WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal Homepage: www.ijrpr.com ISSN 2582-7421

Adaptive Path Planning for Unstructured Road Navigation

[1]G. Nirosha, [2]B. Tanuja, [2]B. Ramadevi, [2]B. Naga Shailendra, [2]D. Bhargava Naidu

[1] Assistant Professor, Department of Computer Science and Engineering (CSE),2Students, Department of Computer Science and Engineering (CSE) GMR Institute of Technology (Deemed to be University), Rajam, Andhra Pradesh, India
Email ID: tanujabuddepu@gmail.com , ramadevibaki3@gmail.com , bendinagashailendra@gmail.com , bhargavnaidu527@gmail.com, nirosha.g@gmritdu.edu.in

ABSTRACT:

The abstract describes how both the absence of appropriate lanes, reckless actions on the part of both road users and animals, and lack of established flow on the road cannot allow us to implement self-driving vehicles in India. Methods of navigation based on maps of lanes and geometrical laws cannot effectively adapt to such environment conditions. The new mechanism developed by us, solely based on the use of software enables us to ensure adaptive path planning and collision avoidance for autonomous vehicles in the case of poor road conditions in India. The system consists of eight stages starting from the input of cameras and ending with the control of the vehicle, including detection by YOLO, perception of drivable space, localisation, map interface, motion prediction including the collision risk estimate, and adaptive path planning. The process of planning the paths is carried out in the Frenet frame and paths are assessed by means of an adaptive cost taking into account the clearance from the obstacles.to incorporate real sensors for advanced driver assistance systems, autonomous driving, and intelligent traffic management.

Index Terms **-** Autonomous vehicles, adaptive path planning, collision avoidance, unstructured road navigation, dynamic obstacle detection, Frenet frame, CARLA.

1. Introduction

On well-managed highways, lane markings, detailed maps and predictable traffic behaviour give autonomous vehicles a structure that planning software can rely on. Outside such settings, that structure largely vanishes. Worn or absent markings, unpaved surfaces, unexpected obstacles and informal driving customs cause conventional navigation stacks, which lean on lane-level maps and rigid geometric rules, to lose track of the scene and to hesitate or choose unsafe trajectories [1], [8].

Roads in India are extremely complicated due to one particular reason. The cyclists traverse the trail without any signs, animals appear on the route, vehicles are halted on the road, and there are many potholes. Within this environment, it is the responsibility of the car to continue monitoring the surroundings and be able to make crucial decisions rapidly while driving smoothly and comfortably.

As Section II shows, perception and planning are mostly studied in isolation; much of the planning research addresses terrain or off-road driving, and moving obstacles such as pedestrians and animals receive only partial treatment. We therefore propose a system that joins object detection, road understanding, collision-risk analysis and path planning in a single pipeline. Our contributions are:

(1) an eight-stage architecture with a monitoring loop linking camera input to vehicle control; (2) Frenet-frame trajectory sampling with an adaptive cost whose weights follow the situation; (3) collision-risk estimation that assumes road users are uncooperative and feeds the planner directly; and (4) an evaluation protocol built on CARLA scenarios modelled on Indian roads.

Section II surveys related work. Section III presents the methodology, Section IV the evaluation plan, Section V a discussion of how the design closes the identified gaps together with its limitations, and Section VI the conclusions.

2. Related Work

We examined 21 recent studies on navigation in unstructured environments and organize them into three groups: terrain-aware and off-road planning, search- and sampling-based planning, and perception for unstructured and Indian roads (Table 1).

2.1 Terrain-aware and off-road planning

According to Fan et al. [1], there is a single-stage planner which integrates semantic segmentation, Bayesian traversability estimation, entropy-based risk and MPPI that reduced travel time by 4.5% and path length by 24.7%. The subsequent research from the same authors indicates that navigation without GPS becomes possible by a monocular camera combined with NoMaD, FastSAM and CLIP, and tested on real vehicle [11]. In the study by Li et. al [20], an improved version of the A* algorithm is being introduced which provides feedback on dynamic constraints like geometric traversability, the stability of motion and the availability of energy input improving the results by more than 8%. Liu et al. [9] used A* algorithm with Dijkstra method and potential fields taking into account terrain limitations, whereas Li et al. [15] utilized Random Forest for vegetation classification (91.34% accuracy) incorporated in improved A*. Jerome et al. [6] performed the real-time off-road mapping and planning taking 1.5 seconds approximately, meanwhile Shi et al. [18] combined LiDAR and camera data in one grid applying Hybrid A* method, safety potential field and B-splines to increase the minimal distance for safety by more than 17 times compared to classical A* for the mining vehicles.

2.2 Search- and sampling-based planners

Man et al. [4] enhanced the Hybrid A* by incorporating the Reeds-Shepp curves and the smoothing process that minimizes the distance covered and increases the route quality. Zhu et al. [7] developed the two-layer Hybrid A* employing the dynamic phase windows and adaptive node expansion which decreased the planning time from 1.48 s to 0.65 s. However, Lee et al. [5] use the dynamic Bayesian filtering method to determine when the vehicle will find itself in a local minimum, thus enabling to issue the alternative route beforehand. Minařík et al. [21] accelerate the sampling-based motion planning in the cluttered 3D environments by applying the previously used paths (RRT-LIB) that result in an up to 85% reduction of the planning time. Zhou et al. [10] suggested multi-stage trajectory planning with global and local phases as well as conflict and deadlock resolution that is tested in different conditions including quarry area. Reda et al. [8] carry out a review of about 275 publications including 162 works on path planning covering classical, machine learning, meta-heuristic, and hybrid solutions.

2.3 Perception for unstructured and Indian roads

IndiSegNet [2] is a compact segmentation network that scores 67.2% mIoU on the Indian Driving Dataset (IDD) and runs at 112 FPS on a Jetson Nano. Cascade Drive [3] unites DeepLabv3+, YOLOv5 and YOLOv7 to detect lanes (97.26% accuracy), objects (0.92 mAP) and potholes (0.83 mAP). Wang et al. [12] avoid obstacles from a single camera using YOLOv5 and deep reinforcement learning, with detection accuracy between 78.76% and 88.26%. According to a study by Gowri Yaamini et al. [13], based on YOLOv8n, CBAM, and Ghost Convolution, lanes and traffic signs have been recognized on Indian roads with an output of 90.6% lane mAP and 117 FPS. In another study conducted by Chaman et al. [14], the effectiveness of YOLOv8 was compared with YOLOv12 for Advanced Driver Assistance Systems (ADAS), where YOLOv12 achieved 97.3% accuracy and 82.2% mAP@50–95. Selvarajan et al. [16] present a model in which they manage all road issues and detect lane geometries (with 0.877 anomaly mAP and 0.962 drivable area mIoU). Chen et al. [19] recognize drivable areas at 93.89% mIoU and 62.49 FPS on ORFD, and Chen et al. [17] contribute a 3D occupancy dataset of unstructured scenes.The IndiSegNet [2] is a small segmentation network with a performance of 67.2% mIoU on the Indian Driving Dataset (IDD) while functioning at the rate of 112 FPS on a Jetson Nano. Cascade Drive [3] integrates many recognition models, such as DeepLabv3+, YOLOv5, and YOLOv7 for the task of lane recognition with the accuracy of 97.26%, object recognition with 0.92 mAP, and pothole detection with 0.83 mAP. According to the study of Wang et al. [12], YOLOv5 is used for avoiding obstacles with the help of deep reinforcement learning and data from a single camera having the detection accuracy of 78.76-88.26%. The research of Gowri Yaamini et al. [13] shows that the use of YOLOv8n in combination with CBAM and Ghost Convolution makes it possible to recognize lanes and traffic signs on the Indian road with the output of 90.6 % of lane mAP and 117 FPS. Another study by Chaman et al. [14] compares the efficiency of YOLOv8 against YOLOv12 for the purpose of Advanced Driver Assistant Systems (ADAS) where YOLOv12 managed to achieve 97.3% of accuracy and 82.2% of mAP@50–95. In the study of Selvarajan et al. [16], the researchers developed a model for managing road problems and recognizing lane geometries with the parameters of 0.877 anomaly mAP and 0.962 drivable area mIoU. Furthermore, the research

Table 1: Summary of reviewed studies

Ref.	**Technique and key result**	**Limitation**
[1]	Bayesian traversability, entropy risk, MPPI; 24.7% shorter path	Terrain risk; not Indian traffic
[2]	IndiSegNet; 67.2% mIoU (IDD), 112 FPS	No planning
[3]	DeepLabv3+, YOLOv5/v7; 97.26% lane accuracy	Limited replanning
[4]	Improved Hybrid A*, Reeds-Shepp; smoother paths	No perception or dynamic users
[5]	Dynamic Bayesian filtering; early local-minimum detection	Local minima only
[6]	Intermediate maps, graph planning; about 1.5 s	Off-road focus
[7]	Dual-layer Hybrid A*; 1.48 s to 0.65 s	Terrain focus

Ref.	Technique and key result	Limitation
[8]	Review of 275 papers (162 on planning)	No new system
[9]	A*, Dijkstra, APF with terrain constraints	Static factors only
[10]	Global and local planning, deadlock solving	Vehicle conflicts only
[11]	Monocular camera, NoMaD, FastSAM, CLIP; GPS-free navigation	Not Indian traffic
[12]	YOLOv5 + DRL; 78.76–88.26% accuracy	Monocular depth, occlusion
[13]	YOLOv8n, CBAM; 90.6% lane mAP, 117 FPS	No planning
[14]	YOLOv8–v12 benchmark; 97.3% precision	Detection only
[15]	Random Forest + improved A*; 91.34% accuracy	Terrain and vegetation
[16]	YOLOv8 + Transformer; 0.877 anomaly mAP	Perception only
[17]	3D occupancy dataset	Dataset only
[18]	BEV + Hybrid A*; over 17x safety distance	Mining vehicles
[19]	BiSeNet + GCN; 93.89% mIoU, 62.49 FPS	No planning
[20]	Improved A*, constraint feedback; over 8% gain	No dynamic users
[21]	RRT-LIB, ICP; up to 85% faster	Not road-focused

2.4 Datasets and evaluation metrics

IDD, Cityscapes, CamVid, BDD100K, CMIRD, ORFD, RUGD, Rellis-3D, A2D2, AutoMine, and DeepScene rank among the most commonly used datasets in this realm. The performance in perception is described in terms of accuracy, precision, recall, F1-score, mAP, and mIoU, while speed is defined using FPS or inference time. Planning is conveyed through the metrics such as path length, planning time, and smoothness, whereas safety is measured by collision rate, clearance, and time-to-collision (TTC).

2.5 Research gap

The research presented above shows that perception and planning have often been treated differently, that the number of systems meant for unstructured roads in India is very limited, that there are still many dynamic obstacles that have not been accounted for, and that the issues relating to interactions with pedestrians, animals, potholes, and roads without markings still persists. At the same time, although there are several studies on the issue of perception [2], [3], [13], [14], [16], [19], they cannot be considered to developed enough to be used for collision avoidance applications. On the other hand, the studies on planning [1], [4]–[7], [9], [15], [18], [20], [21] do not pay any attention to real-time automation. The survey [8] and the data set [17] do not create a functioning navigation system. The only missing piece of the puzzle is unifying all components of perception, collision risk detection and real-time adaptive planning for unpredictable Indian roads.

3. Proposed Methodology

3.1 System architecture

The system is made up of 8 different stages which may be seen in Fig. 1 plus feedback loop that checks the end result and communicates any required adjustments back to the system. The YOLO detector identifies the cars, pedestrians and other types of objects in the image being captured by the camera. Perception then determines which parts of the scene are road and free space. Localization and a map interface supply the vehicle's position and destination. Next, the prediction stage examines how surrounding road users are moving and estimates the collision risk. The adaptive path planner receives all the information, which processes the global route and combines it with both a local planner and an adaptive cost function, along with dynamic replanning. Such information acts as a source for generating the commands responsible for controlling the movement of the vehicle. In case of any changes on the road, the appearance of a new object on the roadway, or if the car goes off track, the monitoring system informs the prediction function and the planning stage of the changes so that a new route can be created.

Development and testing take place in the CARLA simulator. Planning is done in the Frenet frame, where a point is described by its distance s along the road and its sideways offset d from the road's center line. Because the planner needs only the direction of the road, not exact lane geometry, this

representation works where markings are unclear. Position and speed are optimized jointly so the vehicle is never commanded to do something it physically cannot.

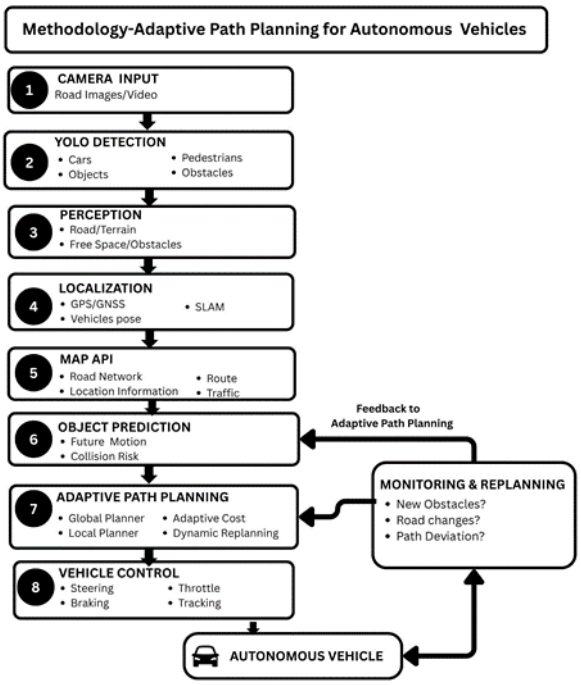


Fig. 1: Workflow architecture of the adaptive path planning system

3.2 Camera input and object detection

A camera mounted on the ego vehicle supplies images and video frames from the simulator. All frames are scaled and standardized and matched to the vehicle state during that time so that the detection can be positioned precisely with respect to the vehicle. The detection is done using YOLO method, which is a one-stage network that classifies the object, gives its bounding box and likelihood of the detection at a time, and is fast enough to be appropriate for a planner that is in eternal need of new data. The detector carries out vehicle, pedestrian, object, and obstacle detection. The existing model can be used out-of-the-box or fine-tuned based on the traffic clips similar to the Indian one; and if there are any uncertainties, the confidence levels are sent too. The specific YOLO version and size will be selected by trading accuracy against speed.

Perception and scene understanding

In addition to object recognition, the planner must also determine where the vehicle is able to travel. As such, during the perception stage of the process, the drivable area and terrain is estimated and free space is determined in relation to impediments such as cars, pedestrians, animals, potholes and stationary or temporary obstructions. Since lane markings may often be faded, missing or obscured, this estimate is made on the basis of the road surface itself instead of painted lines. In CARLA, depth is available from a depth sensor or directly from the simulator. With a real single camera, depth estimation is harder and objects can occlude one another, a limitation to bear in mind when reading the results.

Localization and map integration

GPS/GNSS offers accurate geographical location but using SLAM can improve position estimation in case of disrupted satellite signals, such as in urban canyons. Vehicle pose, which is a combination of position, heading, and speed, is then used by controller and planner. The mapping API gives information about road infrastructure and other points of interest but it is limited in accuracy because the vehicle can only utilize information on the basis of its onboard camera, which only shows part of route it is moving along. Thus, despite the fact that GPS data may be inaccurate sometimes, the vehicle generally can move normally.

Object prediction and collision-risk estimation

Knowing where an object is now is not enough; safe driving requires an estimate of where it will be shortly. The prediction stage tracks every detected vehicle and pedestrian and forecasts its motion over a short window. Road users are assumed to be uncooperative, since a two-wheeler may swerve, a pedestrian may step out and a driver may fail to slow, so a safety margin is kept around each one. The predicted motion is then compared against the candidate paths of the ego vehicle to gauge the danger of collision and identify areas that should be kept clear. The calculated risk value and clearance

zones are directly integrated into the cost function of the planner and are updated every time the monitoring module detects any changes. These values are merely estimates and therefore, the module tends to be conservative.

Adaptive path planning

Planning involves two components. The global planner picks the best overall route connecting the current position and destination using the map API road network. The local planner calculates many possible candidate trajectories of the Frenet frame such as holding the lane, deviating a little bit left or right or decelerating, keeping those which can be achieved by the vehicle physically. The adaptive cost function assesses each candidate trajectory τ as follows:

$$J(\tau) = \sum_k w_k(\sigma)\, C_k(\tau) \quad (1)$$

Where Ck includes the costs of proximity to the obstacle, expected probability of collision, curvature of the path, jerk in speed, progress on the route and offset from the route, with the weights wk depending on the current state σ. Such dependency is what makes the cost adaptive: in close proximity to a cluster of pedestrians, for instance, the cost of clearance is more important than speed. The position and speed are optimized simultaneously, and candidate trajectories with sharp turns or rapid speed changes are rejected even before they are sent to the controller.

The speed calculations include distance, curvature and jerk which are computed using vectorized NumPy and SciPy functions. In case when the monitoring component signals that the current path is not safe, the planner rejects it and calculates another path, known as dynamic replanning

3.7 Vehicle control and monitoring

The tracking controller translates the selected trajectory into steering, throttle, and brake commands, which evaluate the performance of the vehicle with respect to following the selected trajectory. Braking is smooth in case the threat rises gradually but strong when there is sudden appearance of an object. The Monitoring and Re-planning block completes the feedback loop by constantly evaluating whether there is any new object that appears, if there is any change in road conditions, or if there is deviation in vehicle path. Any of the above responses generates feedback that will be sent to prediction and planning blocks to select a new path. This enables the vehicle to handle situations like pedestrian appearing suddenly or potholes on the road without sudden maneuvering that might discomfort passengers.

3.8 Feedback and continuous improvement

There are two feedback loops which are employed: one operates during driving when the monitoring module continuously updates the prediction and planning modules; another operates between different experiments. Detections log, planned paths, obstacle clearance, jerk, collisions and close encounters are studied to make changes to cost weights, number of candidate paths, the length of the prediction horizon, conditions for the need in replanning, and detector parameters. Each new setting is verified on the same set of scenarios prior to its implementation. This allows improving the system configuration, but provides no guarantees for the unseen situation.

The development process involves the following five phases: (1) simulation set up with the maps of CARLA, ego vehicle, camera and scenarios for testing; (2) perception including detector and free space estimation; (3) localization and prediction which includes pose estimation, map API and collision risk estimation; (4) planning and control which implements the Frenet frame planning, adaptive cost function, replanning and control; and (5) evaluation and fine-tuning including the scenarios run and parameter adjustment.

4. Evaluation Plan

4.1 Data and information sources

The CARLA simulator supplies 3D environments, vehicles, pedestrians, sensors and control over scenarios. Camera images come from the simulated vehicle. Public datasets such as IDD, BDD100K and Cityscapes may serve to fine-tune the detector or check perception; IDD is particularly relevant because it consists of Indian road scenes, and each dataset will be vetted for suitability first. The map and routing information is extracted from the map API and the simulator map, while vehicle state information (position, orientation, and velocity) are used for localization, planning, and control purposes.

4.2 Test scenarios and baselines

Each and every element of the pipeline will be tested individually initially before testing the entire system collectively in CARLA. The test cases will mimic conditions on Indian roads such as lack of lane markings, pedestrians suddenly crossing the road, animals on the road, potholes, parked cars, unexpected obstacles, and moving vehicles. Tests will be replicable by using seed numbers and success conditions. The suggested planner will be compared with A*, Hybrid A* and a basic local planner and timing of computations will be used to establish that planning is done within the real-time constraint.

4.3 Evaluation metrics

Because detection, prediction, planning and control serve different purposes, they are assessed with different measures, listed in Table 2.

Table 2: Proposed evaluation metrics

Area	**Metrics**
Object detection	Precision, recall, F1-score, mAP, inference speed (FPS)
Perception	mIoU for drivable area; free-space accuracy
Localization	Position and heading error (RMSE)
Object prediction	Average and final displacement error (ADE, FDE); collision-risk accuracy
Path planning	Path length, planning time, curvature, longitudinal jerk, lateral acceleration
Safety	Collision rate, minimum obstacle clearance, time-to-collision (TTC)
Control	Lateral, heading and speed tracking error
Replanning	Replanning time, replanning success rate, route completion rate

This paper reports no accuracy, safety or performance figures. Numbers will be given only once the tests are complete, each with its test conditions and limitations.

5. Results and Discussion

5.1 Results

The planner was run in the CARLA simulator, and Figs. 2 to 12 show what it did. The results are qualitative. They come from the planner's console log and from screenshots of the runs, so they show behaviour and are not accuracy, safety or performance measurements.

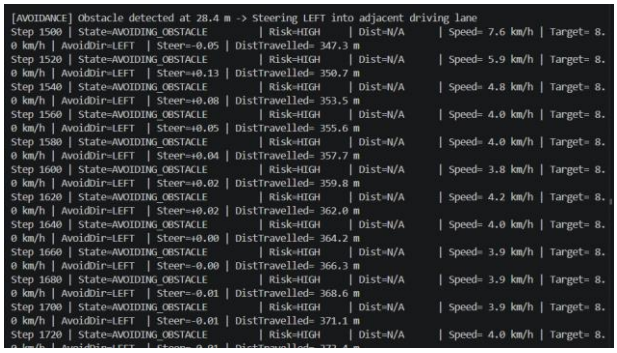


Fig. 2: Console log during the avoidance phase (steps 1500 to 1720)



Fig. 3: CARLA view: ego vehicle on a multi-lane urban road with parked vehicles



Fig. 4: Console log during cruising (steps 1300 to 1480) and start of avoidance at 28.4 m



Fig. 5: CARLA view: ego vehicle passing a stationary truck near a crosswalk



Fig. 6: CARLA view: ego vehicle between a moving car and a parked car

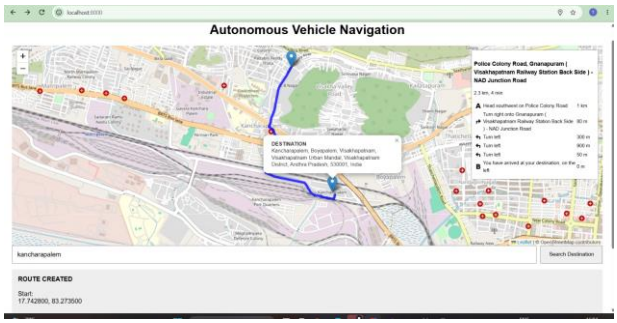


Fig. 7: Navigation interface: route to the searched destination

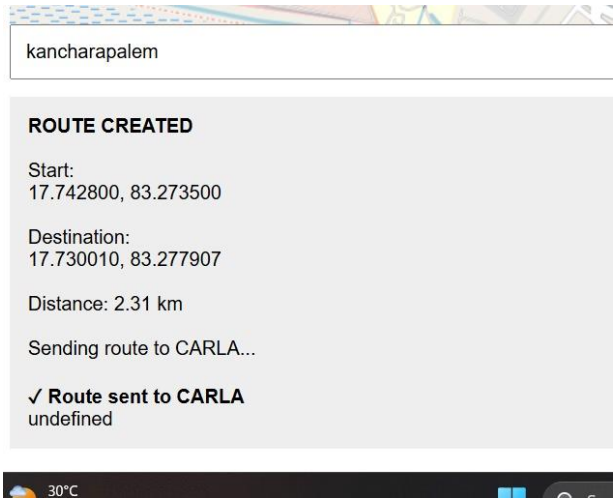


Fig. 8: Route summary (2.31 km) and confirmation that the route was sent to CARLA

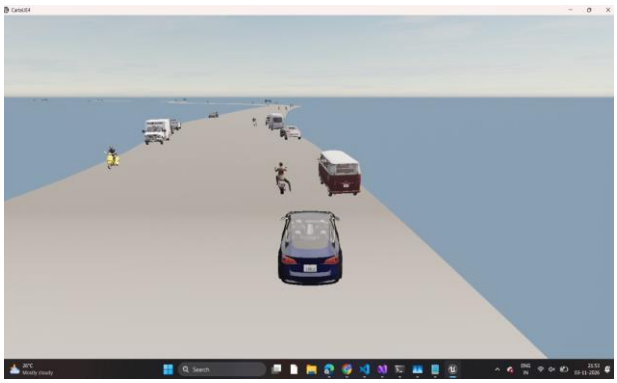


Fig. 9: CARLA view: open road without lane markings, obstacle in front of the ego vehicle

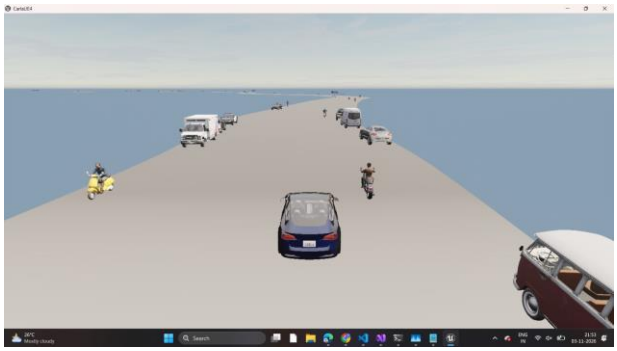
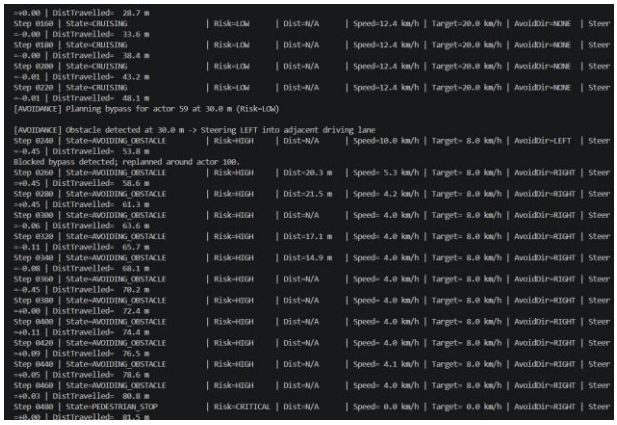


Fig. 10: CARLA view: ego vehicle in stable motion among surrounding traffic



```
=+0.00 | DistTravelled=  28.7 m
Step 0160 | State=CRUISING          | Risk=LOW      | Dist=N/A      | Speed=12.4 km/h | Target=20.0 km/h | AvoidDir=NONE  | Steer
=-0.00 | DistTravelled=  33.6 m
Step 0180 | State=CRUISING          | Risk=LOW      | Dist=N/A      | Speed=12.4 km/h | Target=20.0 km/h | AvoidDir=NONE  | Steer
=-0.00 | DistTravelled=  38.4 m
Step 0200 | State=CRUISING          | Risk=LOW      | Dist=N/A      | Speed=12.4 km/h | Target=20.0 km/h | AvoidDir=NONE  | Steer
=-0.01 | DistTravelled=  43.2 m
Step 0220 | State=CRUISING          | Risk=LOW      | Dist=N/A      | Speed=12.4 km/h | Target=20.0 km/h | AvoidDir=NONE  | Steer
=-0.01 | DistTravelled=  48.1 m
[AVOIDANCE] Planning bypass for actor 59 at 30.0 m (Risk=LOW)

[AVOIDANCE] Obstacle detected at 30.0 m -> Steering LEFT into adjacent driving lane
Step 0240 | State=AVOIDING_OBSTACLE | Risk=HIGH     | Dist=N/A      | Speed=10.0 km/h | Target= 8.0 km/h | AvoidDir=LEFT  | Steer
=-0.45 | DistTravelled=  53.8 m
Blocked bypass detected; replanned around actor 100.
Step 0260 | State=AVOIDING_OBSTACLE | Risk=HIGH     | Dist=20.3 m   | Speed= 5.3 km/h | Target= 8.0 km/h | AvoidDir=RIGHT | Steer
=+0.45 | DistTravelled=  58.6 m
Step 0280 | State=AVOIDING_OBSTACLE | Risk=HIGH     | Dist=21.5 m   | Speed= 4.2 km/h | Target= 8.0 km/h | AvoidDir=RIGHT | Steer
=+0.45 | DistTravelled=  61.3 m
Step 0300 | State=AVOIDING_OBSTACLE | Risk=HIGH     | Dist=N/A      | Speed= 4.0 km/h | Target= 8.0 km/h | AvoidDir=RIGHT | Steer
=-0.06 | DistTravelled=  63.6 m
Step 0320 | State=AVOIDING_OBSTACLE | Risk=HIGH     | Dist=17.1 m   | Speed= 4.0 km/h | Target= 8.0 km/h | AvoidDir=RIGHT | Steer
=-0.11 | DistTravelled=  65.7 m
Step 0340 | State=AVOIDING_OBSTACLE | Risk=HIGH     | Dist=14.9 m   | Speed= 4.0 km/h | Target= 8.0 km/h | AvoidDir=RIGHT | Steer
=-0.08 | DistTravelled=  68.1 m
Step 0360 | State=AVOIDING_OBSTACLE | Risk=HIGH     | Dist=N/A      | Speed= 4.0 km/h | Target= 8.0 km/h | AvoidDir=RIGHT | Steer
=-0.45 | DistTravelled=  70.2 m
Step 0380 | State=AVOIDING_OBSTACLE | Risk=HIGH     | Dist=N/A      | Speed= 4.0 km/h | Target= 8.0 km/h | AvoidDir=RIGHT | Steer
=+0.00 | DistTravelled=  72.4 m
Step 0400 | State=AVOIDING_OBSTACLE | Risk=HIGH     | Dist=N/A      | Speed= 4.0 km/h | Target= 8.0 km/h | AvoidDir=RIGHT | Steer
=+0.13 | DistTravelled=  74.4 m
Step 0420 | State=AVOIDING_OBSTACLE | Risk=HIGH     | Dist=N/A      | Speed= 4.0 km/h | Target= 8.0 km/h | AvoidDir=RIGHT | Steer
=+0.09 | DistTravelled=  76.5 m
Step 0440 | State=AVOIDING_OBSTACLE | Risk=HIGH     | Dist=N/A      | Speed= 4.1 km/h | Target= 8.0 km/h | AvoidDir=RIGHT | Steer
=+0.05 | DistTravelled=  78.6 m
Step 0460 | State=AVOIDING_OBSTACLE | Risk=HIGH     | Dist=N/A      | Speed= 4.0 km/h | Target= 8.0 km/h | AvoidDir=RIGHT | Steer
=+0.03 | DistTravelled=  80.8 m
Step 0480 | State=PEDESTRIAN_STOP   | Risk=CRITICAL | Dist=N/A      | Speed= 0.0 km/h | Target= 0.0 km/h | AvoidDir=RIGHT | Steer
=+0.00 | DistTravelled=  81.5 m
```

Fig. 11: Console log: bypass blocked and replanned, then pedestrian stop



Fig. 12: CARLA view: pedestrian crossing in front of the ego vehicle

The navigation page (Fig. 7) runs on localhost with Leaflet and OpenStreetMap data. For the destination Kancharapalem in Visakhapatnam it drew a 2.31 km route from 17.742800, 83.273500 to 17.730010, 83.277907, estimated at about 4 minutes, and listed the turn-by-turn instructions. The summary in Fig. 8 confirms that the route was sent to CARLA.

In the city scenes (Figs. 3, 5 and 6) the ego vehicle stays on its carriageway among parked cars, a stationary truck near a crosswalk and a car in the next lane, and it keeps clear of all of them.

Fig. 4 is the log of a straight stretch. From step 1300 to step 1480 the state is CRUISING with risk LOW, the speed is 12.4 km/h against a target of 20.0 km/h, and the steering stays between -0.06 and +0.01. The planner then reports a bypass for actor 40 at 28.4 m and steers left into the adjacent lane. From step 1500 the state is AVOIDING_OBSTACLE, the risk is HIGH and the target speed falls to 8.0 km/h. Fig. 2 continues the same run to step 1720: the speed drops from 7.6 km/h to about 4 km/h, the steering peaks at +0.13 and returns to near zero, and the vehicle gets past the obstacle.

Fig. 11 comes from a second run in which the first bypass does not work. After cruising at 12.4 km/h, the planner plans a bypass around actor 59 at 30.0 m and steers left (steering -0.45 at step 240). The log then reports that the bypass is blocked and replans around actor 100. The avoidance side switches from left to right (steering +0.45 at step 260) and the speed settles near 4 km/h. At step 480 the state becomes PEDESTRIAN_STOP with risk CRITICAL, and the speed goes to 0.0 km/h after about 81.5 m of travel. Fig. 12 shows the pedestrian crossing in front of the vehicle that causes this stop.

Figs. 9 and 10 show the vehicle on an open road with no lane markings, among motorcycles, a white truck and a maroon van. It keeps a steady position while the other vehicles move around it.

Table 3 lists the planner states read from the logs. In both runs the actual speed stayed below the target: about 4 km/h against 8 km/h during avoidance, and 12.4 km/h against 20 km/h while cruising.

Table 3: Planner states observed in the CARLA runs

State	**Risk level**	**Speed observed**	**Target speed**	**Steering command**
CRUISING	LOW	12.4 km/h	20.0 km/h	-0.06 to +0.01
AVOIDING_OBSTACLE	HIGH	3.8 to 10.0 km/h	8.0 km/h	-0.45 to +0.45
PEDESTRIAN_STOP	CRITICAL	0.0 km/h	0.0 km/h	+0.00

5.2 Discussion

5.2.1 How the proposed system addresses the gaps

Table 4 pairs each limitation identified in the literature with the design choice that responds to it.

Table 4: Existing limitations and proposed solutions

Existing limitation	**Proposed solution**
Perception and planning built separately	A single pipeline from camera detection to vehicle control
Emphasis on terrain or off-road driving	Planning that accounts for vehicles, pedestrians, animals and obstacles on public roads
Moving obstacles only partly handled	Motion prediction and collision-risk estimation feeding the planner
Reliance on lane markings or detailed maps	Free-space perception and Frenet-frame planning that work without markings
Limited replanning	A monitoring loop that feeds back to prediction and planning
Fixed cost functions	A cost function whose weights adapt to the situation
Paths that are valid but uncomfortable or infeasible	Joint position and speed optimization with curvature and jerk limits
Planning too slow for real time	Vectorized NumPy and SciPy evaluation of many paths
Studies limited to perception	Detection outputs used directly for planning and control
Few tests on unusual road events	CARLA scenarios with pedestrians, potholes, animals and uncooperative vehicles

5.2.2 Limitations

Detection gets worse in low light situations or situations of partial occlusion and a single camera provides little information about depth. Indian road users are hard to predict, and a predicted risk is an estimate, not a safety guarantee. The data collected by simulators is collected in an ideal environment and can never replicate everything that happens in reality; thus, successful simulation does not imply the same success in reality, as the situation will be different with respect to sensors and physics. This project is a software-based simulation project.

6. Conclusion and Future Work

We presented an intelligent path planning and collision avoidance system for the Indian roads.This system consisted of YOLO-based perception for detection and scene understanding, localization, map information, and motion prediction via Frenet-frame adaptive planning, plus a continual feedback loop system that recalculates the path as the environment changes or parameters change. A smooth, safe vehicle where no lane markings and traffic rules can be used to drive. Each aspect can be tested and improved in isolation, and an evaluation protocol with detection, planning, safety and control metrics has been defined.

In the next steps, the planned CARLA experiments will be run, and quantitative comparisons made with A*, Hybrid A* and conventional local planners. Furthermore, the system can incorporate real-time sensors such as Lidar and radar to account for the depth and occlusion limitations of monocular vision and facilitate ADAS, autonomous vehicles, smart transport, and intelligent traffic management.

7. References

[1] J. Fan et al., “One-stage motion planning with probabilistic traversability risk awareness for unmanned ground vehicles in unstructured environments via accelerated model predictive path integral control,” Green Energy and Intelligent Transportation, 2026, doi: 10.1016/j.geits.2026.100444.

[2] P. Chakraborty et al., “IndiSegNet: Real-time semantic segmentation for unstructured road scenes in intelligent transportation systems,” Intelligent Systems with Applications, vol. 29, Art. no. 200629, 2026, doi: 10.1016/j.iswa.2026.200629.

[3] K. K. Raju et al., “Cascade drive: A unified deep learning framework for multi-featured detection and control,” Scientific Reports, 2025, doi: 10.1038/s41598-025-06567-9.

[4] C. Man et al., “Path planning for tracked vehicles based on improved hybrid A* algorithm,” Transactions of Beijing Institute of Technology, 2025, doi: 10.15918/j.tbit1001-0645.2025.010.

[5] S. H. Lee et al., "Local minima prediction using dynamic Bayesian filtering for UGV navigation in unstructured environments," IFAC-PapersOnLine, 2025, doi: 10.1016/j.ifacol.2025.10.244.

[6] O. Jerome et al., "A real-time framework for intermediate map construction and kinematically feasible off-road planning without OSM," arXiv:2510.03948, 2025. [Online]. Available: https://arxiv.org/abs/2510.03948

[7] T. Zhu et al., "A dual-layer hybrid-A* path planning algorithm for unstructured environments based on phase windows," Sensors, vol. 26, no. 1, Art. no. 43, 2026, doi: 10.3390/s26010043.

[8] M. Reda et al., "Path planning algorithms in the autonomous driving system: A comprehensive review," Robotics and Autonomous Systems, vol. 174, Art. no. 104630, 2024, doi: 10.1016/j.robot.2024.104630.

[9] Q. Liu et al., "Unmanned vehicle off-road path-planning method with comprehensive constraints on multiple environmental factors," International Journal of Digital Earth, vol. 17, no. 1, Art. no. 2408453, 2024, doi: 10.1080/17538947.2024.2408453.

[10] J. Zhou et al., "A multi-level framework for collaborative vehicle trajectory planning in unstructured road environments," Transportmetrica B: Transport Dynamics, 2026, doi: 10.1080/21680566.2026.2659893.

[11] J. Fan et al., "Toward GPS-independent ground vehicle control: Visual navigation and traversability-aware planning in unstructured environments," Defence Technology, 2026, doi: 10.1016/j.dt.2026.04.016.

[12] J. Wang et al., "Monocular visual obstacle avoidance method for autonomous vehicles based on YOLOv5 in multi lane scenes," Alexandria Engineering Journal, vol. 109, pp. 497–507, 2024, doi: 10.1016/j.aej.2024.09.019.

[13] H. S. Gowri Yaamini et al., "Lane and traffic sign detection for autonomous vehicles: Addressing challenges on Indian road conditions," MethodsX, vol. 14, Art. no. 103178, 2025, doi: 10.1016/j.mex.2025.103178.

[14] M. Chaman et al., "Benchmarking YOLO-based deep learning models for real-time object detection in hybrid ADAS and intelligent transportation systems," Results in Engineering, vol. 29, Art. no. 108942, 2026, doi: 10.1016/j.rineng.2025.108942.

[15] B. Li et al., "An off-road path planning algorithm based on fine vegetation classification using remote sensing data," Geocarto International, 2026, doi: 10.1080/10106049.2026.2667676.

[16] N. P. Selvarajan et al., "Safety-aware transformer-enhanced unified multi-task perception for road anomaly detection, drivable area segmentation and lane estimation," Scientific Reports, 2026, doi: 10.1038/s41598-026-60419-8.

[17] L. Chen et al., "Scene as occupancy and reconstruction: A comprehensive dataset for unstructured scene understanding," Scientific Data, vol. 12, Art. no. 1232, 2025, doi: 10.1038/s41597-025-05532-5.

[18] M. Shi, B. Li, and W. Xu, "Unstructured road obstacle avoidance path planning algorithm based on BEV grid map," 2026, doi: 10.54097/0v1jzt98.

[19] X. Chen et al., "Drivable area recognition on unstructured roads for autonomous vehicles using an optimized bilateral neural network," Scientific Reports, 2025, doi: 10.1038/s41598-025-94655-1.

[20] C. Li et al., "Research on autonomous ground vehicle path planning in unstructured environments," IEEE Open Journal of Vehicular Technology, vol. 7, pp. 1–12, 2026, doi: 10.1109/OJVT.2026.11695100.

[21] M. Minařík, V. Vonásek, and R. Pěnička, "Enhancing sampling-based planning with a library of paths," Robotics and Autonomous Systems, vol. 198, Art. no. 105334, 2026, doi: 10.1016/j.robot.2026.105334.