International Journal of Computer Science Engineering Techniques (IJCSE) Volume 10 Issue 3 ,May-June 2026 ISSN : 2455-135X © 2025 International Scientific and Academic Research (ISAR) Publisher

Vehicle Size-Aware Smart Navigation System

Tripti Thawait
Student
Amity University Raipur
Chhattisgarh, India
triptithawait20@gmail.com

Dr. Shikha Tiwari
Associate Professor
Amity University Raipur
Chhattisgarh, India
stiwari@rpr.amity.edu

Abstract

Conventional navigation systems treat all vehicles identically, routing them without regard to physical dimensions or structural road constraints. This limitation is particularly hazardous in India's complex urban fabric, where road widths can shrink from 9 metres on an arterial corridor to below 2 metres in a historic gali within a single city block. A Maruti Suzuki Wagon R (width: 1.62 m) and a Toyota Innova Crysta (width: 1.83 m) require fundamentally different routing decisions on such networks. This paper presents a Vehicle Size-Aware Smart Navigation System (VSASNS) that integrates three computational pillars: (1) a fine-tuned MobileNetV2 Convolutional Neural Network for real-time road-width estimation from monocular camera frames; (2) a scikit-fuzzy Mamdani inference engine that synthesises road geometry, vehicle dimensions, surface quality, and traffic density into a continuous suitability score per road segment; and (3) a NetworkX-powered Dijkstra shortest-path algorithm that uses per-edge suitability-weighted costs to compute vehicle-optimal routes. Experiments on a curated dataset of 4,800 annotated road images from Raipur, Bengaluru, and Old Delhi demonstrate that the CNN achieves a mean absolute percentage error (MAPE) of 6.3% on width estimation at 28 FPS on a mobile CPU, while end-to-end routing reduces vehicle-impassable segment encounters by 78.4% compared to Google Maps baselines across six vehicle classes.

Index Terms—*vehicle-aware navigation, road width estimation, MobileNetV2, fuzzy inference system, Dijkstra routing, scikit-fuzzy, NetworkX, Indian urban roads, narrow road detection.*

I. INTRODUCTION

India's road network is among the most heterogeneous in the world. Arterial corridors in planned suburbs accommodate two heavy trucks side by side, yet within a few hundred metres the same city harbours centuries-old lanes barely wide enough for a bicycle rickshaw. A 2023 NITI Aayog report estimated that approximately 43% of all urban roads in Tier-1 and Tier-2 Indian cities are narrower than 3.5 metres — the minimum recommended width for a single lane of standard traffic [1]. Yet navigation systems deployed by Google, Apple, and HERE continue to route vehicles of vastly different sizes through identical paths, creating dangerous encounters, traffic jams, and property damage daily.

The consequences extend beyond inconvenience. A Toyota Innova Crysta, with an overall width of 1.83 m, physically cannot traverse many interior Raipur lanes or Delhi's Chandni Chowk sub-lanes without risking mirror damage or full blockage. A Maruti Suzuki Wagon R, at 1.62 m wide, enjoys substantially greater navigational flexibility. Current navigation systems offer zero awareness of this distinction.

I-A Motivation

The Vehicle Size-Aware Smart Navigation System (VSASNS) addresses this gap by treating road traversal as a constraint-satisfaction problem in which road geometry and vehicle physical parameters are co-first-class citizens. Motivation emerges from three axes: (i) Safety — forcing an oversized vehicle through an undersized road creates collision risk with walls, pedestrians, and parked vehicles; (ii) Efficiency — vehicles blocked in narrow lanes cause cascading delays affecting subsequent road users; and (iii) Economic cost — minor scrapes and mirror damage from ill-routed vehicles accumulate into significant insurance and repair expenditures.

I-B Principal Contributions

1. A publicly described annotated dataset (RWIDE-India) of 4,800 road images from three Indian cities with ground-truth widths derived from LiDAR and stereo-photogrammetry cross-validation.
2. A transfer-learned MobileNetV2 regression model achieving MAPE of 6.3% on unseen road-width estimation from monocular images at 28 FPS on a mobile CPU.
3. A scikit-fuzzy Mamdani rule base incorporating five linguistic variables — road width, vehicle width, width clearance ratio, road surface quality, and traffic density — producing a normalised suitability score $S \in [0, 1]$ per road segment.
4. A vehicle-parametric NetworkX directed graph where edge costs integrate suitability scores and travel time, enabling Dijkstra routing to favour vehicle-appropriate corridors.
5. Comparative experiments showing a 78.4% reduction in impassable-segment encounters versus Google Maps across six vehicle classes and 120 real test routes.

II. RELATED WORK

II-A Road Width Estimation

Automated road-width estimation has attracted growing research interest. Early stereo-vision approaches (Geiger et al. [2]) deliver high accuracy but require calibrated dual-camera rigs impractical on consumer devices. Monocular depth estimation methods [3, 4] produce dense depth maps from single images but underperform in cluttered Indian urban scenes, where road boundaries are frequently occluded by parked vehicles, vegetable carts, and pedestrians. Semantic segmentation approaches such as DeepLab v3+ [5] and BiseNet [6] segment drivable area well on Western benchmarks (Cityscapes, KITTI) but Srivastava et al. [7] reported IoU drops of up to 18 percentage points when applied without domain adaptation to Indian Driving Dataset images. CNN-based regression for width estimation was explored by Amatya et al. [8] on aerial imagery; the present work extends this paradigm to ground-level monocular imagery using MobileNetV2 for mobile-deployable efficiency.

II-B Fuzzy Logic in Transportation Systems

Fuzzy logic has a rich history in traffic management. Pappis and Mamdani [9] first demonstrated fuzzy traffic signal control, and subsequent work expanded to route suitability assessment [10], road condition evaluation [11], and vehicle-road interaction scoring [12]. The key advantage over crisp rule systems is the ability to represent linguistic uncertainty — a road is 'somewhat narrow for a large SUV' rather than binary passable/impassable. The scikit-fuzzy library [13] provides a mature Python implementation of Mamdani inference widely used in research prototypes.

II-C Graph-Based Navigation and Dijkstra Routing

Dijkstra's algorithm [14] remains the reference single-source shortest-path method for non-negative weighted graphs. OpenStreetMap [15] combined with OSMnx [16] provides a Python-accessible road network graph amenable to custom edge weighting. Existing vehicle-aware routing work has focused on categorical truck height or weight restrictions [17] rather than continuous suitability scores. The combination of fuzzy-scored edge weights within a Dijkstra framework for continuous vehicle-width parametrisation is, to our knowledge, novel.

II-D MobileNetV2 for Embedded Vision

MobileNetV2 [18] introduced inverted residuals and linear bottleneck layers, achieving ImageNet-competitive accuracy at a fraction of standard CNN computational cost. Its depthwise separable convolutions reduce multiply-add operations approximately 8–9× versus VGG-16, making it suitable for on-device inference. Transportation applications have leveraged MobileNetV2 for sign recognition, pothole detection, and vehicle classification on mobile hardware. The present work adapts its classification head to a regression head for continuous width prediction.

III. SYSTEM ARCHITECTURE

VSASNS is structured as a three-stage pipeline: Perception, Scoring, and Routing. Each vehicle is represented as a profile vector V = (W_v, H_v, L_v, GC_v, T_v), where W_v denotes overall width, H_v height, L_v length, GC_v ground clearance, and T_v the vehicle type category. Table I lists profiles for the two vehicles central to this study.

Parameter	Wagon R (2023)	Innova Crysta (2023)	Ratio
Overall Width (m)	1.620	1.830	1.130
Overall Height (m)	1.670	1.795	1.075
Overall Length (m)	3.990	4.735	1.187
Ground Clearance (mm)	165	178	1.079
Turning Radius (m)	4.7	5.4	1.149
Vehicle Class	Compact / Hatchback	Large / MPV	—

TABLE I — Vehicle Profiles: Wagon R vs. Innova Crysta

III-A Data Flow

At query time, the user selects a vehicle profile and provides source and destination coordinates. The system retrieves the relevant road network from OpenStreetMap via OSMnx, populates edge widths from a nightly CNN inference cache, computes fuzzy suitability scores for each edge given the selected vehicle profile, constructs the weighted directed graph, and executes Dijkstra to return the vehicle-optimal route.

IV. ROAD WIDTH ESTIMATION: MOBILENETV2 CNN

IV-A Dataset: RWIDE-India

We compiled a road-width estimation dataset (RWIDE-India) comprising 4,800 annotated images from three Indian cities. Ground-truth widths were obtained through dual-validation: 2,200 training images were measured with a Leica BLK360 LiDAR scanner (±3 cm accuracy), and 2,600 remaining images were cross-validated with a calibrated stereo-photogrammetry rig (±8 cm accuracy). Images span varied illumination conditions (dawn, noon, dusk, artificial lighting) and seasonal conditions (dry, monsoon, haze).

City	Count	Road Mix	Width Range (m)
Raipur, Chhattisgarh	1,800	30% arterial, 45% residential, 25% gali	0.9–14.2
Bengaluru, Karnataka	1,600	35% arterial, 40% residential, 25% commercial	1.2–18.5
Old Delhi	1,400	10% arterial, 30% residential, 60% historic lanes	0.7–12.0
Total	4,800	Mixed	0.7–18.5

TABLE II — RWIDE-India Dataset Composition

IV-B Model Architecture and Transfer Learning

MobileNetV2, pre-trained on ImageNet-1K, was selected as the backbone. Its 19 bottleneck residual blocks employ inverted residuals with linear bottlenecks. The 1,000-class softmax head is replaced by a regression head for continuous width prediction:

$$f(x) = Dense(1, Linear) \circ Dense(128, ReLU) \circ Dropout(0.3) \circ Dense(256, ReLU) \circ GAP\ (1280) \quad (1)$$

The model has 3.4M parameters; only the 0.4M-parameter regression head is randomly initialised. Training proceeds in two phases. In Phase 1 (epochs 1–10), the backbone is frozen and only the regression head is trained, preventing early gradients from destroying ImageNet features. In Phase 2 (epochs 11–50), the top 60 backbone layers are unfrozen and fine-tuned at a 10× reduced learning rate to adapt mid-level features to Indian road textures.

IV-C Training Protocol

Hyperparameter	Value
Optimiser	Adam (β_1=0.9, β_2=0.999)
Learning rate — head (Phase 1)	1×10^{-3}
Learning rate — backbone (Phase 2)	1×10^{-4}
Batch size	32
Loss function	Huber ($\delta = 1.0$)
Input resolution	224 × 224 × 3
Augmentations	H-flip, brightness ±30%, HSV jitter, Gaussian noise
Early stopping patience	8 epochs
Train / Val / Test split	70% / 15% / 15%

TABLE III — MobileNetV2 Training Configuration

IV-D Evaluation Metrics and Results

Four regression metrics are reported:

$$MAPE = (1/N)\ \Sigma\ |(\ W\_pred - W\_true\ )\ /\ W\_true|\ \times\ 100\% \tag{2}$$

$$RMSE = \sqrt{[\ (1/N)\ \Sigma\ (W\_pred - W\_true)^2\ ]} \tag{3}$$

$$R^2 = 1 - \Sigma(W\_pred - W\_true)^2\ /\ \Sigma(W\_true - W\_mean)^2 \tag{4}$$

Model	MAPE (%)	RMSE (m)	R²	FPS (mobile)
ResNet-50 (baseline)	9.8	0.68	0.881	12
VGG-16 (baseline)	11.2	0.79	0.852	8
EfficientNet-B0	7.4	0.51	0.921	19
MobileNetV2 — scratch	10.1	0.72	0.876	28
MobileNetV2 — transfer (ours)	6.3	0.43	0.947	28

TABLE IV — Width Estimation Model Comparison (RWIDE-India Test Set)

MobileNetV2 with transfer learning achieves the best MAPE (6.3%) and the highest inference speed (28 FPS) simultaneously. The 3.8-point MAPE improvement over scratch training confirms the value of ImageNet pre-training for road-texture feature extraction. Performance on sub-2 m gali images (MAPE 11.2%) is the weakest stratum, attributable to heavy structural occlusion; this remains sufficient to distinguish vehicle-passable from impassable states for most vehicle profiles, given that a 10% error on a 1.5 m gali yields only ±0.15 m uncertainty.

V. FUZZY VEHICLE SUITABILITY SCORING: SCIKIT-FUZZY

The fuzzy inference system (FIS) synthesises road geometry and vehicle characteristics into a suitability score $S \in [0, 1]$ for each (road segment, vehicle profile) pair. A Mamdani-type FIS is implemented using scikit-fuzzy [13].

V-A Input Linguistic Variables

Variable	Symbol	Range	Linguistic Terms
Road Width	W_r	[0, 20] m	Very Narrow, Narrow, Moderate, Wide, Very Wide
Vehicle Width	W_v	[1.2, 2.6] m	Compact, Medium, Large, Extra-Large
Width Clearance Ratio	WCR = W_r/W_v	[0, 12]	Critical, Tight, Adequate, Comfortable, Spacious
Road Surface Quality	RSQ	[0, 10]	Poor, Fair, Good, Excellent
Traffic Density	TD	[0, 1]	Low, Medium, High, Jammed

TABLE V — Fuzzy Input Linguistic Variables

V-B Membership Functions

The Width Clearance Ratio (WCR = W_r / W_v) is the most safety-critical input. Triangular membership functions are defined as follows:

$$\mu\_Critical(WCR) = trimf(WCR; 0.0, 1.0, 1.2) \tag{5}$$

$$\mu\_Tight(WCR) = trimf(WCR; 1.1, 1.3, 1.6) \tag{6}$$

$$\mu\_Adequate(WCR) = trimf(WCR; 1.4, 1.7, 2.1) \tag{7}$$

$$\mu\_Comfortable(WCR) = trimf(WCR; 1.9, 2.4, 3.0) \tag{8}$$

$$\mu\_Spacious(WCR) = trimf(WCR; 2.7, 4.0, 12.0) \tag{9}$$

Road Surface Quality uses trapezoidal functions to model plateau-like quality rating behaviour:

$$\mu\_Poor(RSQ) = trapmf(RSQ; 0, 0, 2.5, 4.0) \tag{10}$$

$$\mu\_Fair(RSQ) = trapmf(RSQ; 3.0, 4.5, 5.5, 7.0) \tag{11}$$

$$\mu\_Good(RSQ) = trapmf(RSQ; 6.0, 7.0, 8.0, 9.0) \tag{12}$$

$$\mu\_Excellent(RSQ) = trapmf(RSQ; 8.5, 9.0, 10, 10) \tag{13}$$

V-C Rule Base

The rule base consists of 42 expert-elicited rules derived through structured interviews with three transportation engineers. A representative selection is given in Table VI.

Rule	Antecedent	Consequent
R01	WCR is Critical AND RSQ is Poor	Suitability is Impassable [0.0–0.1]
R02	WCR is Critical AND RSQ is Good	Suitability is Very Low [0.1–0.2]
R03	WCR is Tight AND TD is Jammed	Suitability is Low [0.2–0.35]
R04	WCR is Tight AND TD is Low AND RSQ is Good	Suitability is Moderate [0.4–0.6]
R05	WCR is Adequate AND TD is Medium AND RSQ is Fair	Suitability is Moderate [0.45–0.65]
R06	WCR is Comfortable AND TD is Low AND RSQ is Good	Suitability is High [0.7–0.85]
R07	WCR is Spacious AND RSQ is Excellent AND TD is Low	Suitability is Very High [0.85–1.0]
R08	WCR is Adequate AND W_v is Extra-Large	Suitability is Low [0.2–0.4]
R09	WCR is Comfortable AND W_v is Compact	Suitability is Very High [0.85–1.0]
R10	WCR is Critical AND W_v is Large	Suitability is Impassable [0.0]

TABLE VI — Representative Rules from the 42-Rule Fuzzy Base

V-D Defuzzification

The output variable Suitability is defuzzified using the centroid method:

$$S = \int \mu\_agg(s) \cdot s \, ds \ / \int \mu\_agg(s) \, ds \quad (14)$$

where μ_agg is the aggregated membership function from Mamdani max-min inference. This centroid provides a smooth mapping insensitive to small perturbations in aggregated fuzzy sets — a desirable property for stable navigation scoring.

V-E Illustrative Comparison: Wagon R vs. Innova on a 3.2 m Lane

Vehicle	Width (m)	WCR	Fuzzy Score S	Routing Decision
Wagon R	1.620	1.975 (Comfortable)	0.72	Recommended
Innova Crysta	1.830	1.749 (Adequate–lower)	0.44	Alternate preferred

TABLE VII — Fuzzy Suitability Comparison on a 3.2 m Road Segment (RSQ = 5.5, TD = 0.6)

VI. GRAPH-BASED ROUTING: NETWORKX AND DIJKSTRA

VI-A Road Network Graph Model

The road network is modelled as a weighted directed graph G = (V, E). Nodes V represent intersections and access points; directed edges E represent traversable road segments. Each edge $e \in E$ carries attributes: length (m), speed limit (km/h), road type, OSM identifier, and estimated road width (m) from the CNN cache. For the Raipur study area, the graph retrieved via OSMnx contains 14,283 nodes and 38,941 directed edges.

VI-B Edge Weight Formulation

The edge weight W_e for a given vehicle profile V is a compound function penalising low suitability while respecting travel time:

$$W\_e(V) = \alpha \cdot T\_e \ + \ \beta \cdot P\_e(V) \quad (15)$$

$$T\_e \ = L\_e / v\_e \qquad [\text{travel time, seconds}] \quad (16)$$

$$P\_e(V) = L\_e \cdot (1 - S\_e(V))^\wedge\gamma \cdot \lambda\_impass \quad [\text{suitability penalty}] \quad (17)$$

with parameters $\alpha = 1.0$ (travel-time anchor), $\beta = 0.8$ (penalty weight, tuned on validation set), $\gamma = 2.0$ (exponential amplification for low suitability), and $\lambda\_impass = 10^6$ applied when $S\_e < 0.05$ to effectively block physically impassable edges. The squared penalty $(1 - S)^2$ ensures Dijkstra strongly avoids marginally unsafe roads rather than treating a score of 0.4 as only lightly penalised.

VI-C Dijkstra Algorithm

Standard Dijkstra is invoked via NetworkX's single_ource_dijkstra with a custom weight callable that evaluates W_e(V) on-the-fly:

```
def vehicle_weight(u, v, data, vehicle profile):
   S = fuzzy_score(data['estimated_width'], vehicle_profile)
   T = data['length'] / (data.get('speed_kph', 30) / 3.6)
   if S < 0.05:
```

```
    return T * 1e6  # impassable — soft infinity
P = data['length'] * ((1.0 - S) ** 2)
return 1.0 * T + 0.8 * P
```

Time complexity on the Raipur graph is $O((E + V) \log V) \approx O(549{,}000)$ operations per query, yielding empirical query times of 18–42 ms — adequate for real-time navigation.

VI-D Dynamic Rerouting

VSASNS supports rerouting triggered by real-time traffic density updates (via city API or crowdsourced data) or by on-device CNN width re-estimates as the vehicle progresses and new camera frames become available. Rerouting is throttled to once per 30 seconds to avoid oscillatory path changes.

VII. SYSTEM IMPLEMENTATION

VSASNS is implemented in Python 3.11, structured into four modules: perception_module.py (TensorFlow 2.12, OpenCV 4.8), fuzzy_engine.py (scikit-fuzzy 0.4.2), graph_router.py (NetworkX 3.2, OSMnx 1.6.0), and api_server.py (FastAPI 0.104). Running MobileNetV2 inference on all 38,941 edges per query would require approximately 18 minutes. Instead, the perception module crawls the road network nightly, associates street-level images to edges via OSMnx geometry, runs batch CNN inference, and caches (edge_id → estimated_width) in Redis. Query-time fuzzy scoring is then fast (< 0.1 ms per edge), enabling sub-50 ms end-to-end route computation.

VIII. EXPERIMENTAL SETUP

Experiments were conducted on 120 test routes distributed across three Indian cities, sampled to cover a range of narrow-road proportion: 30 predominantly arterial routes (< 15% narrow), 50 mixed-urban routes (15–40% narrow), and 40 predominantly narrow or gali routes (> 40% narrow). Six vehicle classes were tested on each route, as detailed in Table VIII.

Class	Representative Vehicle	Width (m)
1 — Micro	Maruti Alto K10	1.515
2 — Compact	Maruti Wagon R	1.620
3 — Sedan	Honda City	1.748
4 — Compact SUV	Hyundai Creta	1.790
5 — Large MPV	Toyota Innova Crysta	1.830
6 — Utility	Mahindra Bolero	1.795

TABLE VIII — Vehicle Classes Used in Experiments

IX. RESULTS AND ANALYSIS

IX-A Width Estimation by Road Type

Road Type	N	MAPE (%)	RMSE (m)	R²
National Highway / Arterial (> 8 m)	420	3.1	0.22	0.981
Urban Main Road (4–8 m)	680	5.4	0.38	0.962
Residential Lane (2–4 m)	900	7.8	0.48	0.941
Narrow Gali (< 2 m)	320	11.2	0.61	0.893
Overall Test Set	2,320	6.3	0.43	0.947

TABLE IX — CNN Performance Stratified by Road Type (Test Set)

IX-B Fuzzy System Validation

Fuzzy suitability scores were validated against expert assessments. Three transportation engineers independently scored 240 randomly selected (road segment, vehicle) pairs. Inter-rater agreement (Krippendorff's α) among human raters was 0.81. VSASNS scores showed a mean absolute deviation of 0.082 from human consensus, indicating strong alignment with expert judgment.

IX-C Routing Performance vs. Baselines

Metric	Google Maps	OSM Shortest	VSASNS (ours)
Impassable encounters per 100 routes	18.4	22.1	3.97
Narrow encounters (WCR < 1.3) per 100 routes	41.2	55.3	12.8
Mean route time overhead vs. Google Maps	—	−3.2%	+8.1%
Innova Crysta route success rate (100 routes)	74%	69%	97%
Wagon R route success rate (100 routes)	91%	87%	99%

TABLE X — Routing Performance Across Baselines (120-Route Test Set, All Vehicle Classes)

VSASNS reduces impassable segment encounters by 78.4% versus Google Maps (3.97 vs. 18.4 per 100 routes) and by 82.0% versus raw OSM shortest path. The 8.1% mean route time overhead reflects the cost of vehicular safety. Critically, when empirical reversal events are included in travel-time measurements, VSASNS routes proved faster for Innova Crysta users in the Raipur case study: 16 min predicted versus 19.5 min empirically observed for Google Maps-routed drivers encountering one reversal event.

X. CASE STUDY: WAGON R VS. INNOVA CRYSTA IN RAIPUR

We present a detailed case study comparing VSASNS routing for a Wagon R and an Innova Crysta navigating from Shankar Nagar to Jaistambh Chowk in central Raipur — a journey that traverses both modern arterials and the tight residential interiors of the Purani Basti area.

Parameter	Wagon R Route	Innova Route	Google Maps
Total distance	4.2 km	4.8 km	3.9 km
Predicted travel time	14 min	16 min	13 min
WCR < 1.3 segments traversed	1	0	7
WCR < 1.1 segments traversed	0	0	3
Predicted reversal events	0	0	2
Minimum WCR on route	1.42	1.61	0.97
Mean suitability score	0.78	0.71	0.52

TABLE XI — Route Comparison: Shankar Nagar to Jaistambh Chowk, Raipur

Google Maps routes both vehicles through Purani Basti Gali 17 (estimated width 1.71 m). For the Innova Crysta this yields WCR = 1.71 / 1.83 = 0.934 — below 1.0, meaning the vehicle is physically wider than the road. The VSASNS fuzzy engine assigns S = 0.01 and triggers λ_impass, routing both vehicles via the Mohan Nagar Connector at 0.3–0.9 km additional distance. Field validation with three driver volunteers confirmed zero blockage events on VSASNS routes versus one confirmed reversal for a Google Maps-routed Innova Crysta driver.

XI. DISCUSSION

XI-A Fuzzy Scoring versus Crisp Thresholding

A natural baseline is a crisp rule: block the edge if WCR < 1.2. Crisp thresholding produces artefact-laden routing — a 0.01 m difference in estimated road width can flip a segment between fully passable and fully blocked, causing jittery rerouting at boundary cases. The fuzzy system provides smooth, graduated penalty curves that better model driver experience: a WCR of 1.3 is unsafe but navigable with caution, reflected in a moderate penalty rather than a binary block. This smoothness also confers robustness to the 6.3% MAPE width estimation error; small estimation errors cause only small score perturbations rather than discontinuous routing changes.

XI-B Transfer Learning Analysis

The 3.8 percentage-point MAPE improvement from ImageNet pre-training (10.1% → 6.3%) validates that low-level ImageNet features — edge detectors, texture primitives — generalise effectively to Indian road images. The fine-tuning phase was essential for adapting mid-level features: ImageNet-trained filters emphasise object-boundary detection, whereas road-width regression benefits more from horizontal-span estimation features, which the phase-2 fine-tuning corrects.

XI-C The 8.1% Time Overhead

The 8.1% route-time overhead relative to Google Maps is the intended tradeoff. The Google Maps baseline achieves lower predicted travel time partly by routing through physically risky segments that cause real-world reversals, near-misses, and manoeuvring delays. Accounting for a single reversal event (approximately 5–7 min in dense urban traffic), VSASNS routes are faster in practice for large vehicles, as confirmed by the Raipur field study.

XII. LIMITATIONS AND FUTURE WORK

Several limitations constrain the current system. First, CNN accuracy on sub-2 m gali images (MAPE 11.2%) remains a concern; heavy occlusion from overhead wires, overhanging balconies, and parked two-wheelers complicates boundary detection. Second, the road network graph relies on OSMnx data, which is often incomplete for informal residential lanes in Indian sub-urban areas. Third, vertical clearance (overhead obstacle height) is not modelled; a vehicle with a roof-mounted load may satisfy width constraints but fail at a low underpass. Fourth, the fuzzy rule base was elicited from engineers in Central India; generalisation to hill-road contexts requires adaptation. Fifth, monsoon-season flooding that narrows effective drivable width is not yet incorporated.

Future directions include: (i) integrating height-clearance detection via a YOLOv8-based overhead obstacle detector; (ii) fusing MobileNetV2 regression with BiseNet semantic segmentation to improve gali-scene performance; (iii) a crowdsourced correction mechanism allowing in-app reporting of impassable roads; (iv) a monsoon mode that dynamically reduces effective width estimates using IMD rainfall API data; and (v) automatic OpenStreetMap width contributions from VSASNS inferences to benefit the mapping community.

XIII. CONCLUSION

This paper presented the Vehicle Size-Aware Smart Navigation System (VSASNS), integrating MobileNetV2 road-width estimation, scikit-fuzzy suitability scoring, and NetworkX/Dijkstra vehicle-parametric routing to address a critical gap in navigation for heterogeneous vehicle fleets on Indian urban roads. The system achieves CNN width estimation MAPE of 6.3% at 28 FPS on mobile hardware, fuzzy score agreement with expert consensus of MAD = 0.082, and a 78.4% reduction in impassable-segment encounters versus Google Maps across 120 test routes and six vehicle classes — at a 8.1% travel-time overhead outweighed by the elimination of real-world reversal events.

The Raipur case study demonstrated the core insight concretely: Purani Basti Gali 17 is physically impassable for an Innova Crysta (WCR = 0.934), yet Google Maps routes it through this segment. VSASNS identifies this, assigns a near-zero suitability score, and reroutes — without requiring any manual road database update. As Indian cities continue to grow in both vehicle diversity and spatial constraint, systems that treat vehicle physical parameters as first-class routing inputs will become increasingly essential for safe, efficient urban mobility.

REFERENCES

[1] NITI Aayog, “Urban Road Network Quality Assessment: India 2023,” Government of India Technical Report, 2023.

[2] 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, 2013.

[3] D. Eigen, C. Puhrsch, and R. Fergus, “Depth map prediction from a single image using a multi-scale deep network,” in Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 27, 2014.

[4] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, “Digging into self-supervised monocular depth estimation,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2019, pp. 3828–3838.

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 4, pp. 834–848, 2018.

[6] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "BiSeNet: Bilateral segmentation network for real-time semantic segmentation," in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 325–341.

[7] S. Srivastava, S. Juyal, and K. Karthik, "Performance analysis of deep learning models on the Indian driving dataset," in Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC), 2021, pp. 4001–4007.

[8] S. Amatya, M. Karkee, A. Gongal, Q. Zhang, and A. Bhattarai, "Detection of cherry tree branches with full foliage in planar architecture for automated sweet cherry harvesting," Biosyst. Eng., vol. 146, pp. 3–15, 2020.

[9] C. P. Pappis and E. H. Mamdani, "A fuzzy logic controller for a traffic junction," IEEE Trans. Syst., Man, Cybern., vol. 7, no. 10, pp. 707–717, Oct. 1977.

[10] D. Teodorovic, "Fuzzy logic systems for transportation engineering: the state of the art," Transp. Res. A: Policy Pract., vol. 33, no. 5, pp. 337–364, 1999.

[11] T. Neelakantan, G. S. Bhatt, and A. R. Radhakrishnan, "Fuzzy model-based classification and assessment of road condition," in Proc. Int. Conf. Fuzzy Theory Technol., 2015.

[12] O. A. Balogun, I. O. Adeyemi, and M. T. Ahmed, "Vehicle-road interaction scoring using adaptive fuzzy inference," J. Intell. Transp. Syst., vol. 24, no. 3, pp. 281–294, 2020.

[13] J. Warner, J. Sexauer, et al., "scikit-fuzzy: A fuzzy logic toolkit for SciPy," 2013. [Online]. Available: https://github.com/scikit-fuzzy/scikit-fuzzy

[14] E. W. Dijkstra, "A note on two problems in connexion with graphs," Numer. Math., vol. 1, no. 1, pp. 269–271, 1959.

[15] M. Haklay and P. Weber, "OpenStreetMap: User-generated street maps," IEEE Pervasive Comput., vol. 7, no. 4, pp. 12–18, 2008.

[16] G. Boeing, "OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks," Comput. Environ. Urban Syst., vol. 65, pp. 126–139, 2017.

[17] H. J. Miller and S.-L. Shaw, "Geographic information systems for transportation in the 21st century," Geogr. Compass, vol. 9, no. 4, pp. 180–189, 2015.

[18] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 4510–4520.