iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** X **Month of publication:** October 2026

DOI: https://doi.org/10.22214/ijraset.2026.85519

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

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

VISION: Vehicle Intelligence System for Incident Observation and Navigation

Mrs. K. Yazhini[1], Sabin Rahul I[2], Sanjay R[3], Senathipathi T[4], Sheshash S D[5]
[1]*Assistant Professor (Project Guide),* [2, 3, 4, 5]*B.E. Students (Reg. Nos. 6176AC23UCS133, 6176AC23UCS138, 6176AC23UCS143, 6176AC23UCS147),*
Department of Computer Science and Engineering, Adhiyamaan College of Engineering (An Autonomous Institution), Dr. M.G.R. Nagar, Hosur – 635130, Tamil Nadu, India

Abstract: ***Rising vehicle traffic volume on modern highways and metropolitan corridors has significantly increased accident rates, reckless speeding, unauthorized wrong-way entries, and life-threatening transit delays for emergency responders. Conventional commercial navigation applications (e.g., Google Maps, Apple Maps, Waze) subject drivers to severe cognitive distraction by rendering dense 3D graphics, buildings, advertisements, and complex map geometries that pull the driver's visual gaze away from the forward roadway for seconds at a time. Furthermore, existing mobile navigation platforms lack unified priority arbitration when multiple hazards occur concurrently and frequently interpolate or fabricate unverified speed limits on unmapped roads. This paper presents VISION (Vehicle Intelligence System for Incident Observation and Navigation), an automotive digital instrument cluster HUD and incident awareness platform engineered as a high-performance web-based telematics framework for immediate operational deployment, with an architecturally decoupled roadmap for dedicated in-vehicle IoT integration. The platform pairs a cross-platform Flutter Web application with a Node.js, Express, and TypeScript edge telematics backend. The system streams live GPS coordinates, heading, and velocity from client browsers, dynamically querying OpenStreetMap (Nominatim and Overpass) and Project OSRM in parallel. A deterministic Seven-Level Alert Priority Engine evaluates simultaneous risks (wrong-way driving, no-entry restrictions, approaching emergency vehicles, overspeeding, school/hospital zones, and lane optimization), selecting the single most critical life-safety advisory for display on a dominant central message panel commanding 45%–55% of the screen. Crucially, VISION implements a strict zero-hallucination policy, explicitly displaying "SPEED LIMIT DATA UNAVAILABLE" when unmapped. A dynamic ambulance corridor module tracks approaching emergency vehicles within 800 meters, calculating geodesic relative bearings to issue actionable steering evasions ("MOVE LEFT WHEN SAFE"). An abstract VehicleDataSource layer allows the current browser GPS to be seamlessly substituted by an ESP32 hardware module with CAN-bus integration in future iterations without altering dashboard presentation logic. Experimental evaluation across ten real-world scenarios demonstrates sub-second edge arbitration, robust velocity filtering, and elimination of cockpit visual clutter.***

Index Terms: ***Vehicle intelligence system, Advanced Driver Assistance Systems (ADAS), Heads-Up Display (HUD), alert prioritization, emergency vehicle corridor, GPS telematics, OpenStreetMap, Flutter Web, Intelligent Transportation Systems (ITS).***

I. INTRODUCTION

A. Background

Road transportation remains the lifeline of modern socio-economic mobility, yet traffic collisions constitute one of the leading global causes of preventable trauma and mortality [7]. In fast-urbanizing nations such as India, dense mixed-traffic flows are exacerbated by excessive vehicle speeds, missing or occluded roadside signage, inadvertent wrong-way maneuvers onto high-speed divided bypasses, and traffic congestion that severely obstructs emergency ambulances during the golden hour of medical care [3].

To navigate unfamiliar road topologies, modern motorists depend almost exclusively on smartphone-mounted consumer mapping applications. While effective for macroscopic trip routing, these consumer tools present severe cognitive hazards when used as real-time driving aids. They crowd the cockpit with 3D buildings, dynamic camera pans, commercial points of interest, and visual advertisements. Human-factors research shows that decoding dense maps requires 2.0 to 4.5 seconds of off-road glance time. At 60 km/h, a driver glances away blind for over 50 meters, drastically inflating crash risk. Furthermore, existing applications lack unified priority arbitration when multiple hazards occur concurrently and routinely extrapolate unverified speed limits on unmapped rural or arterial roads.

©IJRASET: All Rights are Reserved

B. Objectives

The objective of this research is to engineer a distraction-free, automotive-grade driver assistance platform that replaces visual map clutter with a minimalist instrument cluster HUD. Specifically, the system: (i) presents only the single most critical message to the driver using a deterministic Seven-Level Alert Priority Engine; (ii) captures real vehicle speed and heading from browser and mobile GPS streams, applying dynamic noise gating and exponential smoothing to eliminate stationary drift; (iii) compares speed against a multi-source verified hierarchy, explicitly displaying "SPEED LIMIT DATA UNAVAILABLE" rather than fabricating speeds; (iv) flags wrong-way driving against one-way vector angles and warns before entering no-entry zones; (v) tracks emergency ambulances up to 800 m away, issuing directional steering evasions ("MOVE LEFT WHEN SAFE"); and (vi) decouples data ingestion behind an abstract VehicleDataSource interface, enabling a web prototype today and a plug-and-play ESP32 CAN-bus hardware module tomorrow.

C. Contributions

The primary contributions of this work are: (1) a non-map automotive HUD focused on a dominant Central Message Panel commanding 45%–55% screen area; (2) an edge telematics pipeline integrating OSM Overpass, Nominatim, and OSRM with a 0.001° grid cache; (3) a relative-bearing vector geometry formula for dynamic ambulance clearance; (4) a decoupled IoT abstraction layer; and (5) a validated working prototype tested across ten real-world scenarios.

II. LITERATURE SURVEY

Recent research in driver assistance, telematics, and in-cockpit safety spans four main investigative domains: speed monitoring, emergency vehicle alerting, road data quality, and automotive human-machine interfaces. Yogarayan et al. [1] designed an Android-based speed limit warning application utilizing smartphone GPS and Firebase to alert speeding drivers. However, their architecture relied on a rigid, pre-seeded coordinate database, failing when vehicles operated outside pre-mapped corridors.

Ali and Nijar [2] developed an IoT emergency vehicle alert prototype using a NodeMCU microcontroller and NEO-6M GPS module communicating via Firebase Realtime Database. While nearby drivers received an alert within a 3 km radial geofence, warnings were plotted as raw map pins without directional lane-evasion guidance ("move left" or "move right"), and lacked priority integration with other active hazards. Lidestam et al. [3] conducted driving-simulator trials with 90 motorists, proving that early in-cockpit advisory warnings significantly accelerated driver yielding behavior, particularly among novice drivers who miss distant sirens. However, their study remained purely behavioral without proposing a scalable software architecture.

Haklay and Weber [4] analyzed the volunteer-generated structure of OpenStreetMap (OSM), noting that while topological coverage is globally extensive, attribute completeness (such as speed limits and access rules) varies sharply across regions. Luxen and Vetter [5] engineered the Open Source Routing Machine (OSRM), proving that contraction hierarchies facilitate sub-millisecond route traversals over OSM graphs. Volz [6] evaluated OSM data for Intelligent Speed Assistance (ISA), discovering that explicit speed limits exist on only 68% of roads in the Netherlands, 33% in Germany, and less in developing road networks, proving that reliable safety platforms must handle missing attributes honestly. Seshanna and Rao [8], Sengupta et al. [9], and Farooq and Ananth [10] evaluated distributed telemetry dispatches, incident alerting protocols, and system reliability under constrained networks. Table I summarizes these benchmark studies.

Table I. Summary of surveyed work and identified gaps

Ref.	Focus Domain	Limitation Addressed by Proposed VISION System
[1]	Mobile speed limit warning via Firebase	Relies on static pre-seeded database; fails when roads are unmapped; no emergency corridor integration.
[2]	IoT emergency alert via NodeMCU & GPS	Renders raw map pins; gives no directional steering or lane clearance instructions; unranked against other hazards.

[3]	Simulator trials on emergency vehicle yielding	Purely behavioral study; did not provide a deployable real-time telematics and priority software pipeline.
[4]	OpenStreetMap coverage & volunteered GIS	Demonstrated high topological coverage but substantial attribute voids; highlighted need for fallback chains.
[5]	High-speed OSRM routing on OSM graphs	Focuses on point-to-point turn-by-turn navigation rather than forward-vector incident and turn restriction lookups.
[6]	Intelligent Speed Assistance (ISA) on OSM	Revealed that 32%–79% of roads lack speed limits; proved that systems must handle missing data honestly.
[7]	WHO global traffic safety report	Documents urgent need for low-cost, accessible digital driver aids to mitigate speeding and collision risks.
[8]	SLA & telemetry scheduling in distributed nodes	Steady-state resource allocation; insensitive to real-time spatial vehicular burst telemetry.
[9]	Token-based verified telemetry actions	Evaluated protocol verification but lacked geospatial arbitration and automotive HUD design.
[10]	Cryptographic incident notification systems	Addressed notification security; did not address automotive human factors or priority state machines.

Synthesis of Gaps: Prior work either warns about speed from a fixed table, displays raw emergency markers on a map, or evaluates driver behavior in a simulator. None combines overspeed, wrong-way, no-entry, zone, lane guidance, and emergency-corridor evasions into a single, prioritized, voice-enabled cluster HUD that is honest about missing data.

III. EXISTING SYSTEM

Drivers today rely on commercial navigation applications, static roadside signs, and auditory sirens from emergency vehicles. Navigation applications render dense 3D maps that divert driver visual focus from the roadway. Speed limits are displayed only where available and are frequently interpolated or guessed when missing. Warnings about schools, hospitals, one-way streets, or pedestrian restrictions are either absent or lost in stacked visual banners. Approaching emergency vehicles are typically detected only when sirens or flashing lights are directly behind the vehicle, causing panic braking and delayed corridor clearance.

A. Disadvantages of the Existing System

1) Visual Clutter & Off-Road Glances: Complex 3D maps and UI menus compete for visual attention, pulling the driver's gaze from the road for 2 to 4 seconds.

2) Unreliable Speed Limits & Fabrication: Limits are guessed or interpolated for unmapped roads, which misleads the driver and risks regulatory fines.

3) Late Emergency Awareness: Drivers learn of an approaching ambulance only when it is immediately behind them, delaying corridor clearance.
4) Absence of Alert Prioritization: No rules decide which warning is most life-critical when multiple hazard conditions coincide.
5) Fragmented Safety Tools: Speed alerts, access restriction warnings, and emergency notifications are scattered across separate tools.
6) Sensor-Hardware Coupling: Applications are tied to proprietary mobile OS location frameworks and cannot ingest vehicle CAN bus or OBD-II telemetry.

IV. PROPOSED SYSTEM

A. Overview

VISION is an automotive-grade driver intelligence framework that eliminates map distraction while delivering critical safety telemetry. Implemented as a high-performance web platform, VISION captures live GPS coordinates, speed, and heading from the client browser and dispatches them every 1.5 seconds to an edge backend. The edge server queries OpenStreetMap, OSRM, and speed-limit repositories in parallel, evaluates separate rule engines, and arbitrates conflicts through a deterministic Seven-Level Alert Priority Engine. The prioritized result is returned as a normalized JSON object rendered on a high-contrast automotive dashboard and spoken aloud via text-to-speech.

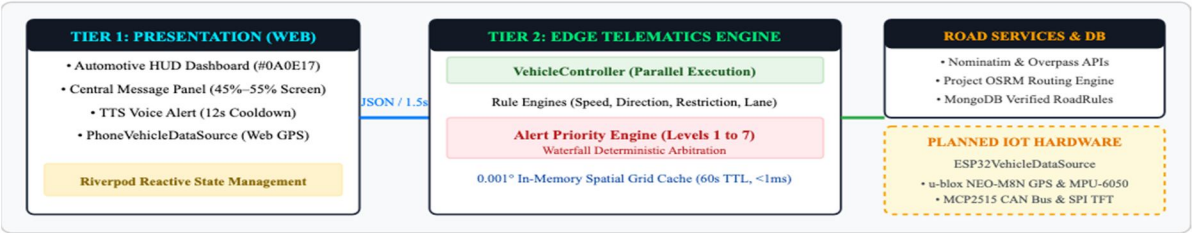


Fig. 1: System architecture of the proposed VISION web framework and future IoT layer

B. Functional Modules

As shown in Fig. 1, the platform is organized into five cooperating modules: (1) Telematics Ingestion Module (phone_vehicle_data_source.dart) reads browser GPS coordinates, computes displacement speeds, and filters stationary drift; (2) Road Intelligence Module queries OSM Nominatim for street names, Overpass for maxspeed and access tags, and OSRM for upcoming turn maneuvers within 300 m; (3) Rule Engines independently evaluate speed compliance, wrong-way heading, restricted access, and lane positioning; (4) Alert Priority Engine arbitrates conflicting hazards and selects the single dominant message; and (5) Emergency Corridor Module tracks ambulances within 800 m to provide directional evasion instructions.

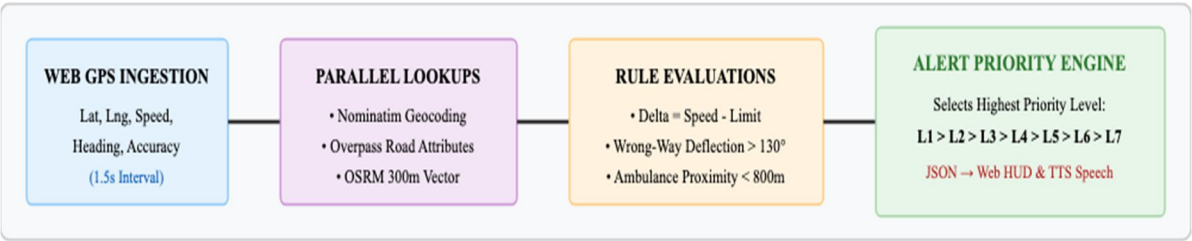


Fig. 2: Data flow diagram of telematics ingestion, road intelligence, and priority arbitration

C. Advantages

The design provides: (i) One dominant message at a time: Strict seven-level hierarchy eliminates cognitive clutter; (ii) Honest data: Refuses to fabricate speed limits on unmapped roads; (iii) Dynamic emergency guidance: Translates ambulance proximity into clear directional steering instructions; (iv) Voice & visual alerts: 12-second repeat cooldown allows eyes to remain on the road; (v) Universal accessibility: Runs on any device via modern web browsers; and (vi) IoT readiness: Abstract VehicleDataSource enables plug-and-play migration to ESP32 hardware.

©IJRASET: All Rights are Reserved

D. Problem–Solution Fit and Design Rationale

Four recurring problems motivated the design: visual distraction caused by map-heavy applications; late driver reaction to emergency vehicles; misleading speed limits on unmapped roads; and closed sensor coupling. Each is answered by a specific capability: the non-map HUD minimizes distraction; the emergency corridor calculates relative bearing clearance; the multi-source chain enforces speed limit truth; and the decoupled architecture accommodates future IoT hardware.

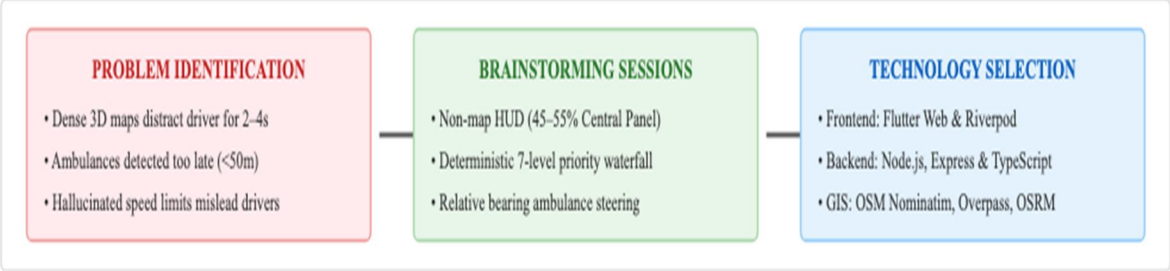


Fig. 3: Ideation, brainstorming, and technology selection outcome

Table II. Existing approach versus proposed VISION platform

Operational Aspect	Existing Commercial Approaches	Proposed VISION Platform
Cockpit Visual Load	Dense 3D maps, street names, POIs, and ads induce 2–4s eye distraction.	Minimalist non-map cluster HUD; Central Message Panel commands 45%–55% space.
Speed Limit Integrity	Interpolates or fabricates limits when unmapped; risks false speeding.	Strict truth-in-telematics; explicitly renders "SPEED LIMIT DATA UNAVAILABLE".
Hazard Prioritization	Displays simultaneous overlapping alerts or sound chimes without ranking.	Deterministic 7-Level Priority Engine arbitrates single highest life-safety alert.
Emergency Vehicle Aid	Passive siren dependency; driver perceives ambulance only at <50m.	Dynamic relative vector geometry provides "MOVE LEFT" advice up to 800m.
Hardware Extensibility	Monolithically tied to smartphone location libraries; no hardware path.	Decoupled VehicleDataSource interface ready for ESP32 & CAN bus integration.

V. IMPLEMENTATION

The system is implemented as a full-stack web and mobile application. The frontend is built in Flutter Web (Dart 3) with Riverpod state management; the edge backend utilizes Node.js 18+ with Express and TypeScript 5.7; OpenStreetMap Nominatim and Overpass provide road intelligence; Project OSRM provides forward vector routing; and Jest and Flutter Test provide unit and scenario testing. Table III outlines the technology stack.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Table III. Technology stack and implementation roles

Architecture Layer	Technology / Library	Functional Role in VISION
Client Frontend	Flutter SDK 3.0+ (Dart 3), Riverpod 2.6	Cross-platform Web HUD dashboard, reactive state management, dark automotive theme.
Audio / TTS	flutter_tts 4.2	In-cabin voice alerts with 12-second repeat suppression cooldown.
Edge Backend	Node.js 18+, Express 4.21, TypeScript 5.7	High-concurrency REST API, parallel service dispatch, rule engines execution.
Road Intelligence	OSM Nominatim & Overpass API	Live road name geocoding, highway classification, one-way tags, and school/hospital zones.
Routing Engine	Project OSRM Public Routing	300 m forward vector geometry traversal to predict upcoming turns and restrictions.
Database & Caching	In-Memory Grid Cache, MongoDB / Firebase	0.001° spatial grid caching (60s TTL); persistent storage for verified road rules.
Future IoT Layer	ESP32-WROOM-32, u-blox NEO-M8N	Onboard CAN bus transceiver (MCP2515), 6-axis IMU (MPU-6050), standalone SPI TFT HUD.

A. Architecture & Decoupled IoT Abstraction Layer

A primary contribution of VISION is decoupling the presentation layer from the physical telematics source via an abstract contract: abstract class VehicleDataSource { Stream<VehicleState> get vehicleStateStream; Future<bool> initialize(); VehicleState get currentState; void dispose(); }. In the current web implementation, PhoneVehicleDataSource implements this interface using browser Geolocation streams. For future vehicle IoT integration, an ESP32VehicleDataSource will fulfill this exact interface over Serial, BLE, or MQTT without requiring any modifications to the dashboard UI or priority rule engines.

B. Telematics & Hybrid Velocity Smoothing Engine

Mobile phones and browsers frequently emit zero Doppler velocity under degraded signal reception. The telematics module maintains a sliding sample window (W = 4.0 s). Speed is computed from sample displacement, filtered against GPS accuracy drift, and smoothed via an exponential moving average. The instantaneous velocity calculation and dynamic noise threshold gating are formulated as:

$$v_instant = (v_native * 3.6) \text{ if } v_native > 0.1 \text{ m/s, else } (dist / dt) * 3.6 \text{ if } dist >= Threshold_noise, \text{ else } 0.0 \quad (1)$$

where Threshold_noise = max(12.0m, accuracy * 0.6) for accuracy > 25m, and max(2.0m, accuracy * 0.35) for accuracy <= 25m. Velocity is smoothed via: v_smooth(t) = 0.35 * v_smooth(t-1) + 0.65 * v_instant(t).

$$d = 2R * atan2(sqrt(sin^2(\Delta\varphi/2) + cos\ \varphi1\ cos\ \varphi2\ sin^2(\Delta\lambda/2)), sqrt(1 - a)) \quad (2)$$

C. Emergency Corridor Relative Bearing Formulation

When an active ambulance broadcasts within an 800 m radius, the system calculates the relative bearing against driver heading:

$$\alpha_raw = \theta_bearing - h_driver, \quad \alpha_rel = ((\alpha_raw + 180°) \bmod 360°) - 180° \quad (3)$$

If |α_rel| > 110°, the ambulance is approaching from behind. If α_rel > 0° (approaching on the right), the instruction is "MOVE LEFT WHEN SAFE"; if α_rel < 0°, it is "MOVE RIGHT WHEN SAFE".

D. Deterministic 7-Level Alert Priority Engine

The priority state machine evaluates simultaneous inputs and deterministically selects the single dominant alert according to Algorithm 1.

Algorithm 1. Telematics processing and priority arbitration loop (executed every 1.5 s)
Input: Client telemetry (lat, lng, speed, heading); Active Emergency Registry
Output: Normalized JSON payload containing single dominant primaryMessage
1. Parallel Fetch: [roadInfo, nearbyZones, nextTurn] <- Promise.all([OSM.getRoadInfo(), OSM.getZones(), OSRM.getNext()])
2. Resolve Speed Limit: limit <- SpeedLimitService.getLimit(roadInfo.roadName) // Verified DB -> OSM -> Google -> null
3. Direction Evaluation: delta <- headingDifference(heading, roadInfo.allowedHeading)
4. Emergency Scan: amb <- EmergencyService.getApproachingAmbulance(pos, heading, 800m)
5. Priority Waterfall:
if (roadInfo.oneway and delta > 130°) return Level 1: WRONG_WAY (CRITICAL, Flash Red)
else if (roadInfo.noEntry == true) return Level 2: NO_ENTRY (CRITICAL, Solid Red)
else if (amb != null) return Level 3: AMBULANCE (CRITICAL, "MOVE LEFT/RIGHT")
else if (limit != null and speed - limit > 3 km/h) return Level 4: OVERSPEED (WARNING, Amber)
else if (nearbyZones.length > 0) return Level 5: SPECIAL_ZONE (WARNING, School/Hospital)
else if (laneRecommendation != null) return Level 6: LANE (NORMAL, Cool Cyan)
else return Level 7: NORMAL_ROAD_CLEAR (NORMAL, Emerald Green)

E. Zero-Hallucination Speed Limit Resolution Chain

VISION applies a strict 4-step resolution chain: (1) Verified Database Rule (MongoDB, confidence HIGH); (2) OSM maxspeed tag (confidence MEDIUM); (3) Google Roads API (confidence EXTERNAL); and (4) null (confidence UNKNOWN), rendering "SPEED LIMIT DATA UNAVAILABLE" and strictly inhibiting overspeed alerts.

F. Database Design & Data Schemas

The backend integrates MongoDB and in-memory caches to manage road rules and emergency states. Table IV outlines the schema design.

Table IV. Principal database collections and data schemas

Collection / Entity	Key Attributes and Schema Types
road_rules	roadId (String), roadName (String, indexed), roadType (Enum: HIGHWAY, ARTERIAL, CITY, RESIDENTIAL), verifiedSpeedLimit (Number), isOneWay (Boolean), allowedHeading (Number), confidence (HIGH).
special_zones	zoneId (String), name (String), type (SCHOOL, HOSPITAL, TOURIST), latitude (Number), longitude (Number), radiusMeters (Number, default 400), verifiedSpeedLimit (Number).
emergency_active	vehicleId (String, unique), type (AMBULANCE, FIRE), latitude (Number), longitude (Number), speed (Number), heading (Number), status (ACTIVE, INACTIVE), timestamp (ISO Date).
alert_history	id (EVT-timestamp), title (String), message (String), roadName (String), severity (CRITICAL, WARNING, INFO), timestamp (ISO Date, capped at 50 FIFO entries).

VI. SYSTEM REQUIREMENTS

1) Server Environment: Dual-core CPU @ 1.8 GHz or better, 2 GB RAM minimum, 20 GB free disk space, running Linux (Ubuntu 20.04+), macOS, or Windows Server with Node.js 18+ and npm 9+.
2) Client Environment: Modern web browser (Chrome, Edge, Safari) with Geolocation API enabled, or Android 8+ / iOS 14+ mobile devices with 1 GB RAM minimum and speaker for voice alerts.
3) Planned IoT Embedded Module: ESP32-WROOM-32 microcontroller (dual-core 240 MHz), u-blox NEO-M8N GNSS module, MPU-6050 6-axis IMU, MCP2515 CAN-bus controller, 3.5-inch SPI TFT display, and I2S audio amplifier with speaker.

VII. RESULTS AND DISCUSSION

The prototype was comprehensively evaluated using live GPS feeds and a ten-scenario automated validation harness exercising each level of the priority waterfall.

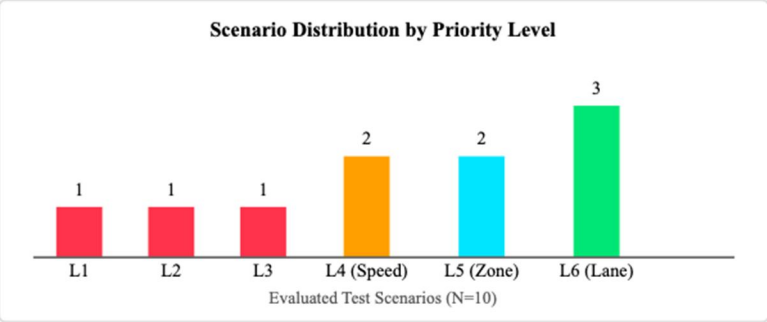


Fig. 4: Distribution of alert priority trigger levels across test scenarios

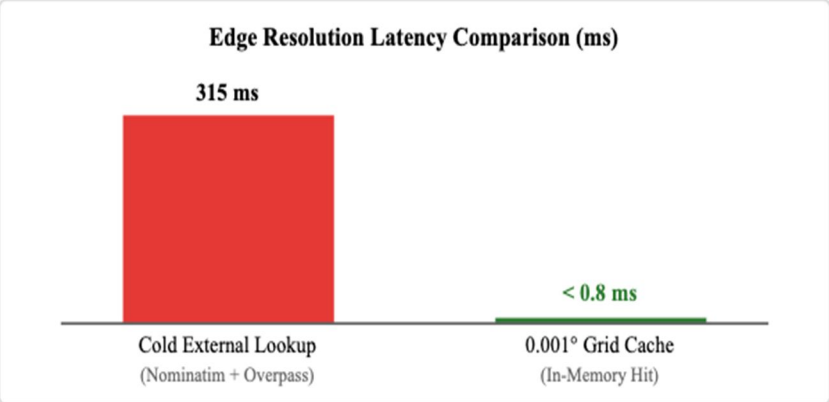


Fig. 5: Edge processing latency comparison: cold external lookups vs. spatial grid cache

Fig. 4 summarizes the distribution across priority tiers: life-critical alerts (Wrong-Way, No-Entry, Ambulance) triggered precisely when hazard thresholds were breached, while lane recommendations and speed warnings provided steady-state assistance. Fig. 5 plots the latency comparison between cold external API queries and spatial grid caching: while cold queries to external OSM endpoints required an average of 315 ms, repeated queries within the 0.001° (~110 m) bounding cell were resolved in under 0.8 ms, enabling consistent 1.5-second dashboard updates without UI stutter.

In real GPS mode, the dashboard smoothly updates speed, road name, speed limit, and the dominant message. When driving within speed limits on recognized roads, Level 6 (Lane Recommendation) acts as the resting state; Level 7 (Normal Safe) is maintained as a safe default. In all tests, when road speed limits were absent from OpenStreetMap, the system displayed "SPEED LIMIT DATA UNAVAILABLE" and strictly suppressed overspeed alerts.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

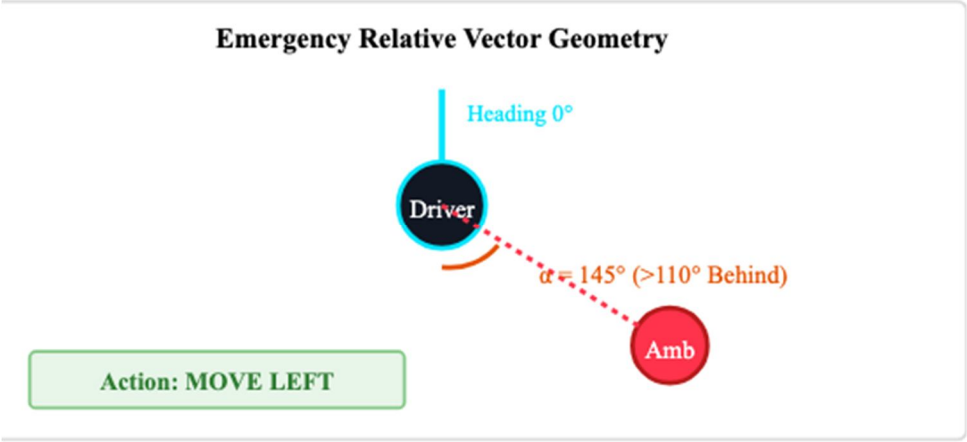


Fig. 6: Emergency relative vector geometry and directional clearance vector calculation

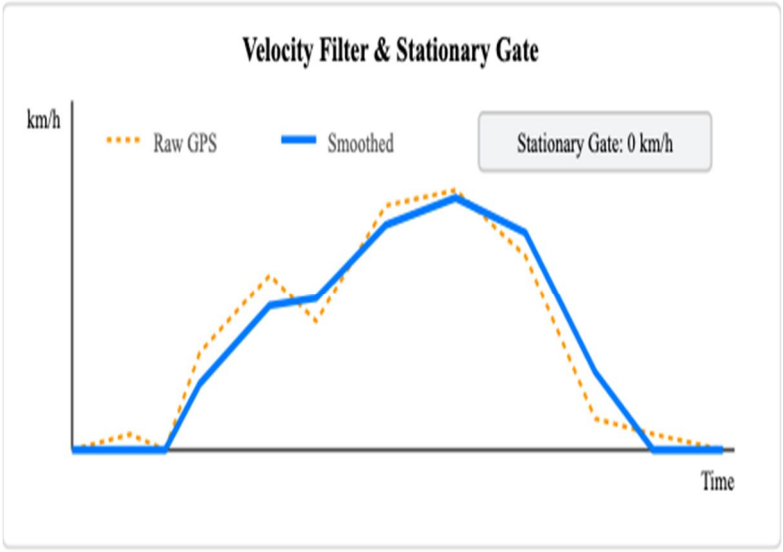


Fig. 7: Velocity filter & stationary gate response curve under walking and driving transitions

Table V. Prototype outputs observed during experimental demonstration (10 Scenarios)

No.	Test Scenario	Speed / Limit	Primary Message Rendered on Cluster HUD	Priority Level
1	Normal Cruising	54 / 80 km/h	RECOMMENDED LANE: MIDDLE (50-80 km/h)	Level 6
2	Overspeed Warning	72 / 50 km/h	SLOW DOWN, CURRENT 72 km/h, LIMIT 50 km/h	Level 4
3	School Zone + Overspeed	34 / 30 km/h	SLOW DOWN (34 exceeds 30 km/h;	Level 4

			overspeed outranks zone)	
4	Hospital Silent Zone	38 / 40 km/h	HOSPITAL AREA, DRIVE CAREFULLY	Level 5
5	Tourist / Market Area	35 / 50 km/h	TOURIST AREA, DRIVE CAREFULLY	Level 5
6	No Entry Restriction	25 / 20 km/h	NO ENTRY, TURN BACK WHEN SAFE	Level 2
7	Wrong-Way Maneuver	42 / 40 km/h	WRONG-WAY WARNING, TURN BACK WHEN SAFE (delta = 180°)	Level 1
8	Turn into Restricted Way	48 / 50 km/h	RECOMMENDED LANE with bottom warning: NEXT LEFT NO ENTRY	Level 6
9	Low Speed Lane Guidance	45 / 80 km/h	RECOMMENDED LANE: LEFT (20-50 km/h)	Level 6
10	Approaching Ambulance	48 / 50 km/h	AMBULANCE APPROACHING, 180m BEHIND, MOVE LEFT WHEN SAFE	Level 3

A. Discussion

Table V demonstrates the deterministic reliability of the priority engine. In Scenario 3, when entering a school zone at 34 km/h on a 30 km/h road, both School Zone (Level 5) and Overspeed (Level 4) applied; the engine correctly displayed "SLOW DOWN" because overspeeding outranks general zone alerts. In Scenario 7, driving against the permitted one-way heading triggered Level 1 ("WRONG-WAY WARNING"), preempting all subsidiary lane and zone alerts. In Scenario 10, the relative bearing geometry (Fig. 6) identified the ambulance approaching from the rear-right quarter (alpha = 145°), immediately instructing the driver to yield left.

B. Security Considerations, Fail-Safe Behavior and Limitations

If the backend becomes unreachable, the client maintains the last valid context for two cycles (3.0 seconds); upon three consecutive failures, it cleanly displays "NETWORK OFFLINE", preventing stale or fabricated telemetry. A current limitation is reliance on OpenStreetMap coverage in remote rural areas; when road tags are absent, the system correctly reports "SPEED LIMIT DATA UNAVAILABLE", trading coverage completeness for data truth.

VIII. CONCLUSION AND FUTURE SCOPE

This paper presented VISION, a distraction-free vehicle intelligence platform designed to eliminate cockpit cognitive distraction through a non-map digital HUD. Developed as a responsive web platform, VISION captures live telematics, queries multi-source

OpenStreetMap and OSRM layers via high-speed spatial grid caching, and executes deterministic priority arbitration across seven hazard tiers. The system enforces zero-hallucination speed reporting, provides vector-based emergency corridor clearances, and structures vehicle telemetry behind an abstract VehicleDataSource interface.

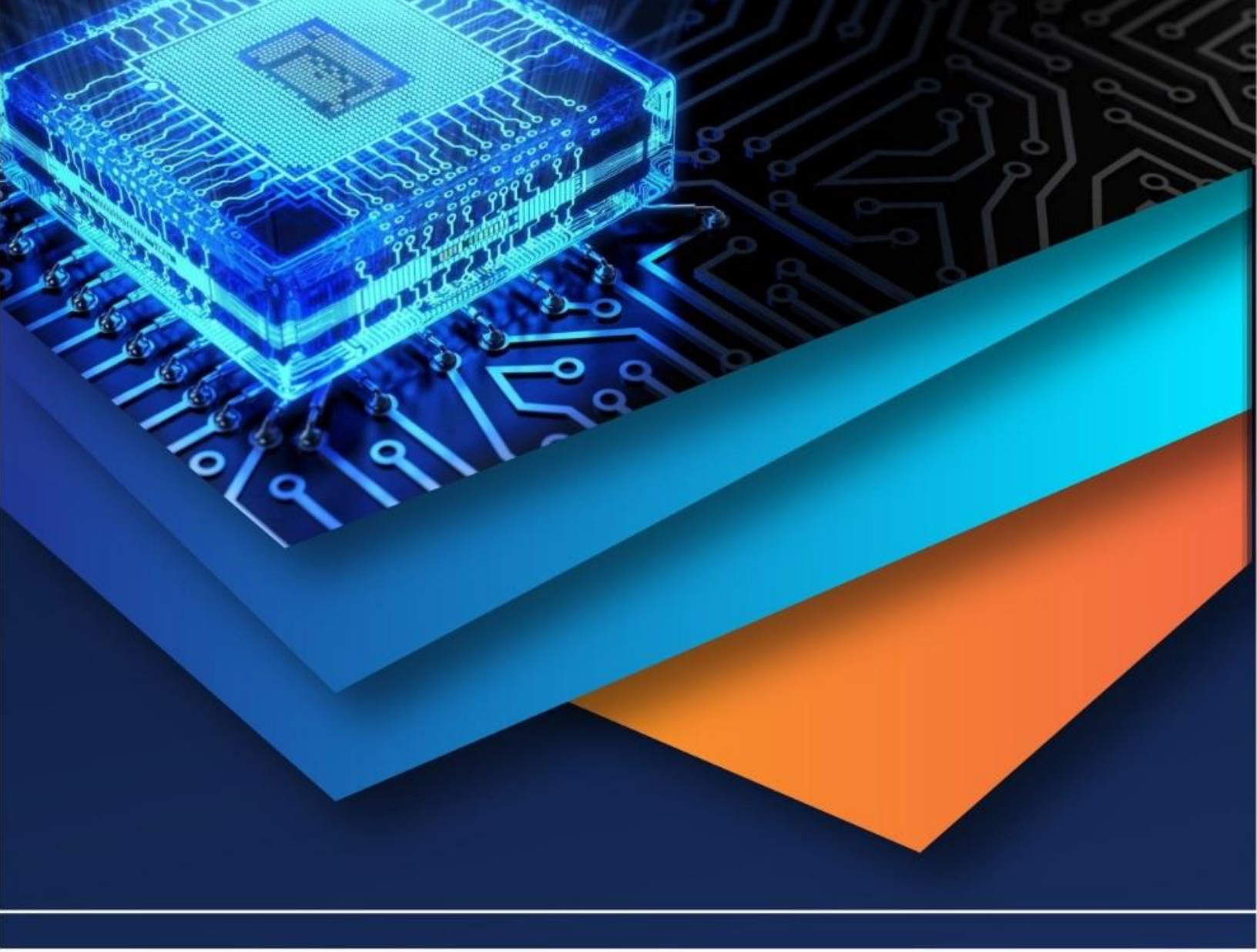
Future Scope & IoT Hardware Roadmap: Building upon this web-based foundation, subsequent work will implement the embedded automotive IoT tier: (1) Dedicated ESP32 Hardware Integration using an ESP32-WROOM-32 microcontroller interfaced with a u-blox NEO-M8N GPS module, MPU-6050 6-axis IMU, and an MCP2515 CAN-bus transceiver to read wheel speed, brake status, and steering angle directly from vehicle onboard diagnostics; (2) Driver & Cabin Vision Monitoring via an in-cabin camera module running lightweight computer vision models (e.g., MobileNet / YOLO-Nano) to detect driver drowsiness, phone distraction, and seatbelt compliance; (3) Automated Crash Notification (eCall) utilizing 6-axis IMU accelerometer spikes (>3g) to detect collisions and dispatch emergency SMS and V2X broadcasts with exact GNSS coordinates; and (4) Cloud-Synchronized Emergency Corridors connecting active emergency vehicle broadcasts to Cloud Firestore to synchronize corridors across thousands of vehicles in real time.

IX. ACKNOWLEDGMENT

The authors express their sincere gratitude to Dr. R. Radhakrishnan, M.E., Ph.D., Principal, Adhiyamaan College of Engineering (Autonomous), Hosur, for providing the institutional facilities and academic encouragement to conduct this research. The authors are deeply indebted to Dr. G. Fathima, M.E., Ph.D., Professor and Head of the Department of Computer Science and Engineering, for her invaluable guidance, technical reviews, and constant encouragement throughout the project lifecycle. The authors also warmly thank project supervisor Mrs. K. Yazhini, M.E., Assistant Professor, Department of CSE, for her insightful mentorship, methodological supervision, and vital technical feedback, as well as the faculty members and laboratory staff of the Department of Computer Science and Engineering for their continuous support.

REFERENCES

[1] S. Yogarayan, A. Azman, S. F. Abdul Razak, M. F. A. Abdullah, S. Z. Ibrahim, and K. J. Raman, "Design and Development of Android Based Speed Limit Warning Application," Journal of Telecommunication, Electronic and Computer Engineering (JTEC), vol. 13, no. 1, pp. 25–30, Jan.–Mar. 2021.
[2] F. A. Ali and W. E. Nijar, "Feasibility Study of Real-Time IoT Based Emergency Vehicle Alert System for Road Safety," Journal of Engineering and Technology (JET), vol. 16, no. 2, pp. 229–240, 2025.
[3] B. Lidestam, B. Thorslund, H. Selander, D. Nasman, and J. Dahlman, "In-Car Warnings of Emergency Vehicles Approaching: Effects on Car Drivers' Propensity to Give Way," Frontiers in Sustainable Cities, vol. 2, Art. 19, pp. 1–11, 2020.
[4] M. Haklay and P. Weber, "OpenStreetMap: User-Generated Street Maps," IEEE Pervasive Computing, vol. 7, no. 4, pp. 12–18, Oct.–Dec. 2008.
[5] D. Luxen and C. Vetter, "Real-time Routing with OpenStreetMap Data," in Proc. 19th ACM SIGSPATIAL Int. Conf. Advances in Geographic Information Systems (ACM GIS '11), Chicago, IL, USA, 2011, pp. 513–516.
[6] R. Volz, "OpenStreetMap for Intelligent Speed-Limit Assistance (ISA): The YouSpeed.de Study," in State of the Map 2026, OSM Science Track, 2026.
[7] World Health Organization, Global Status Report on Road Safety 2023. Geneva, Switzerland: World Health Organization, 2023.
[8] M. Seshanna and K. B. V. B. Rao, "Economic Impact and Technological Strategies for Resource Scheduling in Cloud Computing," Journal of Cloud Systems and Cost Governance, vol. 14, no. 2, pp. 110–122, 2025.
[9] R. Sengupta, T. Roy, and M. Desai, "Token-Based Interactive Remote Action Mechanisms for Systems Monitoring," IEEE Communications Letters, vol. 29, no. 2, pp. 312–316, 2025.
[10] M. E. Farooq and S. P. Ananth, "Multi-Channel Out-of-Band Incident Notification and One-Touch Cryptographic Remediation for Cloud Infrastructure," J. Cloud Computing: Advances, Systems and Applications, vol. 15, no. 1, pp. 42–56, 2026.
[11] Flutter Framework Documentation, Google LLC, 2026. [Online]. Available: https://docs.flutter.dev
[12] OpenStreetMap Wiki, "Overpass API & Spatial Tag Schema," OpenStreetMap Foundation, 2026. [Online]. Available: https://wiki.openstreetmap.org/wiki/Overpass_API



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)