

Intelligent Driver Monitoring and Fleet Safety Management System for Commercial Vehicles

Vallapureddy Shanmuka Nandini¹, Ruchika S. V.², Vaishnavi³

¹²³Dept. of Computer Science & Engineering, Sir M. Visvesvaraya Institute of Technology, Bengaluru, India,
Under the Guidance of Mr. Shreyas Sridhar, Assistant Professor, Department of Data Science,

Abstract - Driver fatigue and impairment contribute to road-crash risk, and commercial fleets also need remote visibility of driver state. We describe a working prototype of a connected driver-monitoring and fleet-safety system and its ongoing extension into a multimodal, connectivity-resilient architecture. The implemented prototype combines a MediaPipe Face Landmarker front end with per-frame Eye Aspect Ratio (EAR), Mouth Aspect Ratio (MAR), sliding-window PERCLOS, blink, microsleep, yawn and head-nod measures; a rule-based fusion into an operational, KSS-like 1–9 score modelled on the Karolinska Sleepiness Scale; a three-tier alert manager; ArcFace-based driver authentication; and a FastAPI/WebSocket backend with a React/Tailwind owner dashboard for drivers, vehicles, trips, alerts, analytics and live monitoring. An LSTM fatigue predictor has a training pipeline but is not yet in the live decision path. Current software evidence comprises 43 named unit/functional test cases for the drowsiness engine (software checks on synthetic landmarks and injected signals, not real-world experiments), further named tests for supporting services, deterministic analysis of the scoring function, and an LSTM trained and validated only on synthetic driver sessions (validation KSS MAE 0.158; critical-crossing accuracy 0.860), which does not represent real-driver performance. No real-driver dataset is used, and no real-driver accuracy, latency or throughput is claimed. Offline-first operation with deferred telemetry synchronisation, an infrared camera, an MQ-3 alcohol-related sensor and a CNN module are under development and unevaluated; we specify them and a planned real-driver evaluation.

Key Words: driver drowsiness detection, eye aspect ratio, PERCLOS, head pose, Karolinska Sleepiness Scale, offline-first, fleet telemetry, multimodal sensing

1. INTRODUCTION

Camera-based driver monitoring is non-contact and can act before vehicle-dynamics symptoms such as lane departure appear; assessment programmes now specify driver-state detection criteria [1]. Three gaps motivate this work. Single-cue detectors (e.g., an isolated blink counter) are sensitive to speech, glances and eyelid anatomy, so several cues should be combined over time. Many connected designs place alerting behind a network link, whereas a safety alert should not depend on Internet connectivity even though remote reporting inevitably does.

Finally, fleet operators need centralised trip, incident and driver records.

The project is not finished, so every component and claim carries one of four status labels: IMPLEMENTED (present in the project code), SYNTHETIC (evaluated only on synthetic signals or sessions, not real-driver evidence), UNDER DEVELOPMENT (being built, not part of the evaluated prototype) and PLANNED (evaluation to follow implementation).

Contributions: (1) an IMPLEMENTED interpretable landmark-based pipeline with time-based event detection and a rule-based operational KSS (Sections 4–5); (2) an IMPLEMENTED connected-fleet stack with authentication, trip lifecycle, incident logging, a FastAPI/WebSocket backend and an owner dashboard (Sections 3, 8); (3) an LSTM training pipeline evaluated on SYNTHETIC sessions only, with baselines and explicit caveats (Sections 5, 9); (4) the UNDER DEVELOPMENT offline-first, infrared, MQ-3 and CNN extensions with a PLANNED evaluation plan (Sections 6–7, 11). No real-driver dataset is used in this paper, and no real-driver accuracy, latency or throughput is claimed.

2. RELATED WORK

Ocular and sleepiness measures. The EAR [2] gives a scale-insensitive blink signal from eyelid landmarks. PERCLOS, the proportion of time the eyes are closed beyond a criterion (classically 80%), originates from driving-simulator work [3] and was promoted as an alertness measure in an FHWA brief [4]; our PERCLOS is an EAR-threshold proxy, not the calibrated P80 measure. The KSS is a nine-point self-reported scale [5]. The Euro NCAP driver-engagement protocol [1] asks systems to detect a driver reaching a KSS level above 7 “at the latest” (or an equivalent metric), defines microsleep (about 1–2 s of eye closure) and sleep (about 3 s or more), and lists noise factors such as lighting down to about 1 lux, eyewear and talking; we use these as design context, not as compliance claims.

Landmarks, pose, infrared and learning. MediaPipe [6] and its face-mesh model [7] provide dense landmarks on commodity hardware; head pose and gaze estimation are established driver-monitoring components [8], [9]. Active near-infrared illumination has long been used for

eye tracking in poor light [10]. Public datasets include NTHU-DDD [11], UTA-RLDD [12] and YawDD [13]. Keeping availability-critical computation near the data source is a general edge-computing principle [14] that we apply to the alert path.

3. CURRENT SYSTEM ARCHITECTURE

Fig. 1 shows the target architecture with implementation status: the implemented blocks form a vision-to-alert chain and a fleet backend, while the synchronisation layer and extra modalities are under development.

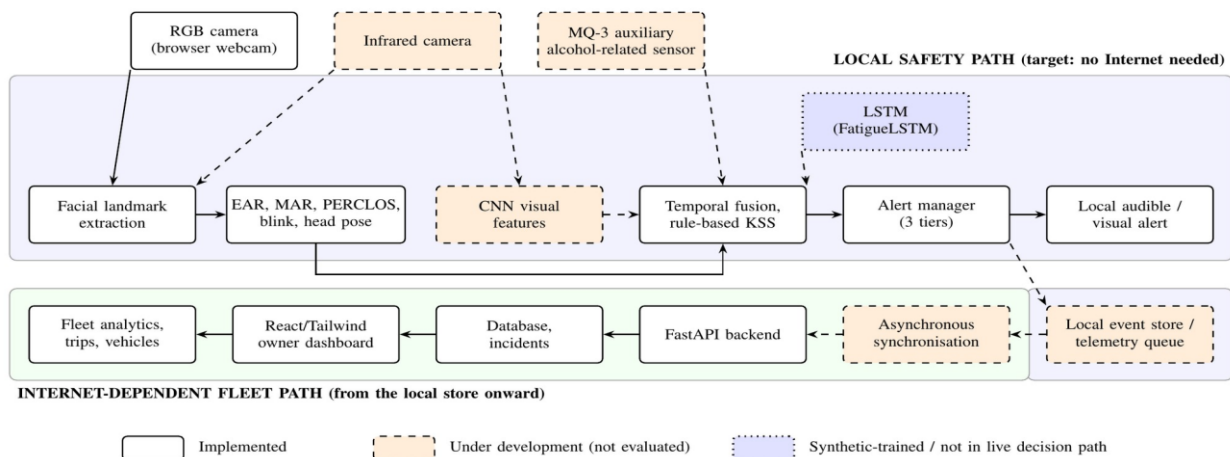


Fig -1: Target multimodal, connectivity-resilient architecture with implementation status; dashed arrows involve blocks that are under development or synthetic-only. In the current prototype the vision chain and alert manager execute inside the backend process that receives browser frames over a WebSocket (Section 3.1); the local event store, synchronisation, infrared camera, MQ-3 sensor and CNN do not yet exist, and the LSTM is not used for live decisions.

3.1 Prototype deployment topology (IMPLEMENTED)

The prototype is a software system exercised with a laptop webcam; no in-vehicle deployment has been performed. A browser client captures 640×480 frames, JPEG-encodes them (quality 0.6) and sends them over a WebSocket to a FastAPI service, one frame in flight at a time with at least about 85 ms spacing (an offered-rate bound of roughly 12 frames/s, an implementation characteristic of this browser/WebSocket topology and not a measured throughput). The server runs OpenCV [15] and the MediaPipe Face Landmarker [6], [7], updates the trackers of Section 4, computes the operational KSS, applies the alert manager and returns metrics, an annotated frame and any alert; the client plays tier-specific audio tones. In this topology the alarm depends on the client-backend link, harmless when both share one machine but exactly the dependency the offline-first work removes. The MediaPipe bundle is downloaded on first run and InsightFace models are loaded from a local directory (fetched by the library when absent), so offline deployment must pre-provision both.

3.2 Driver authentication (IMPLEMENTED, prototype)

Identity uses 512-dimensional InsightFace/ArcFace embeddings [16]. Registration averages several frames

(default 10, minimum 5) and rejects likely duplicates at cosine similarity 0.55; check-in performs a one-to-many cosine search with an untuned acceptance threshold of 0.45 and opens a monitoring session bound to driver, vehicle and trip. No false-accept/false-reject rates have been measured, no liveness or anti-spoofing check exists, and the search is not scoped to one fleet owner. Face embeddings are sensitive personal data under regimes such as the GDPR [17] and India's DPDP Act [18]; this is a deployment consideration (Section 10), and the prototype is not production-grade biometric security.

3.3 Backend and persistence (IMPLEMENTED)

REST routers cover owner authentication, drivers, vehicles, assignments, monitoring sessions (create, pause, resume, end), incidents, trips, alerts and analytics, plus a session-token-authenticated WebSocket for live monitoring. Persistence uses SQLAlchemy with Alembic migrations (SQLite by default, PostgreSQL via the connection URL). Owner accounts use Argon2id password hashes and JWT session cookies, and records are scoped by owner.

Table -1: Configuration parameters of the prototype (IMPLEMENTED).

Parameter	Value
EAR closed-eye / MAR open-mouth threshold	0.21 / 0.55
PERCLOS window; warning / critical scoring level	60 s; 0.15 / 0.40
Microsleep / yawn minimum duration	1.5 s / 1.5 s
Head-nod minimum duration; pitch	1.0 s; < -15°
KSS critical threshold	7 (1-9 scale)
Alert cooldown (per tier)	8 s
LSTM horizon / step / sequence	300 s / 5 s / 24 steps
Face match / duplicate threshold	0.45 / 0.55 (cosine)

4. MULTI-FEATURE VISUAL MONITORING

All quantities are computed per frame from facial landmarks. The thresholds in Table 1 are engineering configuration parameters, not clinically validated and not tuned on real-driver data.

EAR. For each eye, six contour landmarks $p_1, \dots, p_6$ give [2]

$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2 \|p_1 - p_4\|}. \quad (1)$$

The eyes are averaged into EAR_{avg} and an eye is closed when $EAR_{avg} < 0.21$. No per-driver calibration is implemented.

PERCLOS and blinks. With timestamped closed/open samples, PERCLOS is the closed fraction over the last 60 s,

$$PERCLOS(t) = \frac{1}{|W_t|} \sum_{k \in W_t} 1[EAR_{avg,k} < 0.21], \quad (2)$$

where W_t is the sample set in $[t - 60 \text{ s}, t]$. It is scaled down during the first 20 s to suppress start-up spikes, faceless frames count as open, and being time-based it is independent of frame rate. Blinks are counted on closed-to-open transitions. This is an EAR proxy for the 80%-closure PERCLOS [3], [4].

MAR and yawns. MAR is the ratio of mean vertical lip separations to mouth width on a fixed subset of eight lip landmarks; a yawn is $MAR > 0.55$ held for at least 1.5 s. Only synthetic landmark arrays have exercised this computation; verifying the landmark pairing and threshold on real yawns is pending (Section 10).

Head pose. Six landmarks (nose tip, chin, eye and mouth corners) are matched to a generic 3D face model with OpenCV's iterative PnP solver, approximate intrinsics (focal length equal to image width, centred principal point, no distortion) and Euler angles wrapped to $[-90^\circ, 90^\circ]$. Pitch below -15° indicates a dropping head and drives head-nod detection. Yaw and roll are reported but not used in alert logic, so distraction detection is not implemented; pose accuracy is unmeasured.

Event detectors. Microsleep, yawn and head-nod events fire when their condition holds continuously for the minimum duration, re-arm after a 1.5 s cooldown and reset on face loss. The merged runtime passes EAR_{avg} to the microsleep detector; a both-eyes-closed variant exists only in the methodology notebook, so single-eye-occlusion robustness is not established for the runtime.

5. TEMPORAL FATIGUE FUSION AND LSTM

5.1 Rule-based operational KSS (IMPLEMENTED)

Let P be PERCLOS, b the blink rate per minute, τ the current sustained-closure time in seconds, and m, y, n the microsleep, yawn and head-nod counts in the last 60 s. The operational, KSS-like score $K \in [1, 9]$, intended to be read on the KSS scale [5], is

$$K = \text{clip} \left(1 + 3.5 \text{sat} \left(\frac{P-0.10}{0.30} \right) + 0.8 \text{sat} \left(\frac{\delta-0.3}{0.7} \right) + c(\tau) + 2 \text{sat} \left(\frac{m}{2} \right) + \text{sat} \left(\frac{y}{2} \right) + 1.5 \text{sat} \left(\frac{n}{2} \right), 1, 9 \right), \quad (3)$$

with $\text{sat}(u) = \min(1, \max(0, u))$, $\delta = |b-16|/16$, and $c(\tau) = 2$ for $\tau \geq 1.5 \text{ s}$, 3.5 for $\tau \geq 2.5 \text{ s}$, else 0. The weights are hand-set and were not fitted to human KSS ratings; $K \geq 7$ is the prototype's operational critical threshold. This is an operational index, not a clinically validated estimate of self-reported sleepiness, and the value 7 is a design choice (the protocol wording cited above refers to levels above 7 [1]).

5.2 FatigueLSTM pipeline (IMPLEMENTED pipeline; SYNTHETIC evaluation)

A fusion engine aggregates observations into 5 s steps and keeps 24 steps ($\approx 2 \text{ min}$) of six features: normalised PERCLOS, normalised blink rate, microsleep rate, yawn rate, head-nod rate and the rule-based KSS. A recurrent network [19] maps this 24×6 sequence to a current-KSS estimate and the probability of reaching the critical level within 300 s; without trained weights the design falls back to the rule-based score. The LSTM is not invoked in the current live path, which uses the instantaneous rule-based score; wiring its output into alerts is UNDER DEVELOPMENT.

No labelled real-driver sequence dataset exists in the project, so the training script generates 400 synthetic sessions of 240 five-second steps. Each follows a hidden KSS trajectory (drifting random walk with noise and occasional recovery) from which the six features are generated as noisy functions. Each 24-step window has a current-KSS target and a binary target for the hidden KSS reaching 7 within the next 60 steps, giving 62,400 windows. Training uses Adam [20] (learning rate 10^{-3} , batch 64, 12 epochs), a 15% validation split, MSE loss for KSS and binary cross-entropy for the critical flag. The notebook records a validation KSS MAE of 0.158 and critical-crossing accuracy of 0.860. Because features are generated from the variable that defines the labels, these figures measure inversion of the synthetic generator, not real-driver accuracy (see Section 9.2).

6. CONNECTIVITY-RESILIENT, OFFLINE-FIRST OPERATION (UNDER DEVELOPMENT)

Status. The reviewed prototype contains no store-and-forward queue or synchronisation service; this section specifies the architecture being implemented, and nothing here has been evaluated.

Principle. Two paths are separated (Fig. 1). The local safety path (capture, facial processing, features, temporal fusion, KSS, alert manager, audible/visual alert and local event storage) is to run on a vehicle-side computer with no Internet dependency (“edge-side safety inference”). The fleet path (local event store → asynchronous synchronisation → FastAPI backend → database → owner dashboard) carries telemetry, incidents and evidence and may be delayed or interrupted. This is the target design: the current prototype has the alert manager inside the backend process (Section 3.1) and no completed store-and-forward service.

Intended behaviour. During an outage, monitoring and alerts continue locally and important incidents are written to durable local storage. After reconnection an asynchronous synchronisation agent uploads stored events in order with identifiers so retries do not duplicate them, the backend accepts late events with their original timestamps, and the dashboard is updated and shows stale/last-sync state rather than implying live data. Further work items are pre-provisioned model bundles, offline driver check-in with a protected local template cache, and prioritising critical incidents in the queue. Existing local persistence mechanisms provide a starting point, but the durable store-and-forward telemetry queue and synchronization service required for offline operation are not yet implemented.

Planned evaluation (PLANNED). Fault-injection tests will disconnect and restore the link during trips and measure whether local alert latency is unaffected, loss and

duplication of queued events (the goal being no loss of critical incidents), synchronisation delay versus outage length and queue size, and behaviour under repeated disconnections and restarts.

7. ONGOING MULTIMODAL HARDWARE AND CNN EXTENSIONS (UNDER DEVELOPMENT)

Each extension targets a specific limitation of the landmark pipeline (Table 2).

Table -2: Intended role and status of each modality/component.

Component	Intended role	Status
RGB camera	Normal facial/ocular monitoring	Implemented
Infrared camera	Low-light robustness	Under development
EAR, MAR, PERCLOS	Interpretable ocular and oral fatigue cues	Implemented
Head pose	Nodding and attention information	Implemented (pitch used)
CNN	Learned visual representation	Under development
LSTM	Temporal context, future-fatigue prediction	Synthetic only; not in live path
MQ-3	Auxiliary alcohol-related signal	Under development
Offline engine	Safety continuity without Internet	Under development
Fleet dashboard	Remote monitoring, incident management	Implemented

Infrared camera. The infrared sensing module is under development and will be experimentally compared against the visible-light camera under controlled illumination conditions. It is intended to improve visibility of facial and ocular features at night and in low light. It is unknown whether the visible-light-trained landmark model stays reliable on infrared frames. PLANNED experiments use paired RGB/IR capture in daylight, low light, night, glare, shadow and with eyewear, measuring landmark availability, agreement of blink and microsleep events with manual annotation, and false alerts per hour. No improvement is claimed.

MQ-3 alcohol-related sensor. The MQ-3 is a low-cost metal-oxide gas sensor whose datasheet specifies a heated element, a limited operating temperature/humidity range and a long pre-heat period [21]. It is planned as an

auxiliary alcohol-related safety sensing modality, not a certified breathalyser; the system will not estimate blood alcohol concentration. Its signal is intended to become another input to the multimodal safety decision layer, most plausibly as a separate flag rather than mixed into the sleepiness score. The dashboard's sensor panel and the alcohol-status field of the trips API are non-functional placeholders. Before any accuracy statement, work is needed on warm-up, placement, airflow, temperature and humidity dependence, cross-sensitivity, drift, calibration against a reference instrument, false positives and negatives, and quantitative validation. No accuracy is reported.

CNN visual module. The current chain is facial landmarks $\rightarrow$ EAR/MAR/PERCLOS/head pose $\rightarrow$ interpretable rules; the planned complementary chain is camera $\rightarrow$ CNN representation or classification. The CNN module is under development and will be evaluated after implementation using an appropriate labelled dataset. The PLANNED comparison is (A) landmark-based, (B) CNN-based and (C) multimodal fusion, with subject-independent splits on public datasets [11]–[13] and on the authors' own data, reporting accuracy and edge-hardware compute cost. No CNN performance is claimed. Rule-based logic remains the alert arbiter so decisions stay auditable; any fusion weights will be set from real data, not from the synthetic generator.

8. ALERT MANAGEMENT AND OWNER DASHBOARD

8.1 Three-tier alert manager (IMPLEMENTED)

The manager evaluates the conditions of Table 3 in priority order each frame. These are the current implemented logic: fixed values inside the alert manager, distinct from the configuration thresholds of Table 1 and from the closure classes in the literature cited in Section 2, from which they are not derived. Mechanisms limiting repeated alerts are a per-tier 8 s cooldown, a latched critical state suppressing duplicates until recovery (eyes reopen, no sustained closure), state tracking, and a full reset when the face is lost or a trip is paused/resumed. Alerts are also logged locally; hazard/SOS actuation is simulated and disabled by default. Critical events store a composite evidence image and a database incident; in the current path only critical events are persisted, while nudges and warnings go to the driver client and the local log.

Table -3: Implemented alert conditions (IMPLEMENTED).
 τ : closure time; P: PERCLOS; K: rule-based KSS; m, y: microsleeps, yawns in last 60 s.

Tier	Conditions (any)
Critical	$\tau \geq 2.5$ s; $\tau \geq 1.5$ s with pitch $< -15^\circ$ or a sustained yawn; $m \geq 2$ with $P \geq 0.35$ and $K \geq 7.5$
Warning	$\tau \geq 1.8$ s; sustained nod (≥ 1.5 s, pitch $< -15^\circ$); $y \geq 2$ with $K \geq 6.0$; $P \geq 0.30$ with $K \geq 6.5$
Nudge	$\tau \geq 1.5$ s; sustained yawn (≥ 1.5 s); $P \geq 0.15$ with $K \geq 5.0$

8.2 Owner / fleet dashboard (IMPLEMENTED)

The owner dashboard, a React/Tailwind application reading from the backend API, is the fleet-facing half of the system. Overview: fleet indicators (registered drivers, active sessions, infractions, a fleet safety index), a seven-day incident history, a live alert feed and active drivers. Drivers and vehicles: directory, profiles, vehicle management and assignment. Live monitoring: per-session EAR, MAR, PERCLOS, head pose, KSS with a status label, and pause state, polled once per second from a metrics-only cache (no video is exposed to the owner view). Trips: duration, paused time, alert counts, peak KSS and a session safety score. Alerts/incidents: incident list with evidence images and level filtering. Analytics: alerts by driver, score distributions and trends. The session safety score subtracts weighted alert counts (critical 8, warning 3, nudge 1) from 100; the weights are heuristic and unvalidated. The current stack is driver client $\rightarrow$ WebSocket $\rightarrow$ backend/database $\rightarrow$ owner dashboard; the offline-first design of Section 6 is intended to decouple the local safety path from this network-dependent dashboard path, which is not yet done. Vehicle location is simulated (fixed coordinates, no GPS hardware), origin/destination fields are placeholders, and non-production configurations can seed demo records that the interface labels as such.

9. CURRENT VALIDATION AND EXPERIMENTAL EVIDENCE

9.1 Tests and synthetic demonstrations

The repository contains 43 named functional/unit test cases for the drowsiness engine, covering EAR, MAR, PERCLOS, head-pose wrapping, event detection, KSS, fusion aggregation, alert escalation, thresholds, cooldowns, duplicate suppression and recovery, plus further named tests for supporting services (authentication, database, owner accounts, APIs, trip pause/resume). They use synthetic landmarks and injected signals and verify implemented behaviour, not detection performance; we report no pass count. The methodology notebook

demonstrates microsleep detection on a synthetic EAR timeline, single-eye occlusion versus sustained two-eye closure, pose wrapping, KSS scenarios and LSTM training/inference, using shorter notebook-local durations than Table 1. On a synthetic timeline with one injected 4 s closure, its short-duration detector reported several events, so short thresholds also fire on heavy-blink bursts. These are not real-world experiments.

9.2 Synthetic LSTM evaluation and baselines (SYNTHETIC)

The 62,400 generated windows were divided using a 15% validation split, yielding 9,360 validation windows. Table 4 sets the recorded LSTM results beside baselines we recomputed from the released generator (numpy, same seed and split as the training script; LSTM figures are those recorded in the notebook, not re-run).

Table -4: Synthetic-session results (validation, 9,360 windows). Not real-driver performance.

Method	KSS MAE	Critical acc.
Majority class / predict train mean	1.71	0.525
Threshold on last-step rule-KSS feature	–	0.850
Linear regression, last-step features	0.324	–
FatigueLSTM (recorded)	0.158	0.860

The critical label is nearly balanced (47.5% positive), so 0.860 is not a class-imbalance artefact, but one threshold on a single input feature (chosen on the training split) already reaches 0.850, only 0.010 below the LSTM, so the experiment does not show a benefit of temporal modelling for the critical flag; the LSTM's KSS error is lower than the non-temporal linear baseline. Caveats: labels and features share one generative process; the validation split is at window level, so windows sharing 23 of 24 steps from the same sessions appear in both training and validation; there is one run and one seed; the generator's rule-KSS feature uses a simplified linear formula differing from the runtime mapper, so trained weights would face a train/deploy mismatch; and the trained weights are not part of the packaged project.

9.3 Deterministic analysis of the rule-based KSS (IMPLEMENTED)

Evaluating the implemented scoring function over sweeps (no driver data) shows that PERCLOS alone cannot exceed $K = 4.5$, even at 60% (Fig. 2). A 2.5 s closure with PERCLOS 40% gives $K = 8.0$, and two microsleeps plus two head nods reach $K = 7$ at PERCLOS of about 31.5%. Reaching the critical level thus requires convergence of

several cues, consistent with the design intent but unchecked against human ratings. Figure 2 is an analysis of the implemented scoring function rather than a measurement of driver behaviour.

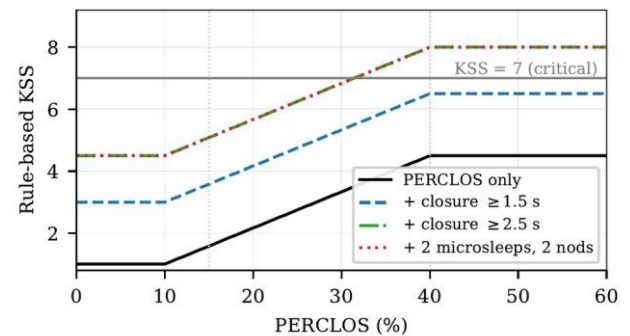


Fig -2: Rule-based KSS from the implemented scoring function versus PERCLOS with additional cues fixed (deterministic; no driver data).

9.4 Evidence status

Table 5 summarises every component; real-driver evaluation has not been done.

Table -5: Component status and available evidence.

Component	Status	Evidence
Vision drowsiness pipeline	Implemented	Unit tests, synthetic demos
Rule-based KSS	Implemented	Tests, sweep (Fig. 2)
LSTM training pipeline	Implemented	Notebook run
Synthetic LSTM evaluation	Synthetic	Synthetic sessions only
Owner dashboard	Implemented	API tests
Fleet telemetry	Implemented	API tests
Offline-first monitoring	Under development	None
Internet-loss synchronisation	Under development	None
Infrared camera	Under development	None
MQ-3 sensor	Under development	None
CNN module	Under development	None
Real-driver evaluation	Planned	None

10. DISCUSSION AND LIMITATIONS

All numeric evidence is synthetic or deterministic; detection accuracy, false-alarm rate, latency and throughput on real drivers and target hardware are unmeasured, and the offered rate of about 12 frames/s limits resolution for short blinks. Thresholds lack per-driver calibration, PERCLOS is an EAR proxy, the operational KSS is hand-weighted and the dashboard safety score is heuristic. Known gaps: MAR pairing and threshold await verification on real yawns; the runtime microsleep detector uses averaged EAR; yaw and roll do not drive alerts; the LSTM is not in the live path; the telemetry "confidence" field is a constant placeholder; only critical incidents are persisted; and owner live monitoring is polling-based. Raw frames travel from browser to backend; in the prototype both run on one machine, but a remote backend would expose video to the network, which the offline-first design avoids. Face templates are biometric data: consent, retention, encryption at rest, transport security, liveness detection and owner-scoped matching are required before deployment [17], [18]. We make no compliance claim, and the MQ-3 is not a breathalyzer.

11. FUTURE WORK

(i) PLANNED Collect labelled real-driver data with informed consent and ethics review (expert annotation, periodic self-reported KSS) and report event-level precision/recall, alert latency, false alerts per hour and KSS agreement with subject-independent splits and per-cue ablations; also benchmark on public datasets [11]–[13]. (ii) UNDER DEVELOPMENT Complete the vehicle-side host, durable queue and synchronisation and run the fault-injection tests of Section 6. (iii) UNDER DEVELOPMENT Finish the infrared, MQ-3 and CNN modules and run the comparisons of Section 7. (iv) Retrain the LSTM on real sequences with session-level splits, align training and runtime features, and integrate it into alerts. (v) Verify MAR landmark pairing, add per-driver calibration, use yaw for distraction, persist all alert tiers, add real GPS and harden security.

12. CONCLUSION

The project currently provides a working connected driver-monitoring and fleet prototype with interpretable landmark-based drowsiness monitoring, an operational KSS-like score, three-tier alerts, ArcFace-based driver authentication, an owner dashboard and fleet telemetry. Current quantitative evidence is synthetic or deterministic, and real-driver accuracy has not been established. Offline-first monitoring with deferred synchronisation, infrared sensing, an MQ-3 auxiliary sensor and a CNN module are ongoing extensions toward a multimodal, connectivity-resilient design; they are not part of the evaluated

prototype and will be evaluated only after implementation, on real data.

REFERENCES

- [1] Euro NCAP, "Safe driving – driver engagement protocol, version 1.2," Jul. 2026, [Online]. Available: https://cdn.euroncap.com/cars/assets/Euro_NCAP_Protocol_Safe_Driving_Driver_Engagement_v1_2_ebce03a443.pdf.
- [2] T. Soukupová and J. Čech, "Real-time eye blink detection using facial landmarks," in Proc. 21st Comput. Vis. Winter Workshop (CVWW), Rimske Toplice, Slovenia, Feb. 2016.
- [3] W. W. Wierwille, S. S. Wreggit, C. L. Kirn, L. A. Ellsworth, and R. J. Fairbanks, "Research on vehicle-based driver status/performance monitoring: Development, validation, and refinement of algorithms for detection of driver drowsiness," Nat. Highway Traffic Saf. Admin., Washington, DC, USA, Final Rep. DOT HS 808 247, Dec. 1994.
- [4] D. F. Dinges and R. Grace, "PERCLOS: A valid psychophysiological measure of alertness as assessed by psychomotor vigilance," Federal Highway Admin., Washington, DC, USA, Tech Brief FHWA-MCRT-98-006, Oct. 1998.
- [5] T. Åkerstedt and M. Gillberg, "Subjective and objective sleepiness in the active individual," Int. J. Neurosci., vol. 52, no. 1–2, pp. 29–37, 1990, doi: 10.3109/00207459008994241.
- [6] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann, "MediaPipe: A framework for building perception pipelines," arXiv preprint arXiv:1906.08172, 2019.
- [7] Y. Kartynnik, A. Ablavatski, I. Grishchenko, and M. Grundmann, "Real-time facial surface geometry from monocular video on mobile GPUs," arXiv preprint arXiv:1907.06724, 2019.
- [8] F. Vicente, Z. Huang, X. Xiong, F. De la Torre, W. Zhang, and D. Levi, "Driver gaze tracking and eyes off the road detection system," IEEE Trans. Intell. Transp. Syst., vol. 16, no. 4, pp. 2014–2027, Aug. 2015.
- [9] A. Sahayadhas, K. Sundaraj, and M. Murugappan, "Detecting driver drowsiness based on sensors: A review," Sensors, vol. 12, no. 12, pp. 16937–16953, 2012.

- [10] Q. Ji and X. Yang, "Real-time eye, gaze, and face pose tracking for monitoring driver vigilance," *Real-Time Imaging*, vol. 8, no. 5, pp. 357–377, 2002.
- [11] C.-H. Weng, Y.-H. Lai, and S.-H. Lai, "Driver drowsiness detection via a hierarchical temporal deep belief network," in *Computer Vision – ACCV 2016 Workshops*. Springer, 2017, pp. 117–133.
- [12] R. Ghoddoosian, M. Galib, and V. Athitsos, "A realistic dataset and baseline temporal model for early drowsiness detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2019, pp. 178–187.
- [13] S. Abtahi, M. Omidyeganeh, S. Shirmohammadi, and B. Hariri, "YawDD: A yawning detection dataset," in *Proc. 5th ACM Multimedia Syst. Conf. (MMSys)*, 2014, pp. 24–28.
- [14] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.
- [15] G. Bradski, "The OpenCV library," *Dr. Dobb's J. Softw. Tools*, 2000.
- [16] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 4690–4699.
- [17] European Parliament and Council of the European Union, "Regulation (EU) 2016/679 of 27 April 2016 (general data protection regulation)," *Official Journal of the European Union*, L 119, May 2016.
- [18] Government of India, "The digital personal data protection act, 2023 (act no. 22 of 2023)," Aug. 2023.
- [19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.
- [20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, 2015.
- [21] Hanwei Electronics Co., Ltd., Technical Data: MQ-3 Gas Sensor, datasheet. [Online]. Available: <https://cdn.sparkfun.com/assets/6/a/1/7/b/MQ-3.pdf>.