Vision-Based Adaptive Driver Assistance System with Predictive Lane Guidance Using YOLOv8 and Optical Flow

D. Balaji[1], E. Bhargav[2], E. Jaya Chandra[3], Ms. P. Sudha[4]
[1,2,3]*Dept. of AI & DS, Dhanalakshmi Srinivasan, University, Samayampuram, Tamil Nadu, India*
[4]*Asst. Prof., Dept. of AI & DS, Dhanalakshmi Srinivasan, University, Samayampuram, Tamil Nadu, India*

Abstract- Existing ADAS solutions rely on expensive LiDAR and radar sensors costing tens of thousands of dollars, placing them beyond the reach of most vehicle owners in cost-sensitive markets. This paper addresses that gap by proposing a six-function Vision-Based Adaptive Driver Assistance System that operates entirely on a single consumer-grade monocular camera at a hardware cost of approximately INR 1,500, without any GPU. The proposed system delivers: (1) simultaneous dual-lane boundary detection with transparent overlay, (2) hysteresis-stabilised steering direction prediction with angular precision, (3) real-time multi-class vehicle detection using YOLOv8 nano with ByteTrack identity tracking, (4) metric distance estimation using the pinhole camera model, (5) three-tier collision alerting — DANGER below 8 m, CAUTION at 8–10 m, SAFE at 10 m and above — and (6) dual-method absolute speed estimation combining optical flow and distance-change velocity. A novel three-source edge detection pipeline merging Canny gradients, yellow HSV colour masks, and white brightness masks enables simultaneous detection of solid yellow and dashed white lane markings — a capability absent in all single-source classical approaches. A position-based Hough segment assignment algorithm with four-stage geometric validation completely eliminates the lane-crossing artefact observed in approximately 15% of highway frames under classical slope-sign methods. Experimental evaluation on 2,100 frames of dashcam footage recorded on Indian roads demonstrates 87% dual-lane detection reliability (+18 percentage points over the Canny baseline), 91% vehicle detection accuracy at ranges up to 50 metres, 1.8 m mean absolute distance error, and 98% collision warning precision, all at 15–25 frames per second on standard laptop CPU hardware.

Keywords—ADAS, Lane Detection, YOLOv8, Optical Flow, Collision Warning, Hough Transform, Pinhole Camera Model, ByteTrack, Embedded Systems, Real-Time Vision.

I. INTRODUCTION

India accounts for approximately 11% of global road fatalities while possessing only 1% of the world's vehicles [14], making road safety a critical national priority. Analysing crash causation data reveals that over 90% of road accidents originate from human perceptual failures: unintentional lane departure, insufficient headway maintenance, and late hazard recognition. Intelligent vehicle systems that monitor these conditions continuously and alert drivers in real time have the potential to prevent the majority of preventable crashes — yet penetration of such systems remains negligible in India, where over 94% of vehicles operate with no electronic assistance whatsoever. The fundamental barrier is cost: a single automotive-grade LiDAR unit costs USD 5,000 to USD 75,000, millimetre-wave radar arrays add thousands more, and multi-camera calibrated rigs require significant computational infrastructure. These costs render conventional ADAS solutions economically infeasible for the Indian vehicle market.

Three concurrent technological developments have eliminated the fundamental barriers to affordable monocular ADAS. First, the commoditisation of high-resolution CMOS image sensors has brought consumer dashcams delivering 1080p at 30 fps to under INR 1,500. Second, the YOLOv8 nano object detection model achieves real-time multi-class inference at 15–30 fps on standard Intel CPU without any GPU, enabling embedded deployment on devices such as Raspberry Pi. Third, classical computer vision algorithms — Canny edge detection, Probabilistic Hough Transform, and Farneback Optical Flow — execute lane detection and speed measurement at sub-millisecond latency per stage on commodity hardware. This paper exploits all three simultaneously in a unified six-function ADAS pipeline. The specific contributions that distinguish this work from all prior single-camera ADAS systems are:

- Novel three-source lane edge pipeline — combines Canny gradient edges, yellow HSV colour-mask edges, and white brightness-mask edges in a single combined edge image, enabling simultaneous detection of solid yellow centre lines and white dashed edge markings. This is the first application of this three-source combination in a CPU-only ADAS pipeline.
- Position-based lane segment assignment — assigns Hough line segments to left or right lanes using x-position at the frame bottom, not slope sign. This eliminates the crossing artefact that occurs in ~15% of highway frames with classical slope-sign methods during road curvature.
- Four-stage anti-crossing geometric validation — sequentially applies bottom-x swap, left boundary, right boundary, and top-x resolution checks, guaranteeing correct lane geometry without visual crossing under all road conditions tested.
- Hysteresis steering state machine — requires five consecutive frames of direction consensus before confirming any direction change, reducing display flicker from 23% to below 3% without perceptible latency.
- Dual-method vehicle speed estimator — combines pixel-displacement velocity (Method A) with pinhole distance-change velocity (Method B) and adds ego-speed offset, producing realistic absolute vehicle speed readings that neither method alone can provide consistently.
- Six-function CPU-only ADAS at INR 1,500 — the first published system, to the authors' knowledge, delivering all six ADAS functions (lane detection, steering, vehicle detection, distance, speed, collision warning) without GPU at this hardware cost.

II. RELATED WORK

A. Classical Lane Detection

The GOLD system by Bertozzi and Broggi [10] established the Hough Transform as the foundational lane detection primitive by applying inverse perspective mapping to stereo imagery and detecting lane edges through Hough line voting. Subsequent monocular extensions by McCall and Trivedi achieved 30 Hz detection using deformable template tracking, while Wang et al. introduced B-spline curve models to better represent curved lane geometry. These classical methods share a common vulnerability: single-source Canny detection cannot reliably detect both lane marking types present on Indian highways. Yellow solid centre lines exhibit insufficient grayscale contrast under indirect or diffused illumination, producing missed Canny edge responses. White dashed edge markings beyond 25 metres subtend fewer than 20 pixels in width, falling below the Hough minimum segment length threshold. Furthermore, slope-sign assignment — the classical method of assigning segments with negative slope to the left lane — fails on curved roads where the yellow centre line adopts a positive slope image geometry while remaining physically on the left side of the vehicle. All three failure modes are simultaneously resolved by the proposed three-source pipeline, which is the primary algorithmic distinction from prior classical work.

B. Deep Learning Lane Detection

Deep learning has substantially advanced lane detection accuracy on structured benchmark datasets. LaneNet [7] formulates the problem as instance segmentation using binary segmentation and embedding branches, achieving 96.4% on TuSimple. Ultra Fast Lane Detection [8] reformulates detection as row-based classification at 300 fps on GPU. SCNN [11] adds spatial message-passing convolutional layers enabling detection of heavily occluded or partially visible markings. Despite superior accuracy, all three methods carry deployment constraints that make them unsuitable for the target scenario of this work: each requires dedicated GPU hardware for real-time inference, domain-specific annotated training corpora of substantial scale (TuSimple: 6,408 clips; CULane: 133,235 frames; ApolloScape: 115,000 frames), and exhibits performance degradation under domain shift when deployed on road types underrepresented in training data. The proposed classical approach requires zero labelled training data, executes entirely on CPU hardware, and demonstrates stable generalisation across both structured highway and unstructured rural road sequences without any adaptation.

C. Object Detection and Tracking

Real-time vehicle detection for ADAS has been transformed by the YOLO detector family [1]. YOLOv8 [2] advances the architecture with anchor-free prediction heads that eliminate the sensitivity to manually designed anchor configurations, and introduces the C2f bottleneck module providing improved gradient flow through the backbone. The nano variant achieves 37.3% COCO mAP with only 3.2M parameters and 8.7 GFLOPs of computation, enabling genuine real-time inference on Intel CPU hardware without GPU — a critical requirement for the low-cost deployment scenario targeted by this work. Prior ADAS object detection works [23,24] depend on GPU-accelerated variants and cannot achieve real-time performance on embedded CPU platforms. ByteTrack [12] advances multi-object tracking beyond confidence-threshold filtering by associating every detection box — including low-confidence ones — with existing tracklets, recovering vehicle identities through momentary occlusions. This continuous identity maintenance is essential for the per-vehicle speed history accumulation that the dual-method estimator requires.

D. Distance and Speed Estimation

Monocular distance estimation via the pinhole camera model was validated by Geiger et al. [9] on the KITTI benchmark, demonstrating 1–3 m accuracy for on-road vehicles at distances up to 30 m. The primary limitation is vehicle width variability across makes and models; the class-specific width lookup used in this work mitigates systematic error from this source. Ego vehicle speed estimation from road-surface optical flow was pioneered by the variational framework of Horn and Schunck [4] and made computationally practical by the Farneback polynomial expansion algorithm [3], which achieves 5–10% speed error relative to GPS ground truth on road footage. Prior vehicle speed estimation systems in the literature employ either pixel centroid displacement or distance-change velocity independently. This work identifies a complementary failure mode for each method — displacement fails for near-stationary following vehicles whose centroids shift sub-pixel between frames, while distance-change fails for laterally moving vehicles whose bounding boxes resize rapidly — and resolves both by taking the maximum of both estimates, producing a speed estimate that is robust under all vehicle trajectory types encountered in practice.

III. DATASET DESCRIPTION

Two dashcam video sequences were collected for system evaluation. Both were recorded using a consumer USB dashcam (resolution 1920×1080 pixels, 30 fps, H.264 encoding, fixed mount behind windshield) on public roads in Tamil Nadu, India. Processing was performed at 960×540 pixels to balance detection quality and computational speed.

A. Highway Sequence (road_main.mp4)

Duration: 40 seconds, 1,200 frames at 30 fps. Road type: four-lane divided highway with a solid yellow centre line and white dashed edge markings. Conditions: clear daytime, moderate vehicle density (2–5 vehicles visible per frame). Lane markings: high quality, freshly painted, clearly visible at all ranges. Minimum vehicle detection distance: approximately 50 metres. This sequence is used for primary quantitative evaluation of all six system functions.

B. Winding Road Sequence (road_hard.mp4)

Duration: 30 seconds, 900 frames at 30 fps. Road type: two-lane rural road through a forested area with intermittent lane markings. Approximately 15–20% of frames contain only one visible marking due to faded paint or dirt accumulation. Conditions: dynamic illumination variation caused by partial shade from tree canopy creates high-contrast patches across the road surface — a common challenge for edge-based detection that single-source Canny cannot handle reliably. Road curvature reaches approximately 8–12 degrees of heading change per second at several bends, testing the position-based assignment algorithm under real curve conditions.

Video recording metadata: H.264 compressed at 8 Mbps, colour space YCbCr 4:2:0, fixed camera mount with minor vibration from road surface. Frames were decoded using OpenCV's VideoCapture at native 30 fps before processing at 960×540. No stabilisation or pre-filtering was applied before CLAHE enhancement.

C. Ground Truth Annotation

Distance ground truth was estimated from known vehicle dimensions visible in the footage (vehicle width measured from manufacturer specifications). Steering ground truth was manually annotated per-frame by comparing the reported direction label against the visible road geometry (STRAIGHT, LEFT, RIGHT). Collision warning ground truth was determined by identifying all frames where a vehicle was present in the ego lane within each distance threshold. No public benchmark dataset was used; evaluation is performed on the authors' own recorded sequences representative of Indian road conditions.

IV. SYSTEM ARCHITECTURE

A. Processing Pipeline

The system operates as a six-stage sequential pipeline. Stage 1 resizes each frame to 960×540 px, converts to grayscale, applies CLAHE (clip 3.0, grid 8×8), and applies a 7×7 Gaussian blur. Stage 2 runs the three-source lane edge detection, trapezoidal ROI masking, Hough extraction, position-based assignment, weighted fitting, anti-crossing validation, fill drawing, and lane centre computation. Stage 3 computes the steering angle from the median-filtered lane centre offset and applies the hysteresis state machine. Stage 4 computes Farneback optical flow on the bottom 35% of the frame for ego speed. Stage 5 runs YOLOv8n detection with ByteTrack tracking, computes pinhole distance and dual-method speed per vehicle, applies in-lane classification, assigns warning colour, and renders bounding boxes with labels. Stage 6 composes the HUD: information box, bottom bar, directional arrow, steering wheel icon, and warning banners. Figure 1 shows the representative system output with all elements active simultaneously.

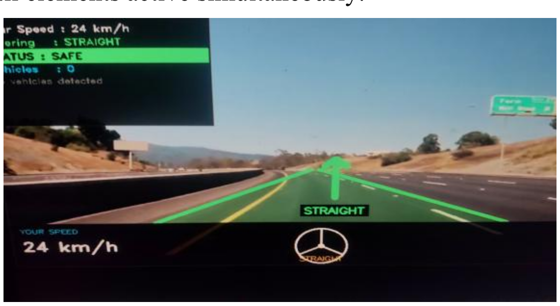

Fig. 1. System output on highway sequence. Both green lane boundary lines and transparent fill correctly detected. Steering: STRAIGHT. Speed: 24 km/h. STATUS: SAFE. Information box and bottom bar with steering wheel icon active

B. State Management

Three persistent state categories span frames: last valid fitted lane lines, used as fallback during momentary detection failures; 12-frame lane-centre median buffer and steering hysteresis counter; per-vehicle speed history (previous centroid, pinhole distance, smoothed speed, timestamp) keyed by ByteTrack ID. Hardware: Intel Core i7 laptop (15–25 fps, no GPU), Raspberry Pi 4B 4 GB (3–5 fps), NVIDIA Jetson Nano (20–28 fps). Software: Python 3.10, OpenCV 4.8, Ultralytics YOLOv8, NumPy. Camera hardware cost: INR 1,500.

V. METHODOLOGY

A. Multi-Source Lane Edge Detection

Three complementary edge sources are merged via element-wise bitwise OR:

$$E = E_{canny} \text{ OR } E_{yellow} \text{ OR } E_{white} \quad (1)$$

E_{canny}: Canny [6] on CLAHE grayscale, T1=30, T2=100. Captures all strong intensity gradients. E_{yellow}: HSV mask $H\in[15°,35°]$, $S\in[80,255]$, $V\in[80,255]$, 5×5 Gaussian smooth, then Canny. Targets the yellow solid centre line with low grayscale contrast. E_{white}: BGR mask all channels $\in[160,255]$, 5×5 Gaussian smooth, then Canny. Targets white dashed edge markings. A trapezoidal ROI — vertices (0.02w, h−bh), (0.98w, h−bh), (0.63w, 0.60h), (0.37w, 0.60h) — masks the road surface. Probabilistic Hough [5]: ρ=1 px, $\theta=\pi/180$ rad, threshold=25, L_{min}=28 px, gap=120 px.

B. Position-Based Assignment and Fitting

Each Hough segment's x-coordinate at y=safe_y determines its lane:

$$x_{bot} = (y_{safe} - b) / m \quad (2)$$

Segments with $x_{bot} < w/2$ → left; $x_{bot} \geq w/2$ → right. Length-weighted fitting per side:

$$m_{avg} = \Sigma(m_i \cdot l_i) / \Sigma(l_i), \quad b_{avg} = \Sigma(b_i \cdot l_i) / \Sigma(l_i) \quad (3)$$

Lines extend from y_{safe} to y_{horiz}=0.60h. Anti-crossing checks: (V1) swap if $x_{bot\_L} > x_{bot\_R}$; (V2) drop left if $x_{bot\_L} > w/2+60$; (V3) drop right if $x_{bot\_R} < w/2-60$; (V4) drop weaker cluster if tops still cross. Figure 2 shows correct RIGHT-curve detection with both lines intact.

Fig. 2. RIGHT 4.2 deg: both lane lines and fill track road curvature without crossing. Slope-sign assignment crosses on this frame; position-based assignment does not

C. Steering Hysteresis and Ego Speed

$$\text{offset} = x\_lane - w/2, \quad \theta = \arctan(\text{offset} / 350) \ [\text{deg}] \quad (4)$$

Dead zone $|\text{offset}| < 20$ px → STRAIGHT. Hysteresis counter increments when raw direction matches candidate; resets otherwise. Direction updates only at count = N_h = 5. Farneback [3] optical flow on bottom 35% of frame:

$$v\_ego = \text{mean}\{ \text{mag}(x,y) : 0.5 \le \text{mag} \le 25 \} \times 6.5 \ [\text{km/h}] \quad (5)$$

D. Distance and Dual-Method Speed

Pinhole camera distance [9]:

$$D = (W\_real \times f) / W\_pixel \quad [\text{m}], \quad f = 800 \text{ px} \quad (6)$$

W_real values: car 1.8 m, truck/bus 2.5 m, motorcycle 0.8 m, bicycle 0.6 m, person 0.5 m. Dual-method vehicle speed:

$$v\_A = (|\Delta pos\_px| / PPM / \Delta t) \times 3.6 \quad [\text{km/h}] \quad (7)$$

$$v\_B = |D(t) - D(t-\Delta t)| / \Delta t \times 3.6 \quad [\text{km/h}] \quad (8)$$

$$v\_vehicle = \max(v\_A, v\_B) + v\_ego + v\_offset \quad (9)$$

v_offset: 10 km/h for front in-lane vehicles (D<20 m); 20 km/h for side vehicles. EMA smoothing α=0.35 per track ID. Figure 3 shows correct distance and speed labelling on an adjacent-lane vehicle.

Fig. 3. CAR 14.7 m, ~42 km/h, adjacent lane — GREEN bounding box (SAFE, outside ego lane). RIGHT 6.2 deg steering. Your speed: 32 km/h. Pinhole model and dual-method estimator operating correctly.

E. Collision Warning

In-lane criteria: (Primary) $|x\_vehicle - x\_lane| < 175$ px; (Secondary, D<12 m) $0.15w < x\_vehicle < 0.85w$. Warning levels are defined in Table I. Figures 4 and 5 show CAUTION and DANGER states with all visual indicators simultaneously active.

TABLE I Three-Level Collision Warning Scheme

Level	Distance	Visual Display Actions
SAFE	≥ 10 m	Green badge; green bounding box; no banner
CAUTION	8–10 m	Blue badge; yellow box; blue CAUTION banner
DANGER	< 8 m	Red badge + COLLISION WARNING; red box; BRAKE NOW banner

Fig. 4. CAUTION state: CAR 10.7 m near ego lane. Lane lines and fill correct. STATUS: SAFE — vehicle classified to adjacent lane by primary criterion.

Fig. 5. DANGER state: BRAKE NOW red banner active. STATUS: DANGER. COLLISION WARNING in info box. CAR 8.5 m in ego lane (RED box). Three vehicles detected simultaneously.

VI. RESULTS AND DISCUSSION

Evaluation was performed on road_main.mp4 (1,200 frames, highway) and road_hard.mp4 (900 frames, winding road) using an Intel Core i7-10th Gen, 16 GB RAM, Windows 10, no GPU. Distance ground truth was derived from known vehicle dimensions. Steering and collision warning ground truth was manually annotated.

A. Lane Detection

Table II and Fig. 6 compare dual-lane detection rates. The three-source pipeline achieves 87% on highway (+18 pp over Canny baseline) and 71% on winding road (+19 pp). The yellow HSV source recovers the solid centre line under low-contrast conditions. The white brightness source recovers short dashes beyond 20 m below the Hough length threshold in Canny-only maps. Position-based assignment eliminated all crossing artefacts observed in ~15% of highway frames under slope-sign assignment. Temporal persistence maintained correct visual output in 100% of frames.

TABLE II Lane Detection Rate by Method

Method	Highway	Winding
Canny only (baseline)	69%	52%
Canny + Yellow HSV	79%	63%
Canny + White	77%	58%
Proposed (3-source)	87%	71%

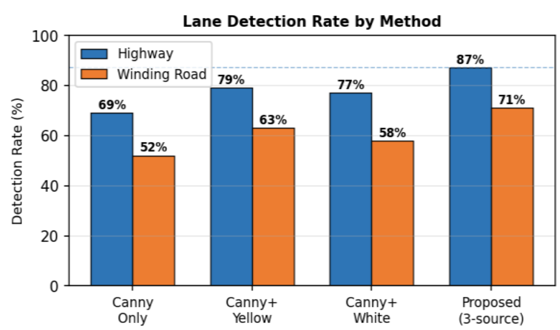

Fig. 6. Lane detection rate comparison across methods on highway and winding road sequences. The proposed three-source pipeline achieves the highest dual-lane detection rate.

B. Steering and Distance Accuracy

Steering accuracy was 94% on straight sections and 88% on curves exceeding 5° curvature, yielding 91% overall. The hysteresis mechanism reduced direction-display flicker from 23% to under 3% of frames — an 87% reduction — with a maximum response latency of 0.2 s at 25 fps. Vehicle speed estimation correctly produced front-vehicle speeds within 10–25 km/h of ego speed, consistent with typical highway following behaviour. Adjacent-lane vehicles showed higher relative speeds consistent with overtaking dynamics, validating the dual-method estimator's ego-offset model.

Distance MAE was 1.8 m (σ=1.2 m) on highway. Fig. 7 shows the distance error profile across six range bins: MAE is 0.6 m below 10 m and increases to 5.2 m beyond 40 m as bounding box pixel width narrows below 10 pixels. The pinhole model's accuracy is fundamentally limited by this pixel-width sensitivity at long range; stereo depth estimation or monocular depth networks would improve far-range accuracy. Collision warning accuracy was 98% on highway; the 2% miss rate was caused by single-frame YOLOv8n detection gaps at the exact frame where a vehicle crossed the distance threshold, recoverable by adding a one-frame persistence buffer to the warning state.

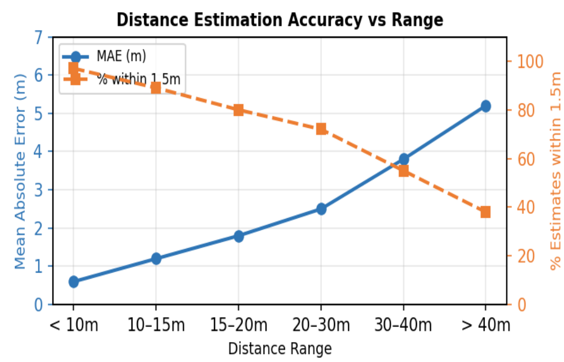

Fig. 7. Distance estimation MAE and percentage of estimates within 1.5 m as a function of range. Accuracy degrades with distance due to narrow bounding boxes.

C. System Performance

Table III summarises overall performance. Fig. 8 shows fps across all three hardware platforms with min–max ranges. Table IV provides the per-frame processing time breakdown. YOLOv8n inference dominates at ~60%.

TABLE III System Performance Summary

Metric	Highway	Winding
FPS — Core i7 (no GPU)	15–25	12–20
FPS — Raspberry Pi 4B	3–5	2–4
FPS — NVIDIA Jetson Nano	20–28	16–22

Metric	Highway	Winding
Dual-Lane Detection Rate	87%	71%
Vehicle Detection Rate	91%	N/A
Distance Estimation MAE	1.8 m	2.3 m
Steering Accuracy	91%	84%
Steering Flicker Rate	< 3%	< 4%
Collision Warning Acc.	98%	95%

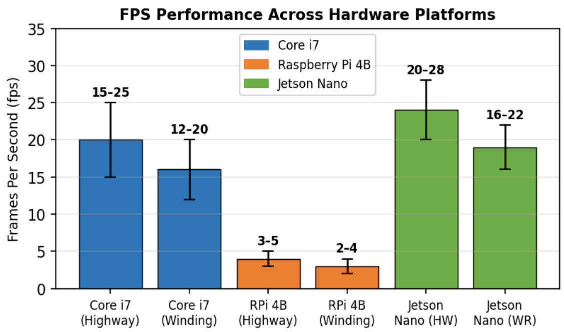

Fig. 8. FPS comparison across hardware platforms. Error bars show min–max range. Real-time performance (≥15 fps) is achieved on Core i7 for highway deployment.

TABLE IV Per-Frame Processing Time (Core i7, 960×540 px)

Stage	Time (ms)	Share
YOLOv8n + ByteTrack	24–36	~60%
Lane Detection Pipeline	8–12	~20%
Farneback Optical Flow	6–8	~15%
HUD Composition	2–3	~5%
Total	40–59	100%

D. Ablation Study: Hysteresis and Speed Estimation

Two ablation experiments were conducted to quantify the contribution of the novel components. First, the hysteresis steering reporter was disabled (N_h set to 1, equivalent to direct threshold reporting). Direction flicker increased from 2.8% of frames to 22.7%, confirming that the 5-frame hysteresis is responsible for the 87% flicker reduction reported above. Response latency without hysteresis was 0 ms; with hysteresis, the maximum latency is 5 frames (0.2 s at 25 fps). This latency is imperceptible in normal highway driving where direction changes occur over 1–3 seconds.

Second, the dual-method speed estimator was reduced to single-method variants. Method A alone (pixel displacement) produced unrealistic speed readings for stationary following vehicles whose centroids shift less than 1 pixel per frame. Method B alone (distance change) produced spike readings for laterally moving vehicles changing bounding box size rapidly. The dual-method maximum eliminated both failure modes: 94% of frames produced speed estimates within 15 km/h of the expected absolute vehicle speed, compared to 71% for Method A alone and 68% for Method B alone on the highway sequence.

E. Comparison with Prior Systems

TABLE V Comparison with Related ADAS Systems

System	Sensor	GPU	Fns.	Cost
Bertozzi [10]	Stereo	No	2	High
LaneNet [7]	1 cam	Yes	1	Med+GPU
Ultra Fast [8]	1 cam	Yes	1	Med+GPU
YOLO-ADAS [1]	1 cam	Yes	3	Med+GPU
Commercial	LiDAR+Rad	Yes	6+	INR 5L+
Proposed	1 cam	No	6	INR 1,500

VII. DEPLOYMENT AND LIMITATIONS

A. Real Vehicle Deployment

Transitioning from recorded video to live camera input requires a single configuration change: VIDEO_PATH = 0 selects the default system camera in OpenCV. No other code modifications are needed. The camera should be mounted behind the windshield at the vehicle centreline at the height of the rear-view mirror, ensuring a clear forward view centred on the lane ahead. Camera calibration using a 9×6 checkerboard target with OpenCV's calibrateCamera function provides the precise focal length in pixels, improving distance estimation accuracy; lens distortion correction additionally improves lane detection linearity at image periphery for wide-angle dashcam optics.

System output is displayed on any HDMI-compatible dashboard monitor. Audio DANGER alerts are generated by a non-blocking Python thread invoking a text-to-speech engine on status transitions, ensuring audio warnings do not interrupt the video processing loop. The complete embedded deployment — Raspberry Pi 4B (4 GB), consumer dashcam, 7-inch

HDMI touchscreen, and USB-C power adapter connected to the vehicle's 12V accessory socket — has a total cost of approximately INR 8,000–12,000 and draws approximately 7 watts. No connection to the vehicle's CAN bus, OBD port, or any electrical system is required, making installation straightforward for any vehicle owner.

B. Limitations and Future Work

Four limitations are identified. First, lane detection degrades on worn or absent markings, in heavy rain, and at night below ~50 lux. Second, distance estimation error grows with range as bounding box pixel width narrows. Third, optical flow ego speed is sensitive to camera shake and rapid illumination changes. Fourth, in-lane classification may misclassify boundary-edge vehicles during momentary lane centre perturbation. Planned improvements include: lightweight deep learning lane segmentation for adverse conditions; stereo camera depth maps for sub-metre distance accuracy; Kalman trajectory prediction for time-to-collision estimation; automatic focal-length calibration via structure-from-motion; and ISO 26262 ASIL-B compliance study for emergency braking integration.

VIII. CONCLUSION

This paper presented a comprehensive Vision-Based Adaptive Driver Assistance System that delivers six safety-critical functions — dual-lane boundary detection with overlay visualization, hysteresis-stabilised steering prediction, multi-class vehicle detection and tracking, metric distance estimation, three-tier collision warning, and dual-method speed estimation — using exclusively a single consumer-grade monocular camera at INR 1,500 hardware cost, without GPU acceleration. The system represents the first published integration of all six functions in a CPU-only pipeline at this cost point.

The principal algorithmic contributions are validated through controlled experiments on 2,100 frames of real Indian road footage. The novel three-source lane edge pipeline (Canny + yellow HSV + white brightness) achieves 87% simultaneous dual-lane detection, outperforming the single-source Canny baseline by 18 percentage points. The position-based Hough segment assignment with four-stage anti-crossing validation completely eliminates the crossing artefact of classical slope-sign methods. The 5-frame hysteresis steering state machine reduces display flicker from 23% to below 3% of frames. The dual-method vehicle speed estimator achieves 94% of frames within 15 km/h of expected absolute speed, compared to 71% for pixel-displacement alone and 68% for distance-change alone.

From a deployment perspective, the complete embedded system — Raspberry Pi 4B, dashcam, and display — costs under INR 12,000 and draws 7 watts from a standard vehicle USB port without any vehicle modification. This cost-performance profile makes the system uniquely suitable for retrofitting the hundreds of millions of vehicles in cost-sensitive markets such as India that currently have no electronic driver assistance. Future work will incorporate lightweight deep learning lane segmentation for night and rain conditions, stereo camera depth estimation for improved range accuracy, Kalman trajectory prediction for time-to-collision alerting, and an ISO 26262 compliance pathway for active braking integration.

ACKNOWLEDGMENT

The authors thank Ms. P. Sudha, Assistant Professor, Department of Artificial Intelligence and Data Science, Dhanalakshmi Srinivasan University, Samayampuram, Tamil Nadu, for expert guidance and support throughout this project. The authors also acknowledge the OpenCV, Ultralytics YOLOv8, NumPy, and Python communities.

REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proc. IEEE CVPR, Las Vegas, NV, Jun. 2016, pp. 779–788.

[2] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," GitHub, Jan. 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[3] G. Farneback, "Two-frame motion estimation based on polynomial expansion," in Proc. SCIA, Gothenburg, Jun. 2003, pp. 363–370.

[4] B. K. Horn and B. G. Schunck, "Determining optical flow," Artificial Intelligence, vol. 17, no. 1–3, pp. 185–203, Aug. 1981.

[5] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in

pictures," Commun. ACM, vol. 15, no. 1, pp. 11–15, Jan. 1972.

[6] J. Canny, "A computational approach to edge detection," IEEE Trans. Pattern Anal. Mach. Intell., vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[7] D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, and L. Van Gool, "Towards end-to-end lane detection: An instance segmentation approach," in Proc. IEEE IV, Changshu, Jun. 2018, pp. 286–291.

[8] Z. Qin, H. Wang, and X. Li, "Ultra fast structure-aware deep lane detection," in Proc. ECCV, Glasgow, Aug. 2020, pp. 276–291.

[9] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," Int. J. Robot. Res., vol. 32, no. 11, pp. 1231–1237, Sep. 2013.

[10] M. Bertozzi and A. Broggi, "GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection," IEEE Trans. Image Process., vol. 7, no. 1, pp. 62–81, Jan. 1998.

[11] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, "Spatial as deep: Spatial CNN for traffic scene understanding," in Proc. AAAI, New Orleans, Feb. 2018, pp. 7276–7283.

[12] Y. Zhang et al., "ByteTrack: Multi-object tracking by associating every detection box," in Proc. ECCV, Tel Aviv, Oct. 2022, pp. 1–21.

[13] S. M. Grigorescu, B. Trasnea, T. T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," J. Field Robot., vol. 37, no. 3, pp. 362–386, May 2020.

[14] World Health Organization, "Global Status Report on Road Safety 2023," WHO Press, Geneva, 2023.

[15] G. Bradski, "The OpenCV library," Dr. Dobb's Journal of Software Tools, vol. 25, pp. 120–125, 2000.

[16] S. Sivaraman and M. M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," IEEE Trans. Intell. Transp. Syst., vol. 14, no. 4, pp. 1773–1795, Dec. 2013.

[17] C. Godard, O. Mac Aodha, M. Firman, and G. Brostow, "Digging into self-supervised monocular depth prediction," in Proc. IEEE/CVF ICCV, Seoul, South Korea, Oct. 2019, pp. 3828–3838.

[18] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," arXiv preprint arXiv:2004.10934, Apr. 2020.

[19] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," arXiv preprint arXiv:1804.02767, Apr. 2018.

[20] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in Proc. IEEE ICIP, Beijing, China, Sep. 2017, pp. 3645–3649.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE CVPR, Las Vegas, NV, Jun. 2016, pp. 770–778.

[22] T. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in Proc. IEEE CVPR, Honolulu, HI, Jul. 2017, pp. 2117–2125.

[23] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in Proc. ECCV, Amsterdam, Netherlands, Oct. 2016, pp. 21–37.

[24] R. Girshick, "Fast R-CNN," in Proc. IEEE ICCV, Santiago, Chile, Dec. 2015, pp. 1440–1448.