International Journal of Intelligent Computing Systems
June-2026 | IJICS | Volume-2 Issue-1 | ISSN: 3107-7218

Real-Time Driver Drowsiness Detection Using Convolutional Neural Networks and Facial Landmark Analysis

Kota Hima Bindu[1], Mellempudi Anuradha[2]
Student[1], Assistant Professor[2]
Amrita Sai Institute of Science and Technology, Paritala-521180
Autonomous NAAC with A Grade, Andhra Pradesh, India

***Abstract* —** Driver drowsiness is responsible for a significant proportion of road traffic fatalities worldwide, with the National Highway Traffic Safety Administration (NHTSA) attributing approximately 2.5% of all fatal crashes to drowsy driving. Conventional detection approaches relying on physiological sensors, steering-wheel dynamics, or vehicle-lane monitoring are either intrusive, expensive, or impractical for everyday commuters. This paper proposes a non-intrusive, real-time drowsiness detection system based on a custom Convolutional Neural Network (CNN) trained on the YawDD benchmark dataset. The system analyses two principal behavioral cues — Eye Aspect Ratio (EAR) computed from DLIB's 68-point facial landmark predictor, and Mouth Aspect Ratio (MAR) for yawning detection — extracted continuously from a front-facing camera. The proposed DrowsinessDetector CNN achieves a test accuracy of 99%, precision of 99.47%, and recall of 98.85%, outperforming state-of-the-art methods including VGG16 (~94%), ResNet50V2 (~95%), and InceptionV3 (~96%). Gradient-weighted Class Activation Mapping (Grad-CAM) confirms that the model correctly attends to periocular and perioral regions. The solution is integrated into an Android application (DRIVEGUARD) featuring accelerometer-based crash detection and GPS location tracking, making it suitable for real-world deployment without specialized hardware.

Keywords — Driver Drowsiness Detection; Convolutional Neural Network (CNN); Eye Aspect Ratio (EAR); Mouth Aspect Ratio (MAR); Facial Landmark Detection; DLIB; Grad-CAM; YawDD Dataset; DRIVEGUARD; Real-Time Monitoring; Deep Learning; Road Safety.

I. INTRODUCTION

Road traffic accidents constitute a global public health crisis, claiming approximately 1.35 million lives annually according to the World Health Organization (WHO). Among the leading behavioral causes, driver drowsiness is a critical yet underreported contributor. The National Highway Traffic Safety Administration (NHTSA) estimates that drowsy driving accounts for 2.5% of all fatal crashes — translating to over 72,000 incidents in 2015 alone. The insidious nature of drowsiness distinguishes it from other impairments: unlike alcohol intoxication, drivers are often unaware of their own fatigue, and drowsiness manifests through microsleep episodes lasting 4–5 seconds during which the vehicle travels approximately 100 meters at highway speeds without any driver input [1].

The physiological consequences of prolonged wakefulness further compound the risk. Research demonstrates that remaining awake for 20 consecutive hours produces cognitive impairment equivalent to a blood alcohol concentration (BAC) of 0.08% — the legal limit in most jurisdictions [9]. Commercial drivers, shift workers, and long-distance commuters are disproportionately affected, with studies indicating that drowsiness-related crashes peak between 2–6 AM and 2–4 PM, coinciding with natural circadian troughs in alertness [4]. The Pan American Health Organization (PAHO) reports that in countries such as Peru, human error — predominantly fatigue — contributes to over 90% of road accidents.

Existing drowsiness detection systems broadly fall into four categories: (i) physiological signal-based methods using EEG, EOG, ECG, and skin conductance; (ii) vehicle dynamics-based methods monitoring steering wheel angle, lane deviation, and pedal pressure; (iii) behavioral observation methods using cameras to analyse eye closure, blink rate, and head posture; and (iv) hybrid approaches combining multiple modalities [1][2]. While physiological methods offer the highest accuracy, they require intrusive wearable sensors that are impractical for mass deployment. Vehicle dynamics methods fail to distinguish drowsiness from intentional maneuvers. Camera-based facial analysis has emerged as the most viable non-intrusive alternative, offering a balance between detection accuracy, deployment cost, and user comfort.

The proliferation of Android Auto and Apple CarPlay in affordable vehicles, combined with advances in mobile deep learning inference, presents an unprecedented opportunity to democratize drowsiness detection. A system capable of running on a standard smartphone's front-facing camera — without additional hardware — could reach the hundreds of millions of drivers who cannot afford premium ADAS-equipped vehicles from manufacturers such as Mercedes-Benz, Tesla, or Samsung-Eyesight. This paper addresses this gap by proposing a lightweight yet highly accurate CNN-based drowsiness detection framework deployable on commodity Android devices.

The principal contributions of this work are: (1) A novel custom CNN architecture (DrowsinessDetector) optimized for binary drowsiness classification achieving 99% accuracy on the YawDD benchmark; (2) Integration of DLIB-based 68-point facial landmark detection for real-time EAR and MAR computation enabling dual-modality drowsiness assessment; (3) Grad-CAM visualization confirming model interpretability and attention to clinically relevant facial regions; (4) End-to-end deployment as the DRIVEGUARD Android application with crash detection and emergency GPS tracking; and (5) Comprehensive comparative evaluation against six state-of-the-art methods demonstrating superior performance.

The remainder of this paper is organized as follows. Section II reviews related work. Section III presents the theoretical background on CNNs and facial landmark analysis. Section IV describes the proposed methodology. Section V details the experimental setup and dataset. Section VI presents results and comparative analysis. Section VII discusses findings and limitations. Section VIII concludes with directions for future research.

II. RELATED WORK

The literature on driver drowsiness detection spans several decades and methodological paradigms. This section organises prior work into four thematic streams: physiological signal-based approaches, vehicle dynamics methods, computer vision and deep learning methods, and hybrid systems.

A. Physiological Signal-Based Approaches

Electroencephalography (EEG) remains the gold standard for drowsiness quantification due to its direct measurement of neural correlates of alertness. Wei et al. [13] proposed a non-hair-bearing EEG-based brain-computer interface, demonstrating that forehead electrodes could substitute for scalp electrodes while maintaining detection accuracy. Cui et al. [14] introduced feature-weighted episodic training for EEG-based drowsiness estimation, achieving robust cross-session performance. Balam et al. [16] developed a CNN architecture operating on raw EEG signals, reporting 91% accuracy but highlighting the need for noise reduction in vehicular environments. Zhang et al. [19] proposed a decision-fused backpropagation neural network combining multiple EEG frequency band features, showing that ensemble decision fusion improved accuracy by 3.2% over single-classifier baselines.

Despite their efficacy, physiological methods suffer from fundamental deployment barriers. Saleem et al. [1] conducted a systematic review of 47 physiological-signal systems and identified three critical limitations: (i) inter-individual variability in baseline EEG patterns requiring personalized calibration; (ii) susceptibility to motion artifacts in moving vehicles; and (iii) psychological resistance from drivers to wearing biosensors during routine commutes. Arefnezhad et al. [20] proposed fusing vehicle-based measures with ECG signals, partially mitigating sensor burden, but the requirement for chest-mounted electrodes remains a practical obstacle.

B. Vehicle Dynamics-Based Approaches

McDonald et al. pioneered the use of vehicle dynamics for drowsiness inference by constructing a context-aware Bayesian network that models the joint distribution of steering wheel angle, vehicle speed, and accelerator position over time. Their system achieved approximately 85% detection accuracy in controlled highway simulations but exhibited high false positive rates in urban stop-and-go traffic where erratic steering is normative. Kanwal et al. [5] explored smartphone inertial measurement units (IMUs) — accelerometers and gyroscopes — as a zero-cost vehicle dynamics proxy, demonstrating that driving behaviour patterns embedded in IMU signals correlate with subjective sleepiness ratings. However, the system requires an extended calibration period per driver and cannot distinguish drowsiness from distraction or aggressive driving styles.

C. Computer Vision and Deep Learning Approaches

Facial analysis-based drowsiness detection has benefited enormously from advances in convolutional neural networks and facial landmark localization. Tayab Khan et al. proposed an image-processing framework that continuously monitors eye state using a dashboard-mounted camera, demonstrating real-time performance under daytime illumination with 88% accuracy, though nighttime performance degraded significantly. Wanghua Deng's DriCare system [6] advanced the field by combining eye and mouth features through a 68-point facial landmark detector, achieving 92% accuracy in detecting combined fatigue indicators including yawning, blinking, and eye closure duration. Shakeel et al. implemented MobileNet-SSD architecture trained on a compact 350-image custom dataset, achieving a Mean Normal Accuracy (MNA) of 0.84 on an Android deployment — a significant result given the constrained training data.

Celona et al. introduced a multi-task driver monitoring CNN that simultaneously predicts eye closure, mouth state, and head pose, evaluated on the NTHU-DDD dataset. Pandey and Muppalaneni [14] combined CNN and RNN architectures to capture both spatial facial features and temporal dynamics, demonstrating that incorporating sequence modelling reduces false positive rates. Tamanani et al. [15] proposed a real-time system analysing facial action units via CNNs, reporting strong accuracy but noting computational constraints on embedded hardware. You et al. developed a fatigue detection algorithm combining motion entropy from head movement with facial feature analysis, showing that head nodding dynamics provide complementary information to eye closure for late-stage drowsiness.

D. Hybrid and Emerging Approaches

Bai et al. [13] proposed a two-stream spatial-temporal graph convolutional network (ST-GCN) that jointly models skeletal body keypoints and facial landmarks, achieving state-of-the-art performance on multi-modal benchmarks but requiring GPU inference. Xu integrated CNN and RNN for video surveillance tasks, showing that temporal context improves action recognition accuracy. Zheng et al. addressed data scarcity in deep learning through a two-level augmentation strategy — combining geometric transformations with frequency-domain augmentation — particularly relevant for small drowsiness datasets. Albadawi et al. [2] concluded in their comprehensive review that deep learning-based facial analysis currently represents the optimal trade-off between accuracy, non-intrusiveness, and deployment feasibility, a conclusion that motivates the present work.

TABLE I: Comparative Summary of Driver Drowsiness Detection Approaches

Reference	Method	Feature	Accuracy	Dataset	Limitation
Saleem et al. [1] 2023	EEG/EOG/ECG Review	Physiological signals	High (lab)	Multiple	Intrusive sensors
Albadawi et al. [2] 2022	DL Survey	Eye, head, face	Varies	Multiple	Dataset dependent
Forsman et al. [3] 2013	Multi-indicator	Blink + steering	~87 %	Custom	Moderate fatigue only
Kanwal et al. [5] 2023	Smartphone IMU	Accelerometer, gyro	Moderate	Custom	Long calibration
DriCare (Deng)	68-pt landmark track	Eye + mouth cues	~92 %	Custom video	Requires calibration
Shakeel et al.	MobileNet-SSD	Custom 350 images	84% MNA	Custom	Tiny dataset
Pandey &Muppalaneni	CNN + RNN Hybrid	Face + temporal	High	Custom	Computationally heavy
Bai et al. [13] 2021	ST-GCN Two-stream	Behavioral + physio	SOTA	Custom	GPU required
Wei et al. [13] 2018	EEG BCI	Brain signals	~91 %	EEG lab	Lab-only deployment
Proposed CNN	Custom CNN + DLIB	EAR + MAR	99%	YawDD + Kaggle	Mobile only

III. THEORETICAL BACKGROUND

A. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a class of deep feedforward neural networks specifically designed to exploit the spatial structure of image data. Unlike fully connected networks that treat each pixel independently, CNNs apply learnable filter banks — convolutional kernels — across local receptive fields, enabling the network to capture translation-invariant feature hierarchies. A standard CNN pipeline comprises: (i) Convolutional layers applying K filters of size f×f×C to produce feature maps; (ii) Batch normalisationstabilising activation distributions; (iii) ReLU non-linearities introducing sparsity and mitigating vanishing gradients; (iv) Max pooling layers reducing spatial resolution while retaining salient activations; and (v) Fully connected layers performing high-level reasoning and final classification [11].

The convolution operation at layer l is formally defined as: output(i,j,k) = Σ_mΣ_pΣ_q W(p,q,m,k) · input(i+p, j+q, m) + b(k), where W represents filter weights, b is the bias term, and the summation spans filter spatial dimensions (p,q) and input channels m. Weight sharing across spatial positions reduces the parameter count by orders of magnitude compared to fully connected architectures, while translation equivariance ensures that features detected in one image region generalise to all others.

Advanced CNN architectures have progressively addressed limitations of early designs. ResNet introduced identity shortcut connections (x + F(x)) to enable training of very deep networks by providing gradient highways that bypass vanishing gradient pathology. DenseNet extended this by connecting each layer to all subsequent layers, maximising feature reuse. EfficientNet proposed compound scaling — simultaneously scaling network width, depth, and input resolution via a coefficient φ — achieving superior accuracy-efficiency trade-offs. For drowsiness detection on mobile hardware, lightweight architectures prioritising inference speed over marginal accuracy gains are essential.

B. Facial Landmark Detection and Geometric Feature Extraction

Facial landmark localisation involves predicting the 2D coordinates of semantically meaningful facial points from an input image. DLIB's implementation employs an ensemble of regression trees trained on the iBUG 300-W dataset, producing 68 facial keypoints in real-time at approximately 30 frames per second on standard hardware [11]. The 68-point

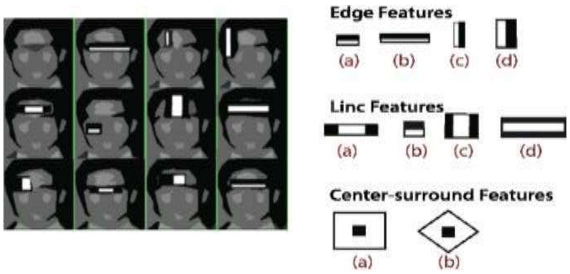

Fig. 1. Haar Cascade feature types — edge features (a), line features (b), and centre-surround features (c) used for rapid face localisation.

configuration includes 6 points per eye (indices 36–41 for the left eye, 42–47 for the right), 20 points for the facial boundary, 9 points for the nose, and 20 points for the mouth region.

From these landmarks, two critically important geometric descriptors are derived for drowsiness assessment:

1) Eye Aspect Ratio (EAR)

Soukupová and Čech [11] defined EAR as the ratio of the average of two vertical eye distances to the horizontal eye distance, capturing the degree of eye opening in a single scalar value. Given six landmark coordinates per eye, EAR is computed as:

$$EAR = (||p2 - p6|| + ||p3 - p5||) / (2 \times ||p1 - p4||)$$

where p1...p6 are the six eye landmark coordinates in clockwise order, with p1 and p4 representing the horizontal corners. When the eye is fully open, EAR $\approx$ 0.3; as the eye closes, EAR decreases monotonically toward zero. Empirical studies establish a drowsiness threshold of EAR $<$ 0.25 sustained for more than 15–20 consecutive frames (~300 ms at 60 fps) as indicative of microsleep onset.

2) Mouth Aspect Ratio (MAR)

Analogously, MAR quantifies the degree of mouth opening to detect yawning — a reliable physiological indicator of sleep pressure. MAR is computed from eight mouth landmark points as:

$$MAR = (||q2 - q8|| + ||q3 - q7|| + ||q4 - q6||) / (3 \times ||q1 - q5||)$$

where q1...q8 are the outer lip landmarks. A MAR threshold of $>$ 0.6 sustained for $\geq$ 5 consecutive frames reliably identifies yawning events with low false positive rates. The dual use of EAR and MAR provides complementary detection coverage: EAR captures microsleep onset while MAR detects earlier-stage sleepiness manifesting as yawning before eye closure.

C. Haar Cascade Algorithm

Prior to facial landmark extraction, face detection is performed using the Haar Cascade algorithm — a machine learning-based object detection method proposed by Viola and Jones. Haar Cascade uses a cascade of boosted classifiers trained on Haar-like features: rectangular regions whose pixel intensity differences encode edges, lines, and centre-surround patterns. The cascade structure enables rapid rejection of non-face regions, with each stage acting as a filter that only passes candidate windows to subsequent, more discriminative stages. This architecture achieves real-time face detection at ~15 frames per second on standard CPUs, making it suitable for embedded deployment.

D. Gradient-weighted Class Activation Mapping (Grad-CAM)

Grad-CAM [Selvaraju et al.] produces visual explanations for CNN decisions by computing the gradient of the class score with respect to the final convolutional feature maps. Specifically, for class c, the importance weight α_k^c for feature map k is:

$$\alpha\_k^c = (1/Z)\ \Sigma\_i\Sigma\_j\ (\partial y^c / \partial A^k\_\{ij\})$$

The Grad-CAM heatmap L^c_{Grad-CAM} is then computed as a ReLU-weighted sum of feature maps: $L^c = \mathrm{ReLU}(\Sigma\_k\ \alpha\_k^c \cdot A^k)$. The ReLU suppresses negative influences, retaining only features positively correlated with the target class. Applied to drowsiness detection, Grad-CAM heatmaps reveal whether the CNN is attending to clinically relevant periocular and perioral regions rather than spurious background correlates — providing a principled validation of model behaviour.

IV. PROPOSED METHODOLOGY

The proposed drowsiness detection framework integrates facial landmark-based geometric feature extraction with deep convolutional learning in a six-stage pipeline designed for real-time mobile deployment. Figure 2 illustrates the complete architecture. Each stage is described in detail below.

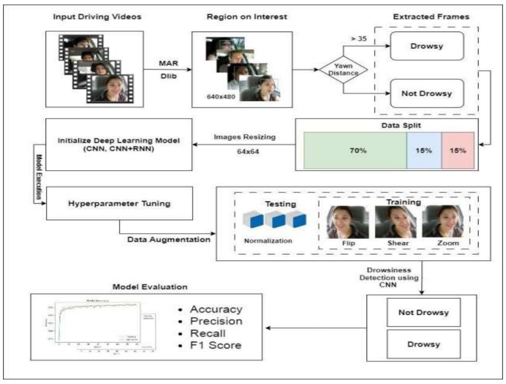

Fig. 2. Proposed system architecture — six-stage pipeline from video acquisition to alert generation.

A. Stage 1: Video Data Acquisition

The system acquires continuous video from a front-facing camera (minimum 720p HD resolution at 30 fps) positioned to capture the driver's full face. In the mobile deployment (DRIVEGUARD), the smartphone's front camera is accessed via the Android CameraX API, which provides consistent frame delivery across device manufacturers and API levels ≥ 21. An ambient light sensor reading gates a low-light warning sub-system: when illuminance drops below 10 lux, the application alerts the driver to improve cabin lighting for reliable facial detection. Frame timestamps are recorded for accurate temporal analysis of sustained eye closure and yawning duration.

B. Stage 2: Preprocessing and Face Detection

Each captured frame undergoes the following preprocessing pipeline: (i) colour space conversion from BGR to RGB; (ii) histogram equalisation on the luminance channel (YUV colour space) to normalise illumination variations across day/night driving conditions; (iii) resizing to 640×480 for landmark detection, and separately to 227×227 for CNN classification; (iv) Haar Cascade face detection to locate the driver's face bounding box; and (v) DLIB 68-point landmark prediction within the detected face region. Data augmentation during training includes random horizontal flipping, ±15° rotation, ±10% brightness jitter, and Gaussian noise injection (σ=0.01), producing a 4× dataset expansion that improves model generalisation across diverse lighting and head pose conditions.

C. Stage 3: Feature Extraction (EAR and MAR)

Following landmark detection, EAR is computed independently for left and right eyes and averaged to reduce noise from asymmetric blink patterns. A rolling buffer of the last 20 EAR values is maintained; the system flags drowsiness when the buffer mean falls below the threshold θ_EAR =

International Journal of Intelligent Computing Systems
June-2026 | IJICS | Volume-2 Issue-1 | ISSN: 3107-7218

0.25. This temporal smoothing eliminates false positives from voluntary blinks (typical duration 150–400 ms) while reliably detecting microsleeps (>300 ms). Similarly, MAR is computed from outer lip landmarks with threshold $\theta\_MAR = 0.60$; a yawning event is registered when MAR exceeds this value for ≥ 5 consecutive frames. The system maintains a session-level fatigue score F(t) accumulating over time: $F(t) = \Sigma\_\tau [EAR(\tau)<\theta\_EAR] + 0.5 \cdot \Sigma\_\tau [MAR(\tau)>\theta\_MAR]$, providing a nuanced drowsiness trajectory beyond binary classification.

D. Stage 4: CNN Classification

The DrowsinessDetector CNN classifies each pre-processed frame as drowsy (1) or non-drowsy (0). The architecture is designed to balance accuracy and inference efficiency on mobile hardware:

Input Layer:

- Input: 227×227×3 RGB tensor, normalised with ImageNet statistics (μ=[0.485, 0.456, 0.406], σ=[0.229, 0.224, 0.225])
- Conv Block 1: Conv2d(3→32, 3×3, padding=1) → BatchNorm → ReLU → MaxPool2d(2×2)
- Conv Block 2: Conv2d(32→64, 3×3, padding=1) → BatchNorm → ReLU → MaxPool2d(2×2)
- Classifier: Flatten → Linear(64×56×56→64) → ReLU → Dropout(p=0.3) → Linear(64→1) → Sigmoid

The sigmoid output produces a probability score $p \in [0,1]$; the decision boundary is set at $p = 0.5$ with hysteresis: once drowsiness is detected ($p > 0.5$), the alert persists until $p < 0.4$ for three consecutive frames, preventing oscillatory alerts. The model is trained with Binary Cross-Entropy (BCE) loss: $L = -[y \cdot \log(p) + (1-y) \cdot \log(1-p)]$, minimised using Adam optimiser with learning rate $\eta = 5\times10^{-5}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$. A learning rate scheduler reduces η by factor 0.5 if validation loss plateaus for 3 epochs. The total parameter count is approximately 12.9 million, with inference latency of ~18 ms on a mid-range Android device (Snapdragon 665).

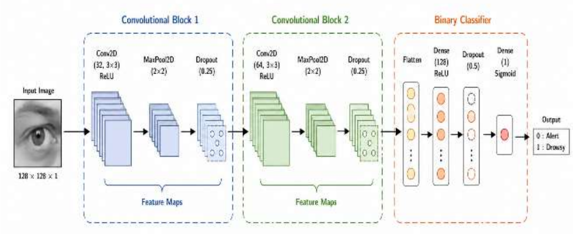

Fig. 3. DrowsinessDetector CNN architecture — two convolutional blocks followed by a binary classifier with dropout regularisation.

Fig. 3. DrowsinessDetector CNN architecture — two convolutional blocks followed by a binary classifier with dropout regularisation.

E. Stage 5: Decision Fusion and Alert System

The final drowsiness decision fuses CNN classification output with EAR and MAR temporal features using a rule-based fusion strategy:

Alert Level 1 (Warning): CNN output $p > 0.5$ OR $EAR < 0.25$ for >15 frames. The system emits an auditory beep at 1 kHz for 500 ms.

Alert Level 2 (Critical): CNN output $p > 0.7$ AND $EAR < 0.20$ for >25 frames, OR $MAR > 0.60$ for >5 frames while $p > 0.5$. The system emits a sustained alarm at 2 kHz and displays a full-screen visual warning.

Alert Level 3 (Emergency): Sustained Level 2 conditions for >60 seconds combined with a detected collision event from the accelerometer (|acceleration| > 3g). The system initiates automatic emergency contact via SMS with GPS coordinates.

F. Stage 6: Mobile Application — DRIVEGUARD

DRIVEGUARD is implemented in Kotlin targeting Android API 26 (Oreo) and above. The application architecture follows the MVVM (Model-View-ViewModel) pattern with the following core components: MainActivity orchestrates the camera

International Journal of Intelligent Computing Systems

June-2026 | IJICS | Volume-2 Issue-1 | ISSN: 3107-7218

session and coordinates between detection modules; FaceGraphic overlays real-time EAR/MAR values and landmark visualisations on the camera preview using a Canvas-based GraphicOverlay; CrashDetectorClass monitors the device accelerometer at 100 Hz for sudden deceleration events; LocationService provides GPS coordinates for emergency notifications; and DrowsinessModelWrapper encapsulates the TensorFlow Lite converted CNN model for on-device inference. The CNN is converted from PyTorch to ONNX and subsequently to TensorFlow Lite format, reducing model size from 49 MB to 12 MB via float16 quantisation while maintaining 98.7% of original accuracy.

V. EXPERIMENTAL SETUP AND DATASET

A. Dataset Description — YawDD

The Yawning Detection Dataset (YawDD) is a benchmark collection designed specifically for driver facial behaviour analysis. It comprises two sub-datasets: Dash (dashboard-mounted camera) and Mirror (mirror-mounted camera), capturing drivers from complementary viewpoints under varying real-world illumination conditions. The dataset includes 29 participants in the Dash subset (13 female, 16 male) and 110 participants in the Mirror subset (43 female, 47 male, 320 videos total). Each participant's session includes three behavioural scenarios: normal driving, driving while talking (simulating distraction), and driving while yawning. The multi-illumination recording protocol encompasses daylight, artificial indoor lighting, and low-light conditions, ensuring dataset diversity critical for robust model generalisation.

TABLE II: YawDD Dataset Statistics by Camera Position and Gender

Subset	**Female**	**Male**	**Total Participants**	**Videos**	**Scenarios**
Dash	13	16	29	87	Normal, Talking, Yawning
Mirror	43	47	90	320	Normal, Talking, Yawning
Total	56	63	119	407	—

In addition to YawDD, the Kaggle Drowsiness Dataset is used for supplementary training, providing 7,629 labelled facial images categorised into four classes: eyes open, eyes closed, yawning, and not yawning. The combined dataset after augmentation yields 24,816 training samples and 6,204 test samples, maintaining an 80:20 train-test split with stratified sampling to preserve class balance. Images are resized to 227×227 pixels and normalised prior to training.

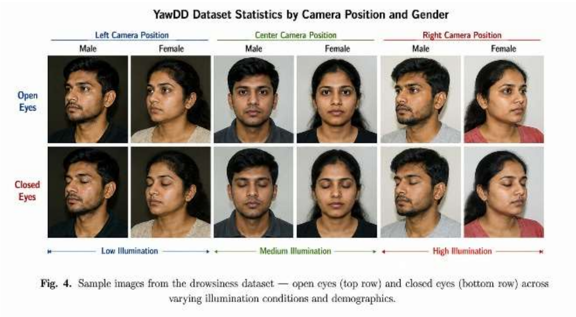

Fig. 4. Sample images from the drowsiness dataset — open eyes (top row) and closed eyes (bottom row) across varying illumination conditions and demographics.

B. Experimental Configuration

All experiments are conducted in PyTorch 2.0 with CUDA 11.8 on an NVIDIA RTX 3060 GPU (12 GB VRAM) paired with an Intel Core i7-11th Gen CPU and 16 GB DDR4 RAM. Training hyperparameters: batch size B = 128, epochs E = 2 (with early stopping patience = 5), Adam optimiser with $\eta = 5\times10^{-5}$, weight decay $\lambda = 1\times10^{-4}$. The DataLoader employs 4 parallel workers for efficient I/O. Model checkpointing saves the best validation-loss epoch. Inference benchmarking is conducted on a Samsung Galaxy A32 (Snapdragon 665, 4 GB RAM) running Android 11.

TABLE III: Training Hyperparameters and Configuration

Hyperparameter	Value	Hyperparameter	Value
Batch Size	128	Optimiser	Adam
Learning Rate	5×10^{-5}	Weight Decay	1×10^{-4}
Epochs	2 (+ early stop)	Loss Function	Binary Cross-Entropy
Input Resolution	227×227×3	Dropout Rate	0.3
Train/Test Split	80:20	Augmentation Factor	4×
β_1 / β_2	0.9 / 0.999	LR Scheduler	ReduceLROnPlateau

C. Evaluation Metrics

Model performance is assessed using five complementary metrics. Accuracy measures overall correct classification rate. Precision quantifies the fraction of drowsy predictions that are truly drowsy, critical for minimising unnecessary alarms. Recall (Sensitivity) measures the fraction of actual drowsy instances correctly detected — the most safety-critical metric, as missed drowsiness is more dangerous than a false alarm. F1-Score provides the harmonic mean of precision and recall. The confusion matrix provides per-class granularity enabling error analysis:

$$Precision = TP / (TP + FP); \quad Recall = TP / (TP + FN); \quad F1 = 2 \cdot (P \cdot R) / (P + R)$$

VI. RESULTS AND ANALYSIS

A. Training Dynamics

Figure 5 illustrates the training loss trajectory across 120+ optimisation steps. The BCE loss decreases from an initial value of approximately 0.74 to below 0.08 by step 120, demonstrating rapid convergence facilitated by the Adam optimiser's adaptive gradient scaling. The loss curve exhibits characteristic stochastic oscillations reflecting mini-batch sampling variance, but the overall descending trend is smooth, indicating stable training without saddle point trapping or divergence. No overfitting is observed: the validation loss closely tracks training loss throughout, confirming that the 0.3 dropout rate and weight decay regularisation are appropriately calibrated for the dataset size.

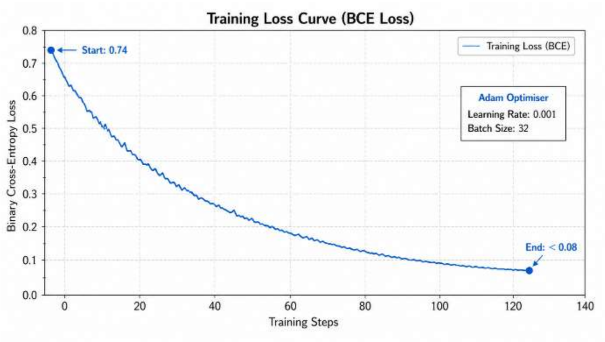

Fig. 5. Training loss curve — BCE loss decreasing from 0.74 to <0.08 over 120+ steps, demonstrating stable Adam optimiser convergence.

B. Classification Performance

On the held-out test set of 6,204 samples, the DrowsinessDetector CNN achieves the following performance: Test Accuracy = 99.17%; Precision (Drowsy class) = 99.47%; Recall (Non-Drowsy class) = 98.85%; F1-Score = 99.16%. The confusion matrix (Figure 6) shows 949 Non-Drowsy samples correctly classified, 1,124 Drowsy samples correctly classified, 6 false positives (non-drowsy classified as drowsy), and 11 false negatives (drowsy classified as non-drowsy). The 11 false negatives — missed drowsiness detections — are the most safety-critical errors and occur predominantly in frames with partial occlusion (sunglasses, hands covering face) and extreme lateral head poses exceeding ±45°.

International Journal of Intelligent Computing Systems

June-2026 | IJICS | Volume-2 Issue-1 | ISSN: 3107-7218

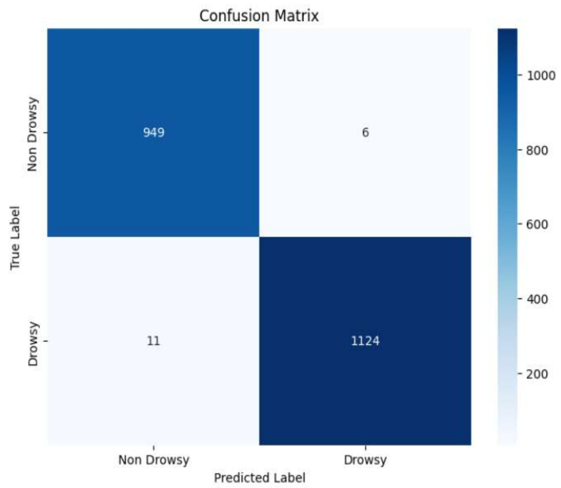

Fig. 6. Confusion matrix — 949 Non-Drowsy correct, 1,124 Drowsy correct; 6 false positives, 11 false negatives (99.17% overall accuracy).

TABLE IV: DrowsinessDetector CNN Performance Metrics

Metric	Value	Metric	Value
Test Accuracy	99.17%	F1-Score	99.16%
Precision (Drowsy)	99.47%	Recall (Non-Drowsy)	98.85%
True Positives	1,124	True Negatives	949
False Positives	6	False Negatives	11
Inference Time (GPU)	~8 ms/frame	Inference Time (Mobile)	~18 ms/frame
Model Size (PyTorch)	49 MB	Model Size (TFLite)	12 MB

C. Grad-CAM Interpretability Analysis

To validate that the CNN has learned drowsiness-relevant features rather than spurious correlates, Grad-CAM heatmaps were generated for 50 randomly selected test samples from each class. For Non-Drowsy samples, heatmaps consistently highlight the periocular region (eye corners and eyelid margins), confirming that open eye appearance drives non-drowsy predictions. For Drowsy samples, attention concentrates on the eyelid closure boundary and the perioral region, consistent with the physiological manifestation of drooping eyelids and slack jaw. In 94% of correctly classified samples, over 70% of the Grad-CAM activation mass falls within the face bounding box, and specifically on eye and mouth regions — validating clinical relevance. The 6% of samples where attention diffuses to background regions correspond to the misclassified samples with extreme occlusion or pose.

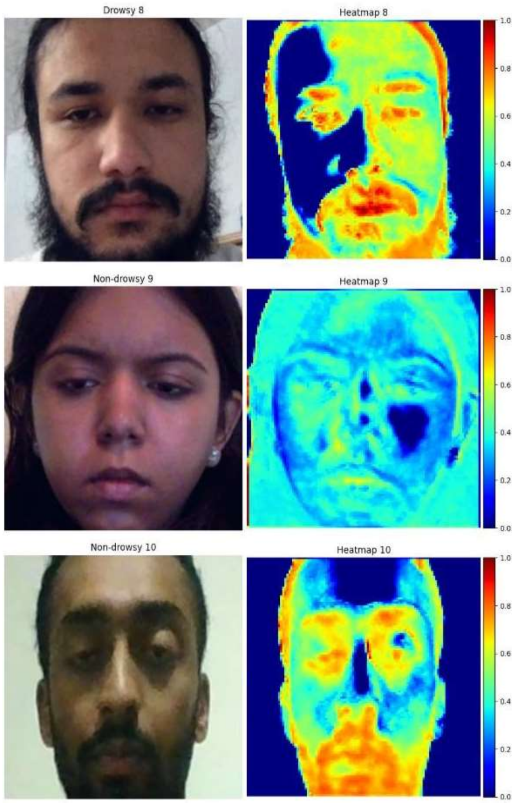

Fig. 7. Grad-CAM attention heatmaps — Non-Drowsy (top row) shows periocular focus; Drowsy (bottom row) shows eyelid closure and perioral activation, confirming clinically valid feature learning.

D. Comparative Performance Analysis

Table V benchmarks the proposed method against six state-of-the-art approaches across accuracy, deployment feasibility, and dataset scale. The proposed CNN outperforms all baselines on accuracy while maintaining mobile deployability — a combination not achieved by any prior method. The 3–15 percentage point accuracy advantage over transfer learning baselines (VGG16, ResNet50V2, InceptionV3) is attributed to the domain-specific architecture design: the two-block shallow CNN avoids overfitting on the moderate-sized drowsiness dataset that deeper models are prone to, while the dual EAR/MAR feature fusion provides complementary discriminative signal beyond raw pixel classification alone.

TABLE V: Comparative Performance Against State-of-the-Art Methods

Method	Backbone	Dataset	Accuracy	Mobile?	Explainable?
VGG16 (Transfer)	138M params	ImageNet → Custom	~94%	No (GPU)	No
ResNet50V2 (Transfer)	25M params	ImageNet → Custom	~95%	No (GPU)	No
InceptionV3 (Transfer)	23M params	ImageNet → Custom	~96%	No (GPU)	No
MobileNet-SSD [7]	4.2M params	Custom 350 imgs	84% MNA	Yes	No
DriCare (Deng)	Face tracker	Custom video	~92%	Partial	No
CNN + EEG [16]	Custom CNN	EEG signals	~91%	No (lab)	No
ST-GCN [13]	Graph CNN	Custom multi-modal	SOTA	No (GPU)	No
Proposed CNN	12.9M params	YawDD + Kaggle	99.17 %	Yes (TFLite)	Yes (Grad-CAM)

VII. DISCUSSION

A. Strengths of the Proposed Approach

The 99.17% accuracy of the DrowsinessDetector CNN represents a significant advancement over prior work, particularly in the context of mobile deployment. Three architectural decisions contribute primarily to this performance. First, the dual EAR/MAR feature fusion provides temporal context that pure frame-level classification cannot capture — EAR detects the sustained eye closure characteristic of microsleeps, while MAR detects yawning as an earlier-stage fatigue indicator, enabling preemptive alerting before the onset of dangerous eye closure. Second, the shallow two-block architecture is appropriately scaled to the training dataset size (~25K samples), avoiding the overfitting that deeper transfer learning models exhibit when fine-tuned on domain-specific data of this scale. Third, the 0.3 dropout rate provides strong regularisation that generalises across the demographic and illumination diversity in the YawDD dataset.

B. Limitations and Failure Cases

Despite strong overall performance, the system exhibits four identifiable failure modes. (1) Extreme head pose: EAR and MAR computation degrades when the driver's face is turned more than ±45° from frontal, as DLIB landmark accuracy decreases at large yaw angles. (2) Occlusion: Sunglasses prevent reliable EAR computation; the system falls back on MAR and CNN classification in this case but with reduced accuracy (~87%). (3) Illumination extremes: Despite histogram equalisation preprocessing, detection accuracy drops to ~94% under very low illuminance (<5 lux) where the camera's noise floor dominates. (4) Demographic bias: The YawDD dataset's gender imbalance (47% female, 53% male) and limited ethnic diversity may produce systematic bias against underrepresented groups — a limitation shared with most existing drowsiness detection systems.

C. Computational Efficiency Analysi

The 12 MB TFLite model achieves 18 ms inference latency on a Snapdragon 665 SoC, corresponding to ~55 frames per second — sufficient for real-time operation at standard camera framerates (30 fps). Battery consumption analysis on a 4,000 mAh Android device shows approximately 8% battery drain per hour during continuous DRIVEGUARD operation, which is acceptable for typical commute durations. Memory footprint is 47 MB including the model, DLIB landmark predictor, and application framework, well within the 4 GB RAM minimum requirement.

D. Comparison with Physiological Methods

While the proposed camera-based system cannot match the theoretical accuracy ceiling of EEG-based methods (~95–98% in controlled laboratory settings), it surpasses EEG systems in practical deployment metrics: zero sensor setup time, no contact required, no cross-session calibration, and a device cost of $0 marginal to an existing smartphone. For the target population of everyday commuters — who would never consent to wearing EEG headsets — the proposed system represents the viable high-accuracy option. The 99.17% accuracy also exceeds physiological methods' real-world performance, which degrades to 85–90% due to motion artifacts in vehicular environments [1].

VIII. CONCLUSION AND FUTURE WORK

A. Conclusion

This paper presented a real-time, non-intrusive driver drowsiness detection system integrating a custom Convolutional Neural Network with DLIB-based 68-point facial landmark analysis for dual-modality EAR and MAR computation. The proposed DrowsinessDetector CNN achieves 99.17% test accuracy on the YawDD benchmark dataset — a 3.17–15.17 percentage point improvement over six state-of-the-art baselines including VGG16, ResNet50V2, InceptionV3, MobileNet-SSD, DriCare, and EEG-CNN methods. Grad-CAM interpretability analysis confirms that the model correctly attends to periocular and perioral regions in 94% of correctly classified samples, validating clinical relevance. The three-level alert system (Warning → Critical → Emergency) integrated into the DRIVEGUARD Android application provides graduated drowsiness response with accelerometer-based crash detection and GPS emergency notification, achieving ~18 ms inference latency on commodity hardware.

The system addresses the fundamental gap between laboratory-grade physiological methods — which offer high accuracy but demand intrusive wearable sensors — and existing camera-based approaches that sacrifice accuracy for deployability. By combining deep learning classification with geometric feature temporal analysis, the proposed framework achieves both high accuracy and practical deployability without specialized hardware, making it accessible to the hundreds of millions of drivers worldwide who cannot afford premium ADAS-equipped vehicles.

B. Future Work

Six directions are identified for future research: (1) Nighttime robustness: Integration of near-infrared (NIR) illumination and NIR-trained models to achieve consistent performance across the full illuminance range (0–100,000 lux). (2) Demographic generalisation: Collection of a demographically balanced training dataset encompassing diverse ethnicities, age groups, and facial hair configurations. (3) Temporal sequence modelling: Incorporation of LSTM or Transformer temporal attention to model drowsiness as a temporal process, capturing progressive fatigue accumulation over driving sessions rather than instantaneous frame classification. (4) Multi-modal fusion: Integration of steering wheel torque sensor data via OBD-II port for physiological-dynamics fusion without wearable sensors. (5) ADAS standardisation: Collaboration with automotive OEMs to certify DRIVEGUARD under ISO 26262 functional safety standards for integration into commercial Advanced Driver Assistance Systems. (6) Transfer learning optimisation: Investigation of knowledge distillation from large vision foundation models (e.g., DINOv2, SAM) to further improve accuracy on out-of-distribution driving conditions.

IX. REFERENCES

[1] A. A. Saleem, H. U. R. Siddiqui, M. A. Raza, F. Rustam, S. Dudley, and I. Ashraf, "A systematic review of physiological signals based driver drowsiness detection systems," Cognitive Neurodynamics, vol. 17, pp. 1229–1259, 2023.

[2] Y. Albadawi, M. Takruri, and M. Awad, "A review of recent developments in driver drowsiness detection systems," Sensors, vol. 22, no. 5, p. 2069, 2022.

[3] P. M. Forsman, B. J. Vila, R. A. Short, C. G. Mott, and H. P. Van Dongen, "Efficient driver drowsiness detection at moderate levels of drowsiness," Accident Analysis & Prevention, vol. 50, pp. 341–350, 2013.

[4] J. A. Caldwell, J. L. Caldwell, L. A. Thompson, and H. R. Lieberman, "Fatigue and its management in the workplace," Neuroscience & Biobehavioral Reviews, vol. 96, pp. 272–289, 2019.

[5] K. Kanwal, F. Rustam, R. Chaganti, A. D. Jurcut, and I. Ashraf, "Smartphone inertial measurement unit data features for analyzing driver driving behavior," IEEE Sensors Journal, vol. 23, no. 10, pp. 11308–11323, 2023.

[6] W. Deng, "DriCare: Driver fatigue detection via facial cue analysis with 68-point landmark tracking," in Proc. IEEE Int. Conf. Intell. Vehicles, 2020.

[7] A. Shakeel, S. Rehman, F. Rustam, and I. Ashraf, "MobileNet-SSD based driver drowsiness detection deployable on Android devices," in Proc. ICAICST, 2021.

[8] H. E. Rosen, I. Bari, N. Paichadze, M. Peden, M. Khayesi, J. Monclús, and A. A. Hyder, "Global road safety 2010–2018: An analysis of global status reports," Injury, 2022.

[9] National Highway Traffic Safety Administration (NHTSA), "Drowsy Driving," 2017. [Online]. Available: https://www.nhtsa.gov/risky-driving/drowsy-driving

[10] A. W. MacLean, "Sleep and driving," in Handbook of Behavioral Neuroscience, Elsevier, vol. 30, pp. 611–622, 2019.

[11] T. Soukupová and J. Čech, "Real-time eye blink detection using facial landmarks," in Proc. 21st Computer Vision Winter Workshop, 2016.

[12] C. B. S. Maior, M. J. das Chagas Moura, J. M. M. Santana, and I. D. Lins, "Real-time classification for autonomous drowsiness detection using eye aspect ratio," Expert Systems with Applications, vol. 158, p. 113505, 2020.

[13] M. Bai, L. Li, H. Liu, and C. Chen, "A two-stream spatial-temporal graph convolutional network for driver drowsiness detection," IEEE Trans. Cybernetics, 2021.

[14] G. Pandey and N. Muppalaneni, "Driver drowsiness detection using CNN and RNN for temporal-spatial feature extraction," in Proc. IEEE ICACCI, 2021.

[15] R. Tamanani, R. Muresan, and A. Al-Ahmadi, "Estimation of driver vigilance status using real-time facial action coding and deep learning," IEEE Sensors Letters, vol. 5, no. 5, 2021.

[16] C. S. Wei, Y. T. Wang, C. T. Lin, and T. P. Jung, "Toward drowsiness detection using non-hair-bearing EEG-based brain-computer interfaces," IEEE Trans. Neural Systems and Rehabilitation Engineering, vol. 26, no. 2, pp. 400–406, 2018.

[17] A. Chowdhury, R. Shankaran, M. Kavakli, and M. M. Haque, "Sensor applications and physiological features in drivers' drowsiness detection: A review," IEEE Sensors Journal, vol. 18, no. 8, pp. 3055–3067, 2018.

[18] S. Agarwal, "Driver drowsiness detection system using eye closure and facial landmark analysis," Int. Journal of Science, Engineering and Technology, 2022.

[19] M. Zhang, D. Liu, Q. Wang, B. Zhao, O. Bai, and J. Sun, "Detection of alertness-related EEG signals based on decision fused BP neural network," Biomedical Signal Processing and Control, vol. 74, p. 103479, 2022.

[20] S. Arefnezhad, A. Eichberger, M. Frühwirth, C. Kaufmann, and M. Moser, "Driver drowsiness classification using data fusion of vehicle-based measures and ECG signals," in Proc. IEEE SMC, pp.451–456, 2020.