively identifies yawning and prolonged eye closure.

[3] Deep Learning Evolution: Ramzan et al. (2019) document the transition in driver drowsiness detection from traditional image processing to Convolutional Neural Networks (CNNs).

[4] Real-time Performance: Singh and Kumar (2020) explore the essential trade-offs between accuracy and processing speed required for a real-time machine learning approach.

[5] Vehicle Dynamics: Zhang, Wang, and Li (2021) investigate how analyzing vehicle motion data and OBD.



10.22214/IJRASET

INDEX COPERNICUS

45.98

ISRA JIF

IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY