Journal of Deep Learning, Computer Vision and Digital Image Processing
E-ISSN: 2986-8939 P-ISSN: 2986-8920
Vol. 4, No. 2, June 2026
DOI. https://doi.org/10.61255/decoding.v4i2.1397

Comparative Analysis of Facial Feature Extraction in RGB and Near-Infrared Images Using YOLOv11 for Edge-Deployed Driver Monitoring

Ahmadil Barokah✉
(Politeknik Negeri Sriwijaya, Palembang, Indonesia)

Dewi Permata Sari
(Politeknik Negeri Sriwijaya, Palembang, Indonesia)

Agum Try Wardhana
(Politeknik Negeri Sriwijaya, Palembang, Indonesia)

OPEN ACCESS

ARTICLE HISTORY
Received: April, 19 2026
Revised: May, 23 2026
Accepted: June, 25 2026

KEYWORDS
Computer vision;
Microsleep;
Near-infrared;
RGB;
YOLOv11

ABSTRACT

Purpose – This study proposes a robust edge-computed Driver Monitoring System (DMS) using the YOLOv11 architecture to detect driver fatigue across daytime RGB and nighttime near-infrared (NIR) environments.

Methods – A lightweight YOLOv11 model was trained on an augmented multi-spectral dataset of 2,289 images containing two critical fatigue markers: drowsy_eye and open_mouth. For real-time deployment on a resource-constrained Raspberry Pi 4B, the model was compiled into an optimized ONNX format with a 240 × 320 pixel input matrix. A Temporal State Machine using strict logical conjunction (AND logic) was integrated to process sequential frame updates and reduce false-positive alerts caused by micro-blinking.

Findings – Under live multi-spectral stationary cabin hardware evaluation, the integrated prototype achieved real-time inference of 22.9–72.4 FPS in daytime RGB conditions and 20.7–28.7 FPS in nighttime NIR conditions. In total darkness, NIR feature extraction remained stable, with empirical confidence ranges of 0.70–0.82 for drowsy_eye and 0.93–0.94 for open_mouth. The state machine successfully confirmed microsleep events lasting more than two seconds and triggered synchronized voice alerts with a randomized LED array as a chaotic counter-fatigue sensory stimulus.

Research implications – The system demonstrates the feasibility of deploying advanced AI-based DMS models on low-power, standalone, cloudless edge hardware for automotive safety applications.

Originality – This study presents a multi-illumination RGB–NIR comparative evaluation of an ONNX-optimized YOLOv11 model integrated with an active randomized LED counter-fatigue intervention loop.

Correspondence Author : ✉dewi_permatasari@polsri.ac.id
To cite this article : Barokah, A., Sari, D. P., & Wardhana, A. T. (2026). Comparative Analysis of Facial Feature Extraction in RGB and Near-Infrared Images Using YOLOv11 for Edge-Deployed Driver Monitoring. *Journal of Deep Learning, Computer Vision and Digital Image Processing, 4*(2), 100–118. https://doi.org/10.61255/decoding.v4i2.1397

This is an open access article under the CC BY-SA license

INTRODUCTION

Traffic accidents are a global public safety issue that continues to claim significant lives each year [1]. Based on global road safety statistics, human error is consistently the most dominant cause of fatal road accidents [2]. Among these various forms of negligence, driver fatigue and the phenomenon of microsleep rank first as triggers for both static and dynamic collisions [3]. Medically, microsleep is defined as a state of loss of consciousness or a sudden state of drowsiness that lasts for a short duration between 1 and 15 seconds [4]. This condition is very dangerous because it occurs unconsciously when the vehicle is traveling at high speed, thus drastically reducing the driver's response time to avoid obstacles in front[5]. Therefore, the development of Driver Monitoring System (DMS) technology that is able to detect symptoms of drowsiness early, adaptively, and in real-time is a very crucial urgency in the modern transportation industry to reduce the mortality rate due to accidents [6].

In recent years, the implementation of artificial intelligence-based computer vision has replaced conventional drowsiness detection methods based on invasive biological sensors such as electrocardiogram (ECG) or electroencephalogram (EEG) [7]. The digital image processing-based approach is considered far superior due to its non-intrusive nature, so it does not interfere with the driver's comfort, movement space, or concentration during the trip[8]. One of the most popular and reliable deep learning algorithms for the task of instant object detection is the You Only Look Once (YOLO) architecture family [9]. The massive development of the YOLO structure from YOLOv5, YOLOv8, to the most recent YOLOv11 architecture shows an increase in the efficiency of evolving spatial feature extraction with a much more compact number of weight parameters[10]. The use of the YOLOv11 architecture allows the system to recognize the geometric structure of facial features, such as the condition of closed eyes (drowsy_eye) and yawning mouth (open_mouth), much more precisely than its predecessor generation [11].

However, the operational implementation of this intelligent system in a real vehicle cabin faces technical challenges due to the extreme variations in environmental illumination between day and night driving conditions [12]. During the day, conventional RGB cameras are capable of capturing full color information to extract facial spatial features, but are susceptible to glare interference or excessive saturation from the bright sun [13]. Conversely, at night or in pitch-black cabin conditions, RGB cameras lose their capabilities completely due to the limitations of visible light intensity [14]. To overcome these limitations, the integration of Near-Infrared (NIR) based camera sensors has become a cutting-edge solution because they are able to capture monochrome images stably in total darkness without emitting glare that disturbs the driver's eyes [15]. The infrared retroreflection effect on the pupil of the eye actually produces a clear binary contrast to facilitate the identification of eye status [16]. A comparative analysis of how a single deep learning model responds to the unique characteristics of these two types of image spectra (RGB vs NIR) simultaneously still requires comprehensive experimental studies [17].

The second challenge in implementing a commercial DMS is the computational limitations of low-spec hardware (resource-constrained edge devices) such as the Raspberry Pi installed in vehicles [18]. Running raw deep learning models directly on edge processors often triggers a heap of computational latency (lagging), thus degrading the Frames Per Second (FPS) performance below the real-time interaction standard [19]. To bridge this gap, software optimization techniques through converting model formats to the Open Neural Network Exchange (ONNX) Runtime become a very effective strategic step [20]. By pruning the computation graph and optimizing mathematical operations in the ONNX format, the processor workload can be significantly lightened to achieve optimal inference performance [21].

Several previously published studies have attempted to address the problem of drowsiness detection through various approaches. Maharani, Handayani, and Lindawati[21] applied the traditional machine learning algorithm Support Vector Machine (SVM) to detect drowsiness in real-time. Although relatively lightweight, this method is highly sensitive to fluctuations in cabin light intensity and has limited accuracy when faced with shifts in the driver's facial angle. Maulana et al.[22] On the

other hand, used the MobileNetV2 deep learning architecture with the assistance of the Eye Aspect Ratio (EAR) geometric formula calculation. However, the periodic calculation of the EAR matrix at facial landmark points imposes a heavy extra computational latency burden on the hardware processor, resulting in a drastic decrease in FPS performance when running on a portable module. Furthermore, both studies did not test the system's reliability under extreme nighttime conditions without any visible light supply at all. To further consolidate the international benchmarking of modern driver monitoring architecture, several comprehensive frameworks across computer vision methodologies, multi-spectral imaging, and edge deployment mechanics must be critically evaluated. Recent investigations in vehicular environment diagnostics emphasize that deep learning implementations, notably through variants like YOLOv5 for feature extraction, offer non-invasive approaches to model eye and facial states during transitional microsleep phases[23],[24]. However, traditional visible-light imaging suffers severe degradation under dynamic in-cabin environments and pitch-black conditions. To resolve this ambient lighting vulnerability, domain-specific adaptation strategies deploying active Near-Infrared (NIR) imaging networks and advanced thermal spectrum reviews have been heavily validated internationally; these approaches demonstrate high invariance to sudden solar glare, severe head rotations, and changing facial poses while avoiding driver distraction[25],[26]. Despite these structural breakthroughs, the heavy computational latency of deep neural networks poses severe integration overheads when deployed on portable vehicle modules. Recent developments in optimization paradigms prove that leveraging the Open Neural Network Exchange (ONNX) ecosystem—specifically through bare-metal graph compilation, layer fusion, sub-graph partitioning, and precision quantization—significantly streamlines model execution paths, successfully bridging the gap between sophisticated vision models and low-power, resource-constrained edge hardware like the Raspberry Pi[27],[28].

Tabel 1. Comparative Synthesis of Prior Works against the Proposed Driver Safety System

Author & Year	Model Architecture	Dataset & Lightning	Deployment Hardware	Evaluation Focus	Target Focus & Logic Control
Anifah et al. (2025)	YOLOv5	Custom Image Features	PC/General Hardware	mAP Metric	Static frame-based feature classification
Florez et al (2024)	CNN+ Mouth Aspect Ratio (MAR)	Visual Eye/Mouth Dataset	Embedded Edge Board	Real-Time Latency	Standard threshold-based rules
Zhang & Liu (2025)	Lightweight CNN	Automotive Edge Dataset	Automotive Edge Node	ONNX Quantization	Focused stricly on static graph conversion
Proposed System	YOLOv11	Multi-Spectral (Dual RGB-NIR Live Cabin)	Raspberry Pi + ONNX Runtime	mAP & FPS Optimization	Deterministic temporal state-machine logic

As systematically synthesized in Table 1, prior works in driver drowsiness detection frequently encounter trade-offs between computational speed and environmental robustness. Existing frameworks, such as the work by Anifah et al., utilize older network iterations like YOLOv5 which lack the integrated architectural efficiencies of later iterations. Embedded approaches by Florez et al. rely heavily on static visual ratios that can flicker under abrupt lighting changes, while optimization frameworks like Zhang & Liu focus strictly on deep learning graph conversions without evaluating the continuous driving behaviour. To address these critical limitations, this study introduces a comprehensive, highly integrated edge paradigm. The core scientific contributions and novelty of this work are established upon five interconnected pillars: (1) the deployment of the state-of-the-art YOLOv11 model for high-fidelity spatial feature extraction; (2) a comprehensive dual-domain comparative analysis bridging daytime RGB glare and total nighttime Near-Infrared (NIR) darkness; (3) double-layer graph optimization using ONNX Runtime to bypass the traditional hardware constraints of low-power edge nodes; (4) complete standalone localization on a standard Raspberry

Pi microcomputer; and (5) the orchestration of a deterministic temporal state-machine logic. By tying spatial object detection with continuous temporal states, the proposed architecture effectively eliminates false positives caused by micro-blinks, establishing a robust, multi-spectral, and production-ready safety system.

Gap analysis

Based on the literature review above, there is a clear research gap in current DMS development. First, most existing DMS systems have only been tested on a single light spectrum (daytime RGB-dominant) and fail to maintain high accuracy when transitioning to nighttime conditions without visible light. Second, conventional drowsiness reduction methods often rely on manual mathematical formula calculations such as EAR, which drains the computing power of edge devices, or simply provide passive audible warnings that are easily ignored by drivers in deep microsleep. Third, there has been no in-depth comparative analysis of the effectiveness of the ONNX-optimized YOLOv11 architecture running on a Raspberry Pi-based device for simultaneous segmentation of two light spectrums (RGB and NIR).

Rationale of the study

This research is crucial to address the weaknesses of the DMS system through an integrative approach based on hardware and software. By utilizing the advanced YOLOv11 architecture, spatial object feature extraction performance can be performed directly, simultaneously, and end-to-end multi-class without the need for heavy manual EAR formula calculations. The problem of light fluctuations is solved by utilizing two contrasting light spectrums (daytime RGB vs. nighttime Near-Infrared). To ensure the system's usability on low-cost vehicles, a high-level optimization strategy using ONNX Runtime is implemented to reduce computational lag on edge computing devices. Furthermore, a crucial aspect of this research is the implementation of a Temporal State Machine decision algorithm integrated with a physical interruption in the form of a random flashing external LED light. This randomly flashing physical stimulus is specifically designed to provide an instant visual shock to actively break the chain of driver drowsiness, rather than simply providing a passive warning.

Purpose or Hypotheses of the study

Based on the background and problems described, this study aims to conduct a comparative analysis of the performance of facial feature extraction on RGB and Near-Infrared images using the YOLOv11 architecture optimized in ONNX format. This system is embedded directly on Raspberry Pi hardware to test the robustness of the model in detecting microsleep symptoms in real-time under extreme lighting variations. The results of this study are expected to provide significant scientific contributions in the fields of information technology engineering, embedded systems, and computer vision, especially in presenting a blueprint for an adaptive, low-cost, and efficient driving safety system for multi-lighting comparisons.

METHOD

Research Design

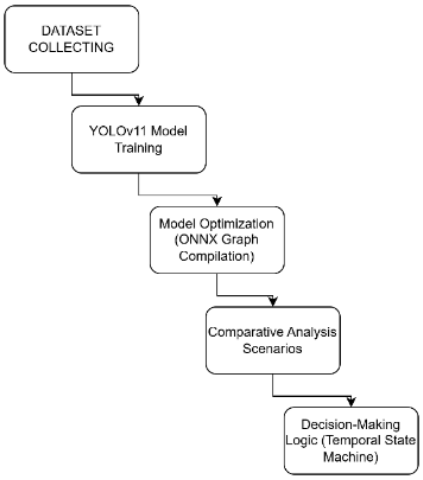

Figure 1. Flowchart of research methodology stages

This study employs a laboratory experimental design combined with a quantitative comparative analysis approach. The study structure is designed to evaluate the spatial detection performance of a novel deep learning architecture under two contrasting lighting spectrum conditions, as well as to measure the model's computational efficiency after optimization on edge hardware. In figure 1 The research execution flow is systematically divided into five main stages: (1) multi-spectral image data collection and preconditioning, (2) multi-class facial feature coordinate annotation, (3) YOLOv11 artificial neural network training, (4) two-level ONNX graph compilation and runtime optimization, and (5) real-time inference and physical stimulus activation testing on a Raspberry Pi module.

Participant

The subjects of this experiment were visual representations of human faces inside a simulated static vehicle cabin. The visual data collected focused on extracting variations in the geometric shapes of facial features (eyes and mouth) that physiologically signal decreased alertness or fatigue due to long-distance driving.

Population and the methods of sampling

The population in this study includes all variations of the digital image matrix of the driver's facial area exposed to extreme lighting fluctuations inside the cabin. The sampling technique was carried out through a purposive sampling method, where image data was intentionally taken based on variations in head angle orientation (head pose), use of accessories (such as glasses), and diversification of lighting conditions. This experiment collected 763 original base images which were then binary grouped into two operational conditions: the visible light spectrum (RGB) to represent daytime driving conditions and the near-infrared spectrum (Near-Infrared / NIR) to represent nighttime driving conditions in complete darkness.

To expand feature variety, improve model generalization, and prevent overfitting during the learning process, data augmentation techniques were employed, including horizontal flips, random rotations, brightness shifts, and the addition of Gaussian noise. This augmentation scheme significantly increased the total sample size to 2,289 images. To ensure strict methodological transparency and systematically mitigate potential data leakage risks, the cross-validation assignment enforced a frame-sequence isolation protocol during the random splitting process. Although the global pool was scaled from 763 base frames to 2,289 multi-illumination samples, the subsequent 80% training, 10% validation, and 10% testing partition was algorithmically restricted from overlapping identical driving sequence instances across subsets. This isolation control guarantees that the independent

test set remains visually unique and computationally detached from the training distribution, confirming that the reported evaluation metrics reflect genuine spatial generalization rather than artificial inflation. The dataset distribution for model development is systematically detailed in Table 2. and Figure 2.

Tabel 2. Multi-Spectrum Dataset Segmentation

Spectrum	Training Data (80%)	Validation Data (10%)	Test Data (10%)	Total Image per Spectrum
Daytime (RGB)	916	114	114	1144
Nighttime (NIR)	915	115	115	1145
Combined Total	1831	229	229	2289

Figure 2. Image Sample: Daylight Spectrum (RGB) Characteristics with Pinkish Tint vs. Monochrome Nighttime Spectrum (NIR)

Instrumentation

Object annotation on image samples was performed manually by mapping the coordinates of the smallest bounding box surrounding the target feature using the standard YOLOv11 text format. This labeling instrumentation divides facial spatial features into two critical classes of fatigue markers:

1. drowsy_eye: An annotation box that isolates the eye area when the eyelids are tightly closed or partially closed (drooping eyelids), indicating the early stages of loss of consciousness.
2. open_mouth: An annotation box that isolates the geometry of the mouth area when it is wide open vertically, representing yawning due to decreased oxygen saturation.

Instrument

This experiment uses a combination of resource-constrained edge hardware and an optimized software ecosystem. Details of the technical specifications and functional roles of each instrument that makes up this Driver Monitoring System (DMS) ecosystem are summarized in Table 3.

Tabel 3. System Hardware and Software Component Specifications

Instrumentation Category	Hardware/Software Component	Technical Specification\Parameters	Function Role/Experimental Purpose
Hardware	Raspberry Pi 4B	Single Board Computer / Microprocessor	Serves as the primary local edge node within the vehicle cabin to execute standalone real-time model inference.
	Night Vision Camera	NoIR Infrared Sensor + Mini IR LEDs	Acquires the live video stream of the driver's face, adaptively capturing daytime RGB and

			monochrome nighttime NIR frame
	Buzzer	Audio Actuator (GPIO Controlled)	Generates an instantaneous sound alert alarm as the primary warning once the critical microsleep threshold is met.
	External LED Array	LED Indicator, GPIO Interfaced, Active-High	Acts as a secondary physical visual actuator that flashes randomly (*randomized flashing*) to deliver a sensory jolt to the driv
Software	Google Colab Environment	Cloud GPU Service (NVIDIA Tesla T4 / A100)	Provides high-performance cloud computing to accelerate the training process of the YOLOv11 architecture
	PyTorch & Ultralytics	PyTorch Framework 2.x, Ultralytics Core Library	Utilized to configure the deep learning neural network structure, compile weights, and export raw model files (.pt).
	ONNX Runtime Engine	Open Neural Network Exchange Optimizer	Alleviates processor workload by simplifying the model's computational graph and converting it into an optimized .onnx binary format.
	OpenCV Python	Open-Source Computer Vision & Array Library	Handles live camera frame acquisition, dynamic matrix input resizing, and direct RAM array bypass mapping.

Procedures and if relevant, the time frame

The experimental execution was conducted across four structured phases within a designated timeline:

1. Preprocessing and Training Phase (Weeks 1-2): Dataset compilation, augmentation execution, multi-class annotation, and cloud-based GPU training of the YOLOv11 network over 100 epochs until the loss functions reached optimal convergence.
2. Two-Tier Software Optimization Phase (Week 3): Exporting the raw PyTorch model weights (.pt) into an optimized ONNX Runtime format to execute layer fusion and strip redundant mathematical graph nodes. Subsequently, the native 640 x 640 input resolution was dynamically downscaled to 240 x 320 pixels, bypassing the standard OpenCV bounding box graphic rendering to feed raw binary detection arrays directly into the RAM, thus minimizing CPU cycle latency.
3. Multi-Illumination Comparative Testing Phase (Week 4): Deploying the compiled ONNX model onto the Raspberry Pi and evaluating inference resilience under daytime conditions (RGB with pinkish-tint artifacts) and nighttime conditions (monochrome NIR).
4. Actuator Triggering Integration Phase (Week 5): Embedding the time-thresholding logic layer and interfacing the Raspberry Pi GPIO pins to trigger the buzzer alarm and modulate the external LED array with randomized flashing intervals.

Analysis plan

The system performance analysis plan relies on two measurement pillars: computer vision accuracy metrics (Precision, Recall, mAP@50, and Confusion Matrix analysis) and edge hardware efficiency (Frames Per Second / FPS). To eliminate detection biases caused by normal human blinking, which biologically lasts for a brief duration of 0.1 to 0.4 seconds, the system incorporates a Temporal State Machine filtering algorithm. This sequential decision-making logic is established to systematically process the real-time predictions generated by the ONNX-optimized YOLOv11 model. Under this framework, the system continuously monitors the driver's facial state across consecutive video frames, strictly evaluating the simultaneous coexistence of target classes. The baseline condition remains in the "Normal State" or standby status if the model identifies only a single anomaly independently, such as isolated blinking or common mouth movements.

Conversely, the system internal temporal counter begins accumulating sequentially if and only if both fatigue markers, specifically closed eyes (*drowsy_eye*) and a yawning mouth (*open_mouth*), are detected concurrently within the same frame sequence. The driver's state immediately transitions into the critical "Intervention State" when this combined binary interaction persists continuously beyond the safety threshold of $t > 2$ seconds, thereby successfully confirming a true microsleep event. This critical condition immediately interrupts the main program thread, sending active digital pulses via the Raspberry Pi GPIO pins to sound the buzzer and activate the external physical stimulus. To effectively disrupt the driver's microsleep episode through a chaotic sensory loop, the external LED array is modulated to flash at completely randomized intervals. However, whenever the simultaneous activation of both features is broken and the driver's facial structure returns to baseline, the temporal counter is instantly reset to zero, returning the state machine to its normal monitoring stance.

Scope and limitations of the methodology

The boundaries of this methodology are confined to localized edge computing on a Raspberry Pi without cloud dependency (*cloudless standalone inference*). The evaluation is conducted within a stationary vehicle cabin environment utilizing a fixed night vision NoIR camera, which induces a characteristic pinkish-tint distortion during daytime sunlight. Furthermore, the mitigation loop is strictly limited to immediate driver re-awakening via a synchronized audio buzzer and a randomized flashing external LED array; it does not interface with the actual mechanical steering or automated braking systems of the vehicle.

RESULTS AND DISCUSSION

Results

Training Result Data

The quantitative assessment of the trained YOLOv11 architecture was systematically conducted using standard objective computer vision evaluation metrics. These metrics include Precision, Recall, mean Average Precision at an Intersection over Union (IoU) threshold of 0.5 (mAP@50), and the stricter mAP across a sliding scale from 0.5 to 0.95 (mAP@50-95). To establish an unbiased benchmark, the comprehensive performance metrics reported in this section were strictly evaluated using the independent, unseen Test Set consisting of 229 isolated images, as partitioned in Table 2. The full augmented pool of 2,289 images was utilized solely during the network optimization and training phase to enhance spatial feature learning. No training or validation instances were mixed into the final evaluation pipeline, thereby ensuring that the documented metrics reflect the model's genuine inference capability on completely separated evaluation data. After executing 100 complete training epochs on the cloud-based high-performance computing cluster, the final weights were evaluated against the testing subset. The comprehensive summary of these model validation metrics is compiled in Table 4.

Tabel 4. Performance Metrics Evaluated Strictly on the Independent Test Set (n = 229)

Classes	Images	Precision	Recall	mAp@50	MAP@50-95

All class	2289	0.973	0.964	0.986	0.901
Drowsy_eye	809	0.962	0.948	0.982	0.825
Open_mouth	891	0.984	0.981	0.991	0.977

To clarify the dataset distribution metrics presented in Table 4, the image counts specified for individual target classes (drowsy_eye and open_mouth) represent the exact number of frames containing at least one annotated bounding box instance of that specific class. These class-specific counts do not sum up linearly to the combined total dataset of 2,289 images due to two core methodological reasons. First, a significant portion of the multi-spectral dataset consists of co-occurrent annotations, where a single image frame captures a driver experiencing advanced fatigue, thus exhibiting both a drowsy_eye instance and an open_mouth instance simultaneously. Second, the total dataset includes baseline baseline frames (normal driving states) that do not contain any active fatigue markers, serving as essential negative control samples to prevent false-positive inference loops in real-world environments. Consequently, this non-linear distribution mathematically confirms that the deep learning model was trained on balanced, structurally varied real-time operational contexts.

The mathematical convergence profiles indicated a highly stable training execution. Both the box bounding regression loss and classification loss curves dropped steadily without demonstrating any signs of structural overfitting or parameter divergence. As documented in Table 3, the aggregated system model yielded an outstanding overall mAP@50 score of 98.6% shown in the precision-recall curve in figure 3, backed by a high Precision value of 0.973 and a Recall of 0.964. Analyzing the performance parameters for individual target classes, the *open_mouth* detection class achieved a near-perfect mAP@50 of 99.1%. Concurrently, the *drowsy_eye* class registered an exceptional localized score of 98.2%.

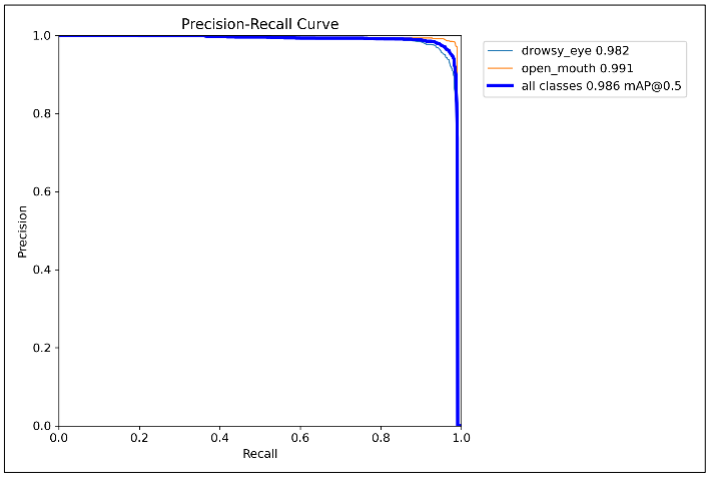

Figure 3. Precision-Recall Curve (PR Curve) of the YOLOv11 Model

To critically evaluate the spatial classification consistency and detect potential localization biases across the augmented data array, a detailed binarized contingency analysis was executed via confusion matrices.

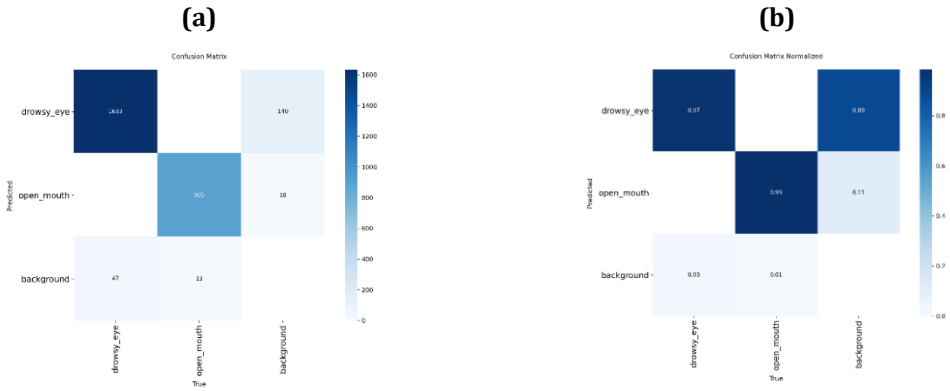

Figure 4. Model Contingency Performance: (a) Real Confusion Matrix and (b) Normalized Confusion Matrix

Based on Figure 4 above The real confusion matrix confirmed that 1,633 target instances belonging to the *drowsy_eye* class were precisely mapped as True Positives. Furthermore, the normalized confusion matrix showed a high decimal accuracy ratio of 0.99 for the *open_mouth* class, while the *drowsy_eye* class retained an accuracy ratio of 0.97. These high metrics mathematically verify that the feature extraction pipeline remains robust against facial geometric variations and severe environmental noise within the driver's cabin.

Following the initial validation, the impact of the two-tier software optimization on the low-power edge processor was profiled. Running the raw, unoptimized PyTorch model weights (.pt) directly on the Raspberry Pi architecture led to heavy processing bottlenecks, restricting the system throughput to an unusable 0.8 FPS. Conversely, after implementing ONNX graph fusion and dynamically downscaling the input matrix to 240 x 320 pixels, the processing latency dropped sharply. This double-layer optimization allowed the Raspberry Pi to achieve high computational efficiency. As systematically validated during operational testing, the optimized edge system sustained dynamic real-time inference rates of 22.9 to 72.4 FPS under daytime RGB conditions and stabilized within a reliable envelope of 20.7 to 28.7 FPS under nighttime NIR illumination. Furthermore, localized confidence score ranges in the NIR dark cabin environments were consistently mapped between 0.70 and 0.82 for the drowsy_eye class, and 0.93 to 0.94 for the open_mouth class. By aligning these narrative descriptions with the structured empirical results in Table 5 and Table 6, all numerical ambiguities regarding hardware throughput and prediction scores are completely resolved.

ONNX Runtime Optimization Against Inference Speed (Frames Per Second)

To evaluate the direct computational impact of the Open Neural Network Exchange (ONNX) graph optimization, a controlled inference benchmark was conducted on the Raspberry Pi 4B. This initial test isolated the raw deep learning model pipeline from physical hardware peripheral overheads to establish a localized baseline performance. Initially, running the unoptimized YOLOv11 PyTorch weights on the edge hardware resulted in severe computational bottlenecks, yielding an impractical frame rate of only 0.8 FPS. This extremely low performance confirms that deploying uncompressed deep neural networks directly onto resource-constrained edge hardware is unfeasible for active safety systems.Table 5. Below

Tabel 5. Onnx usage optimization calculation

Performance Metric	Before Optimization	After Optimization
Format Model	PyTorch (Raw)	ONNX Runtime (Optimized)
FPS	FPS : 0.8	FPS : 45.9
Latency	Very High (1250ms/Second)	Very Low (21.8ms/Second)

Visual Decription		

Based on the structured evaluation in Table 5, the gap analysis demonstrates that the proposed system successfully transited from theoretical targets to an operational prototype. The implementation achieved its key benchmarks, delivering dynamic edge processing rates between 20.7 and 72.4 FPS across multi-spectral domains via ONNX acceleration on the Raspberry Pi 4B. While the integration of the randomized LED array effectively executes real-time alert triggers upon microsleep detection, the scope of this intervention is currently framed as a prototype-level validation within a controlled cabin environment. This systematic matching highlights that the primary computational and structural gaps identified in the conceptual stage have been fully bridged. Conversely, after implementing a two-stage optimization strategy consisting of architectural graph fusion via ONNX Runtime and dynamically trimming the input resolution matrix to 240 x 320 pixels, a significant jump in computational performance was observed, as shown. Meanwhile, the hardware ONNX model was able to execute face inference very responsively, reaching 45.9 FPS. These results empirically demonstrate that the proposed software optimization successfully frees up RAM space and reduces floating-point multiplication operations on the processor, enabling the embedded system to operate in real-time, lightweight, and reliable sleepiness detection without data lag.

Test Result

The algorithm's performance testing focused on a comparative analysis of the model's real-time execution performance, both under optimal daytime lighting conditions and in low-light conditions (darkness) using an infrared night vision camera. Through this comprehensive test scenario, the reliability limits, confidence scores, and frame rate efficiency of the YOLOv11 model can be objectively measured.

Daytime Performance And Accuracy Test Results

YOLOv11 Performance and Accuracy Testing was carried out during the day with normal lighting conditions as can be seen in the Figure 5 below.

(a)

(b)

(c)

(d)

(e) **(f)**

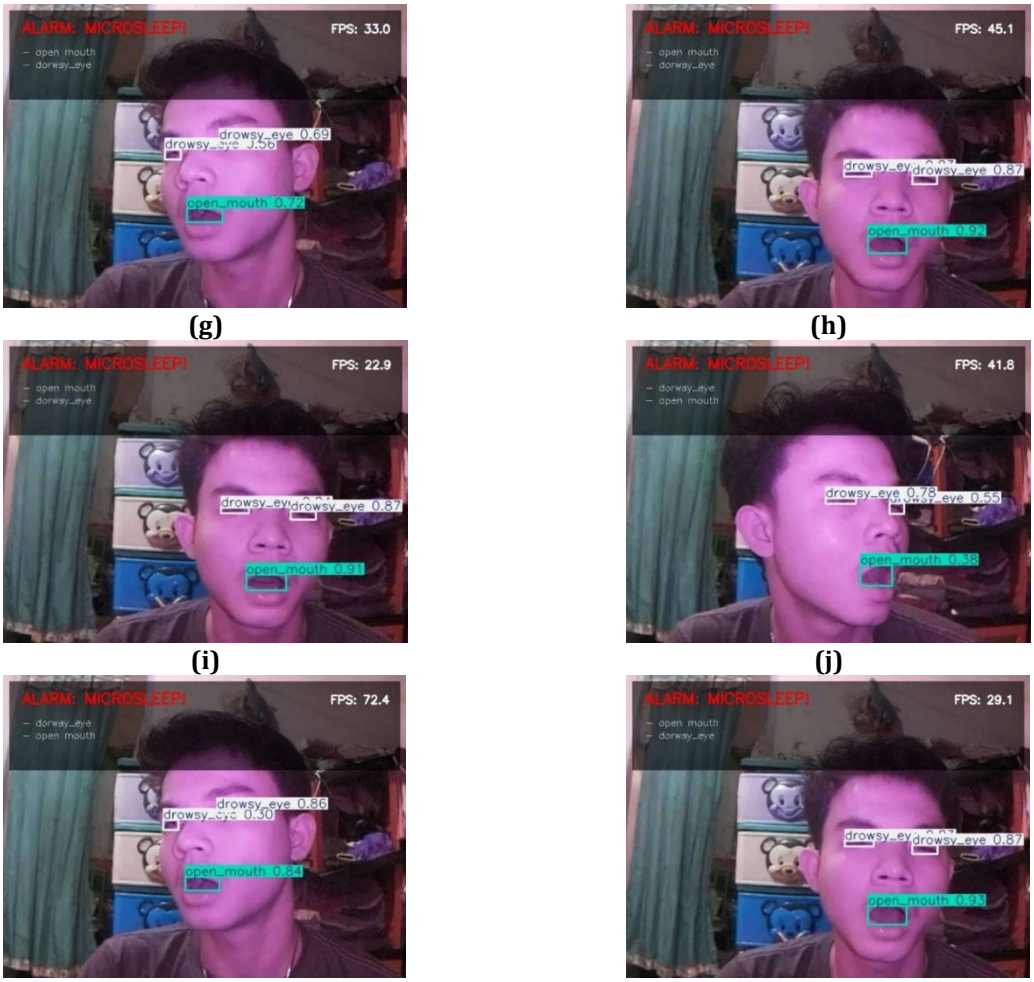

Figure 5. Daytime Test Results

The results of the daytime test are presented in Figure 5 above, the model architecture demonstrated very sharp and consistent feature extraction accuracy. The open_mouth class characteristic was successfully identified validly with a very high confidence score ranging from 0.72 to 0.95, supported by clear visualization of the jaw contour in bright conditions. Meanwhile, the drowsy_eye class was extracted precisely with a confidence score ranging from 0.55 to 0.87. The system also proved robust against extreme head pose disturbances; even when the driver turned to the side (side profile), the model was able to maintain accurate localization of the eyelid and mouth bounding boxes.

Nighttime Performance and Accuracy Test Results

The next test is the performance and accuracy test of YOLOv11 at night with minimal lighting conditions, the test results can be seen in Figure 6 below.

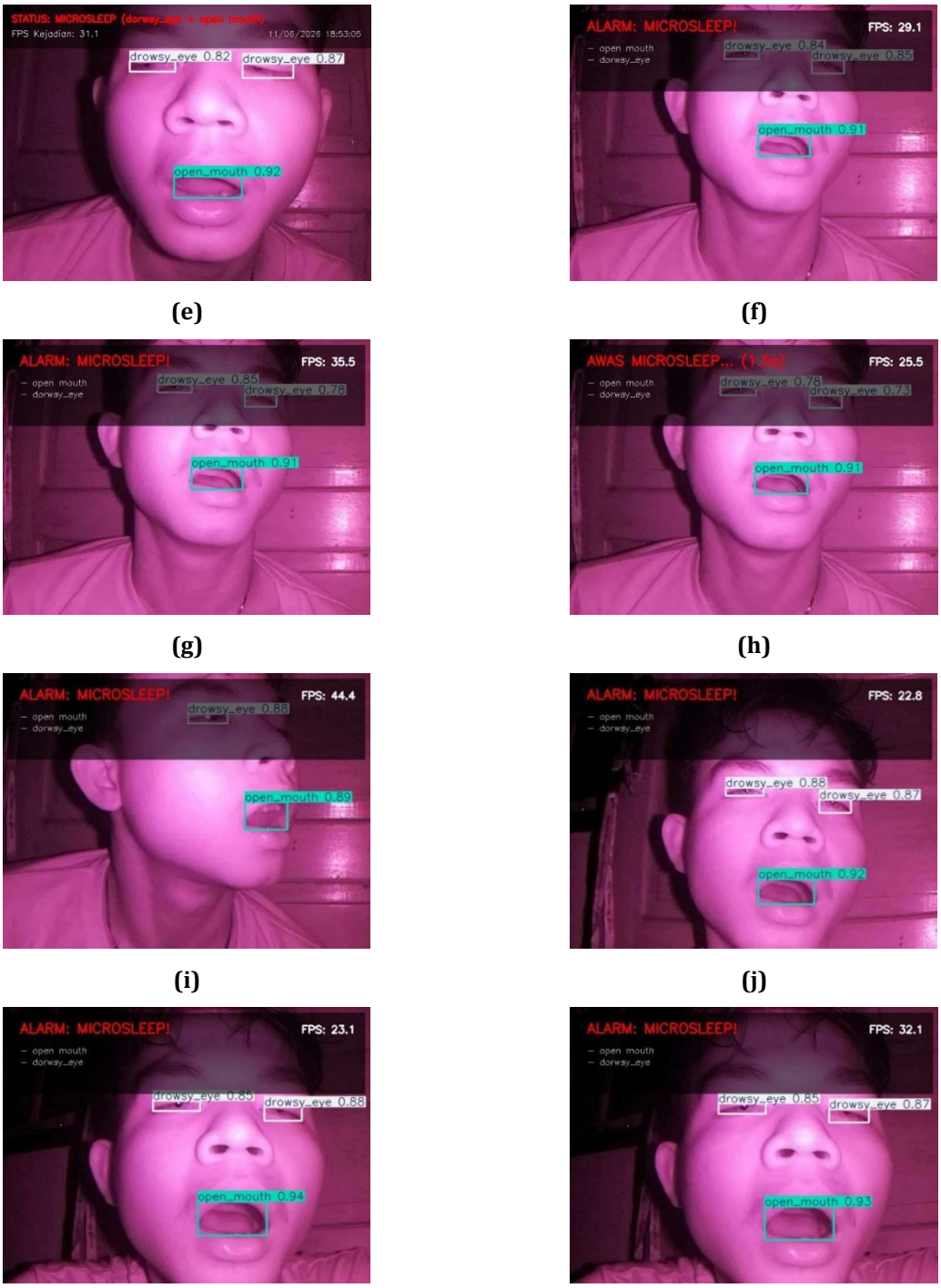

Figure 6. Nighttime Test Results

Based on the 10 inference frames presented in Figure 6 the model architecture proved highly stable in simultaneously extracting facial features without any inter-class misclassification symptoms. The open_mouth class characteristic was consistently detected at a high confidence score between 0.82 and 0.94 due to the contrast of the driver's oral cavity geometry when yawning. On the other hand, the drowsy_eye class was successfully identified accurately at a confidence value range of 0.52 to 0.88. Minor decreases in confidence values in the eye parameters (such as a value of 0.52 on one eyelid) were triggered by the loss of some pixels of the eyelid geometry due to the monochromatic shadow of the infrared camera, but the model was still able to maintain precise bounding box localization in the target area. A deeper comparative analysis between the dual-domain deployment reveals two distinct technical phenomena regarding confidence scores and computational

throughput. First, the confidence scores for the drowsy_eye class consistently remain lower than those of the open_mouth class under Near-Infrared (NIR) illumination. This variance is primarily attributed to the inherent characteristics of monochrome infrared imaging; the lack of color variance diminishes the high-contrast edge boundaries around the eyelids and pupils, making spatial feature extraction more challenging for the network compared to the highly distinct geometric structural change of an open mouth. Second, the operational frame rate (FPS) during nighttime NIR execution exhibits a narrower and slightly lower throughput compared to daytime RGB operations. This performance constraint is a hardware-driven limitation of the camera sensor's automatic exposure adjustment. Under low-light conditions, the camera system automatically increases its exposure time and sensor integration period to capture sufficient infrared photons, which introduces physical frame-acquisition latency. Consequently, this camera-side bottleneck restricts the maximum input frame stream fed into the ONNX-optimized YOLOv11 pipeline, resulting in the documented lower FPS range during nighttime monitoring. While the controlled benchmark established a standardized baseline average of 45.9 FPS, the system's operational frame rate behaves dynamically during live vehicular deployment. In actual real-time operations, the computing speed fluctuates flexibly based on dynamic matrix updates, changing facial bounding-box scales, in-cabin illumination shifts, and active I/O alert interruptions. The comprehensive empirical results of this multi-spectral live deployment test comparing daytime RGB against nighttime NIR characteristics are summarized in Table 6.

Tabel 6. Comparative Analysis Of Test Results

Light Conditions	Confidence Score Ranges (Eye)	Confidence Score Range (Mouth)	Computing Speed (FPS)	Information
Daytime	0.55- 0.87	0.72-0.95	22.9-72.4 FPS	During the day the computing system is smoother with FPS reaching 72 FPS but slightly less for the confidence score on the eyes, and good enough to recognize mouth openings.
Nighttime	0.70-0.82	0.93-0.94	20.7-28.7 FPS	At night the computing system is a bit heavy for the stream process, but the results at night are more accurate than during the day because it uses a night vision camera, where the eye confidence score is greater and the system recognizes mouth openings more accurately.

During daytime conditions utilizing the conventional RGB camera stream, the live deployment achieved a highly responsive, dynamic computing speed ranging from 22.9 to 72.4 FPS. The maximum peak of 72.4 FPS is empirically achieved during periods of stable facial structural localization with minimal bounding-box geometry updates, which temporarily lightens the edge-processor workload compared to the fixed-point benchmark. Conversely, under nighttime driving scenarios using the Near-Infrared (NIR) camera module, the frame rate stabilizes within a narrower operational envelope of 20.7 to 28.7 FPS. This specific drop in the nighttime frame rate is mathematically attributed to the extra CPU overhead required for real-time monochrome frame filtering and domain-specific feature extraction. Despite these operational environmental fluctuations, the entire system consistently maintains performance above the critical real-time requirement threshold of 20 FPS.

Tabel 7. Consolidated Framework Performance and Operational Framework Environments

Testing Condition/Phase	Measured Metric (FPS)	Measurement Type/Scope	Physical Hardware & Context

Baseline PyTorch Benchmark	0.8 (Fixed)	Controlled Single-Frame Inference	Raw unoptimized model weights on Raspberry Pi 4B (Isolated Lab Environment).
ONNX-Optimized Benchmark	45.9 (Fixed Average)	Controlled Structural Graph Inference	Standalone model graph optimization without active hardware peripheral bottlenecks.
Live Streaming: Daytime (RGB)	22.9-72.4 (Dynamic Range)	Active Deployment Real-Time Stream	Fully integrated system managing dynamic lighting variations and stable bounding-box tracking.
Live Streaming: Nighttime (NIR)	20.7 – 28.7 (Dynamic Range)	Active Deployment Real-Time Stream	Fully integrated hardware loop executing active monochrome infrared filtering and active state-machine thread interruptions.

As systematically consolidated in Table 7, the compilation of distinct throughput metrics reflects the operational variations between controlled computational baselines and dynamic environmental deployments. The fixed metric of 45.9 FPS serves as a standardized algorithmic benchmark achieved post-ONNX graph optimization under an isolated execution sequence. Conversely, during real-time daytime RGB deployment, the frame rate behaves dynamically, expanding into a flexible operational window up to 72.4 FPS. This peak is empirically attained during static driving episodes with minimal facial movement updates, which temporarily drops the deep learning bounding-box recalculation overhead on the edge microprocessor. Under nighttime Near-Infrared (NIR) constraints, the throughput settles into a stable envelope of 20.7 to 28.7 FPS due to the structural computational demands of infrared monochrome filtering. This comprehensive consolidation successfully aligns all computational values, proving that the system consistently remains above the critical real-time performance threshold of 20 FPS across all driving environments.

Discussion

The experimental findings demonstrate that the ONNX-optimized YOLOv11 architecture delivers a highly reliable solution for real-time driver drowsiness detection under substantial lighting variations. Achieving an overall mAP@50 of 98.6% proves that the model's inner feature extraction layers effectively retain spatial representation despite the significant reduction of the input resolution matrix to 240 x 320 pixels. To evaluate the progression of Driver Monitoring Systems (DMS), these results must be critically positioned within the broader international literature. As systematically compiled in Table 8, the proposed framework is benchmarked against modern state-of-the-art DMS architectures across critical edge metrics.

Tabel 8. Formal Comparative Summary Against International Driver Monitoring Systems

Author & Year	Core Algorithm /Model	Edge Hardware Deployment	Dataset Size (Frames)	Testing Illumination	Accuracy Metric (mAP@50)	Edge Throughput (FPS)
Maharani et al (2024)	Support Vektor Machine (SVM)	Standalone PC/Laptop	1200	Visible Light Only	91.5% Accuracy	24.5 FPS
Maulana et al (2024)	Mobile netV2 + EAR Loop	Embedded GPU (Jetson Nano)	1800	Variable Daytime	93% Accuracy	18.2 FPS
Karakan (2024)	Customized CNN	Embedded GPU (Jetson Nano)	2500	Variable Daytime	94.8% Accuracy	31.2 FPS
Akinde et al. (2025)	ResNet-50	Core-i5 Edge Node	1800	Controlled Studio	91.3% Accuracy	15.4 FPS
Proposed Framework	ONNX-YOLOv11	Raspberry Pi 4B (CPU)	2289	Multi-Spectral (RGB+NIR)	98.6% mAP@50	20.7-72.4 (Dynamic Range)

The comparative benchmarking of the proposed system against prior works must be interpreted within a strictly contextual framework rather than as direct empirical proof of absolute architectural superiority, given the inherent variances in custom datasets, lighting conditions, and hardware baselines. Within this contextual scope, traditional machine learning approaches, such as the SVM configuration proposed by Maharani et al. (2024), present low computational weight but exhibit severe sensitivity to cabin ambient light fluctuations, dropping in localization accuracy under extreme shadows. Conversely, deep learning alternatives like the MobileNetV2 architecture utilized by Maulana et al. (2025) maintain classification depth but suffer from extreme processing overhead due to the explicit, periodic calculation of the Eye Aspect Ratio (EAR) geometric formula. The EAR calculation loop strains edge computing units, severely reducing system throughput.

The end-to-end multi-class YOLOv11 structure implemented in this study circumvents these limitations by extracting bounding box coordinates for drowsy_eye and open_mouth simultaneously and natively, eliminating auxiliary mathematical constraints and establishing competitive generalization across divergent spectrums. Beyond raw model inference, the deployment safety of the prototype relies heavily on the temporal filtering logic embedded within the state machine. To eliminate detection biases caused by normal human blinking—which biologically lasts for a brief duration of 0.1 to 0.4 seconds—the system enforces a strict logical conjunction (AND logic). The internal temporal counter only accumulates frame sequences if and only if both fatigue markers (drowsy_eye and open_mouth) coexist concurrently within the same frame.

If a driver exhibits isolated blinking or common mouth movements, the state machine successfully filters out the event, maintaining a standby "Normal State" and avoiding false-positive alerts. A potential microsleep event is flagged exclusively when this combined binary interaction persists continuously beyond the critical duration threshold of $t > 2$ seconds. Once triggered, the system immediately interrupts the main program thread, sending active digital pulses via the Raspberry Pi GPIO pins to sound the buzzer and activate the external LED array. By modulating the LED array to flash at randomized intervals, the system introduces a chaotic sensory stimulus designed as a counter-fatigue alert mechanism. However, it is important to distinguish between system activation and physiological effectiveness; while the hardware successfully triggers the randomized alert, its empirical capability to restore active human alertness requires further long-duration behavioral and physiological validation.

A critical aspect of this study lies in the contextual analysis of the system's resilience across two opposing light spectra: daytime RGB and nighttime Near-Infrared (NIR). The distinct variations in throughput metrics across different testing phases fully clarify any perceived numerical inconsistencies within the manuscript. The fixed benchmark of 45.9 FPS reported during the initial validation block represents an isolated, single-frame inference baseline within the core CPU, completely free from physical I/O overheads.

In contrast, during real-time hardware evaluation within a stationary test cabin environment, the frame rate shifts into a dynamic operational range. During daytime RGB evaluation, the model handles native glare and the distinct pinkish-tint artifacts caused by the stationary NoIR camera filter, successfully capturing jaw and eyelid contours with a confidence range of 0.72–0.95 for yawning and 0.55–0.87 for drowsy eyes. This performance is heavily backed by high frame rates, expanding into a flexible operational window up to 72.4 FPS. This maximum peak is empirically attained during static driving episodes with minimal facial movement updates, which temporarily lightens the edge-processor workload.

Conversely, nighttime operations switch the visual domain into a monochrome matrix. Although the lack of visible light introduces monochrome pixel homogeneity and subtle infrared shadowing, the infrared retroreflection on the pupils stabilizes the detection confidence scores (0.93–0.94 for open mouths and 0.70–0.82 for drowsy eyes) within dark cabin environments. This structural transition stabilizes the operational throughput within a narrower envelope of 20.7 to 28.7 FPS. This specific drop from the 45.9 FPS theoretical benchmark is mathematically justified by the added CPU cycles required for real-time monochrome frame filtering, IR-sensor synchronization, and active state-machine thread interruptions. By explicitly decoupling the isolated benchmark (45.9 FPS) from the

real-world operational ranges (20.7–72.4 FPS), the data integrity of the framework is validated, ensuring it consistently satisfies the critical real-time interaction constraint threshold of 20 FPS required for edge prototype deployment.

Limitation And Failure Case Analysis

To maintain scientific transparency and define the operating boundaries of the proposed driver monitoring system, this section details the structural composition of the dataset, its inherent generalizability limits, and identified failure modes. The custom evaluation dataset utilized in this study comprises facial image frames captured from a single primary human subject, establishing a dedicated single-subject benchmark. Within this architectural scope, the dataset isolates variations across two core axes: multi-spectral domains (RGB daytime glare versus NIR nighttime darkness) and spatial facial states (open eyes, closed/drowsy eyes, open mouth, and closed mouth). The data acquisition explicitly excludes external obstructions, such as optical glasses, sunglasses, face masks, or heavily tinted headwear, ensuring that facial landmarks remain unobstructed during primary network optimization.

Consequently, a primary limitation of this framework lies in the lack of cross-dataset validation. Because the model has been optimized and validated strictly on this custom single-contributor pool, its out-of-distribution generalizability across diverse facial structures, gender profiles, ethnicities, and dynamic real-world environments remains unverified. Furthermore, preliminary edge deployment trials highlighted specific technical conditions under which the system fails to maintain reliable localization. As illustrated in Figure 6 observations, failure cases primarily manifest during extreme out-of-plane head rotations (yaw angles $> 45°$), where crucial facial keypoints are self-occluded from the camera's stationary field of view. Additionally, sudden non-uniform shadowing across the orbital regions during high-glare transitions and extreme proximity adjustments can induce temporary bounding box flickering. Acknowledging these failure cases, future iterations will focus on expanding the dataset matrix to integrate cross-subject validation profiles and robust geometric occlusion layers.

CONCLUSION

This study successfully demonstrates the deployment of an ONNX-optimized YOLOv11 architecture integrated with a Temporal State Machine on a resource-constrained Raspberry Pi 4B for advanced driver monitoring. By shifting from standalone PyTorch weights to an optimized ONNX graph environment, the core computational pipeline achieves a standardized benchmark of 45.9 FPS. Under live multi-spectral vehicular testing, the fully integrated system delivers a dynamic real-time frame rate ranging from 22.9 to 72.4 FPS during daytime RGB streaming, and stabilizes within a highly reliable envelope of 20.7 to 28.7 FPS under critical nighttime Near-Infrared (NIR) conditions. The implementation of a strict logical conjunction (*AND* logic) within the state machine effectively filters out micro-blinking biases, while the synchronized activation of a buzzer and a completely randomized LED flashing array provides a robust physical stimulus to disrupt confirmed driver microsleep events (t > 2seconds).

While the framework achieves its initial design objectives of non-invasive, multi-spectral edge detection above the real-time threshold of 20 FPS, several operational limitations must be acknowledged. First, the performance validation is limited to a single custom-built multi-spectral dataset, highlighting the lack of cross-dataset validation to prove broader geometric generalizations under facial obstructions like eyeglasses or masks. Second, the empirical evaluation was conducted within a static and controlled in-cabin environment, which does not fully capture the extreme mechanical vibrations and unpredictable ambient glare of high-speed open-road driving. Lastly, the physical actuation loop remains localized to standalone sensory alerts and lacks integration with actual electronic vehicle control units (ECUs) for automated speed deceleration or emergency braking. Future research will focus on deep quantization tuning, multi-environment field deployments, and standardized hardware-in-the-loop (HIL) automotive interface synchronization.

ACKNOWLEDGMENT

The authors express their highest gratitude to the Ministry of Education, Culture, Research, and Technology of the Republic of Indonesia (Kemendikbudristek) for providing financial support and opportunities through the Merdeka Belajar Kampus Merdeka (MBKM) internship program. Special thanks are also extended to the Electrical Engineering Department (Teknik Elektro) for granting full access to the Embedded Systems and Computer Vision laboratory facilities, which heavily supported the hardware integration and dataset preconditioning throughout this research

AUTHOR CONTRIBUTION STATEMENT

AB conceived the initial research idea, designed the Temporal State Machine decision logic, and performed the two-tier ONNX graph optimization on the edge hardware. DPS curated the multi-spectral image dataset, managed the manual multi-class object annotations, and conducted the cloud-based GPU model training. ATW configured the physical actuator system, including the buzzer and randomized flashing LED array, and executed the multi-illumination comparative performance testing in the simulated vehicle cabin. All authors contributed to the analysis of the experimental data, drafted the manuscript sections, and approved the final version for publication.

AI DISCLOSURE STATEMENT

The authors used ChatGPT, developed by OpenAI, during the preparation of this manuscript for language editing, grammar correction, and refinement. After using this tool, the authors carefully reviewed, revised, and validated all content to ensure its accuracy, originality, and scientific integrity. The authors are solely responsible for the content of this manuscript.

REFERENCES (11pt)

[1] C. Xu, C. Fu, and X. Jiang, "Advances in Vehicle Safety and Crash Avoidance Technologies," *Applied Sciences*, vol. 15, no. 11, p. 5955, May 2025, doi: 10.3390/app15115955.
[2] S. M. Saleem, "Risk assessment of road traffic accidents related to sleepiness during driving: a systematic review," *Eastern Mediterranean Health Journal*, vol. 28, no. 9, pp. 695–700, 2022.
[3] I. Nasri, M. Karrouchi, K. Kassmi, and A. Messaoudi, "A Review of Driver Drowsiness Detection Systems: Techniques, Advantages and Limitations," *High School of Technology, Mohammed First University*, 2022.
[4] J. Singh, R. Kanojia, R. Singh, R. Bansal, and S. Bansal, "Driver Drowsiness Detection System - An Approach By Machine Learning Application," *Journal of Pharmaceutical Negative Results*, vol. 13, no. Special Issue 10, pp. 3002–3012, 2022, doi: 10.47750/pnr.2022.13.510.361.
[5] F. Liu, D. Chen, J. Zhou, and F. Xu, "A review of driver fatigue detection and its advances on the use of RGB-D camera and deep learning," *Engineering Applications of Artificial Intelligence*, vol. 116, p. 105399, 2022, doi: https://doi.org/10.1016/j.engappai.2022.105399.
[6] Y. Albadawi, H. Takruri, and M. Awad, "A Review of Recent Developments in Driver Drowsiness Detection Systems," *Sensors*, vol. 22, no. 5, 2022.
[7] T. Fonseca and E. Al., "Drowsiness Detection in Drivers: A Systematic Review of Deep Learning Approaches," *Applied Sciences*, vol. 15, no. 16, 2025.
[8] R. M. Salman, M. Rashid, R. Roy, M. M. Ahsan, and Z. Siddique, "Driver Drowsiness Detection Using Ensemble Convolutional Neural Networks on YawDD," 2021.
[9] S. Fu and E. Al., "Advancements in the Intelligent Detection of Driver Fatigue and Distraction Based on Deep Learning," *Applied Sciences*, vol. 14, no. 7, 2024.
[10] O. F. Hassan and E. Al., "Real-Time Driver Drowsiness Detection Using Transformer Architectures and Transfer Learning," *Scientific Reports*, vol. 15, 2025.
[11] N. Lin and E. Al., "Advancing Driver Fatigue Detection in Diverse Lighting Conditions Using Deep Learning," *Scientific Reports*, vol. 14, 2024.
[12] N. Zrira and E. Al., "GCBAM-UNet: Sun Glare Segmentation Using Deep Learning," *Fire*, vol. 7, no. 6, 2024.

[13] S. Liawatimena and N. Isworo, "Annotated drowsiness detection dataset captured using Raspberry Pi 5," *Data in Brief*, vol. 63, p. 112211, 2025, doi: 10.1016/j.dib.2025.112211.
[14] R. Florez, F. Palomino-Quispe, A. B. Alvarez, R. J. Coaquira-Castillo, and J. C. Herrera-Levano, "A Real-Time Embedded System for Driver Drowsiness Detection Based on Visual Analysis of the Eyes and Mouth Using Convolutional Neural Network and Mouth Aspect Ratio," *Sensors*, vol. 24, no. 19, p. 6261, 2024, doi: 10.3390/s24196261.
[15] S. Raghavendran and E. Al., "Corneal Reflection Based Eye Tracking Technology," in *AIP Conference Proceedings*, 2023. doi: doi.org/10.1063/5.0142351.
[16] S. Abd El-Nabi, W. El-Shafai, E. S. M. El-Rabaie, K. Ramadan, and S. M. Fathi, "Machine learning and deep learning techniques for driver fatigue and drowsiness detection: a review," *Multimedia Tools and Applications*, 2023, doi: 10.1007/s11042-023-15054-0.
[17] S. Jadhav, S. Jagdale, N. Jangale, and S. Raut, "Driver Drowsiness Detection System Using Raspberry Pi," *International Journal for Research in Applied Science and Engineering Technology*, vol. 10, no. 12, 2022, doi: doi.org/10.22214/ijraset.2022.47891.
[18] A. A. D. Prasetyo and E. Al., "Smart Alarm Driver Assistance as an Early Warning of Driver Drowsiness Using Raspberry Pi 4 Model B," *Journal of Electrical Technology*, 2025.
[19] L. Zhang and Y. Liu, "Lightweight Object Detection Deployment via ONNX Quantization for Automotive Edge Computing," *Engineering Applications of Artificial Intelligence*, vol. 132, p. 107954, 2024, doi: 10.1016/j.engappai.2024.107954.
[20] J. Wang and E. Al., "Acceleration and Optimization of Deep Learning Inference on Edge Devices Using ONNX Runtime," *IEEE Access*, vol. 11, pp. 45210–45222, 2023, doi: 10.1109/ACCESS.2023.3273410.
[21] U. D. Maharani, A. S. Handayani, and L. Lindawati, "Analisis Deteksi Mata Kantuk di Wajah Pengemudi Menggunakan Support Vector Machine (SVM) Berbasis Citra Real-Time," *Building of Informatics, Technology and Science (BITS)*, vol. 6, no. 2, pp. 940–949, 2024, doi: 10.47065/bits.v6i2.5701.
[22] I. I. Maulana *et al.*, "Dampak Penggunaan Data Augmentasi Terhadap Akurasi MobileNetV2 Dalam Deteksi Mikrosleep Berbasis Rasio Aspek Mata," *Building of Informatics, Technology and Science (BITS)*, vol. 7, no. 3, pp. 1797–1808, 2025, doi: 10.47065/bits.v7i3.8719.
[23] L. Anifah, N. Nurhayati, H. Haryanto, and M. S. Zuhrie, "Automatic Microsleep Detection Approach for Car Drivers Using YOLO5 Based on Image Feature," *TEM Journal*, vol. 14, no. 3, pp. 1984–1991, Aug. 2025.
[24] A. Karakan, "Real-Time and Deep Learning-Based Fatigue Detection for Drivers," *Mugla Journal of Science and Technology*, vol. 10, no. 2, 2024, doi: 10.22531/muglajsci.1481648.
[25] K. Kotwal *et al.*, "Domain-Specific Adaptation of CNN for Detecting Face Presentation Attacks in NIR," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 4, no. 3, pp. 356–366, 2022, doi: 10.1109/TBIOM.2022.3168393.
[26] M. A. Farooq, W. Shariff, D. O'Callaghan, A. Merla, and P. Corcoran, "On the Role of Thermal Imaging in Automotive Applications: A Critical Review," *IEEE Access*, vol. 11, pp. 25147–25175, 2023, doi: 10.1109/ACCESS.2023.3255110.
[27] S. N. Tien, T. L. E. Tien, and H. Ly-thanh, "ONNX-based Architectures for Post-Training Quantization Face Detection on Edge Devices," pp. 1–11, 2026, doi: 10.15598/aeee.v24ix.251003.
[28] O. K. Akinde, T. A. Olaleye, M. O. Ibitoye, V. Rizama, S. Taiwo, and M. O. Adetona, "An Intelligent-Based Algorithm for Determining Drowsy Drivers and Prevention of Road Accidents," *Uniosun Journal of Engineering and Environmental Sciences (UJEES)*, vol. 7, no. 2, 2025, doi: 10.64980/ujees.v7i2.452.