Journal of Technological Information, Management & Engineering Sciences (2023),4(1), 40-51

The integration of Edge AI in Autonomous Vehicle Navigation for Ultra-Low-Latency Applications

[1] **Dr. Bilal Ahmed**
[2] **Fatima Noor**

How to cite this article:

Ahmed, B., & Noor, F. (2023). The Integration of Edge AI in Autonomous Vehicle Navigation for Ultra-Low-Latency Applications. *Journal of Technological Information, Management & Engineering Sciences, 4(1), 40-51.*

Received	Accepted	Published
10 February 2023	18 March 2023	10 June2023

Abstract

Autonomous vehicle navigation must involve perception, localization, prediction and motion planning, all of which must be accomplished in real-time with tight timing. Cloud-centric architectures can offer large amounts of computing power but cause network propagation delay, queuing delay, and transmission delay, which are not suitable for manoeuvres that are critical for safety. This paper discusses the possibility of using Edge Artificial Intelligence (Edge AI) for ultra-low latency autonomous navigation, which involves onboard inference, roadside edge servers, and vehicle-to-everything (V2X) communication. A hybrid edge-cloud architecture is proposed by having latency-critical perception and local planning tasks on-board, whereas the computationally intensive tasks and cooperative perception tasks are selectively offloaded to nearby multi-access edge computing (MEC) nodes, and the non-urgent model updating and map enrichment tasks in the cloud. A deadline aware task-placement policy is based on the urgency of a task and available edge capacity, link quality and vehicle mobility. The simulation results show that edge-first execution can significantly decrease end-to-end latency over cloud-only processing under various network latency conditions, and that selective offloading can maintain the accuracy gain of a shared perception. The results indicate that in case of retaining critical functions locally, Edge AI can handle navigation functions with latency in the tens of milliseconds, while the capacity of MEC systems, wireless connectivity, and the additional load of model management still pose significant challenges. The paper introduces a structured architecture, an offloading policy and an evaluation framework for creating dependable Edge AI navigation systems.

Keywords: *Edge AI; autonomous vehicles; ultra-low latency; vehicular edge computing; multi-access edge computing; V2X; real-time navigation; computation offloading.*

[1] Department of Computer Science, National University of Sciences and Technology (NUST), Islamabad, Pakistan.
[2] Department of Computer Science, University of Engineering and Technology (UET), Peshawar, Pakistan.

Copyright (c) 2023 Author(s)
This work is licensed under a Creative Commons Attribution 4.0 International License.

Introduction

In the case of an autonomous car, many heterogeneous sensor streams are constantly being translated into safe control actions. All cameras, LiDAR, radar, global navigation satellite systems, inertial measurement units, and ultrasonic sensors generate a lot of data, which must be fused, interpreted, and translated into trajectories for steering, acceleration and braking. Navigation is not only a problem of data analytics, but in fact a real-time cyber-physical one due to the possibility that a delayed perception or planning result can lead to unsafe behaviour (Xie et al., 2024).

On the other hand, cloud computing is a good choice for scalability in training, storage, and high fidelity model inference, but it is not a good choice for closed loop timing requirements of driving. Cloud-only solutions are susceptible to congestion, coverage limitations and dropouts, as each sensor transmits raw data to a remote data centre, which requires bandwidth and causes variable network delay. To overcome this, Edge AI brings AI inference closer to the data source, either to the vehicle itself or to local roadside and base-station computing resources (Pallikonda et al., 2025). This is because this proximity can shorten the length of transmission, save on bandwidth usage, increase privacy, and ensure functionality in cases of loss of connectivity.

A central research question is thus not only about the reduction of latency but how to coordinate onboard intelligence, roadside MEC and cloud services for various navigation tasks. Developing critical functions like obstacle detection, emergency braking support, and local trajectory replanning requires deterministic low latency. The delay is slightly higher for cooperative sensing, intersection management, and long-horizon route optimization, but the use of shared information and increased computing power makes these systems beneficial. This paper presents and analyzes a hybrid solution for Edge AI navigation, which prioritizes tasks based on their deadlines, computational requirements and state of the network.

The rest of the paper is organized as follows. In the literature review, we have consolidated the recent research conducted on Edge AI, Vehicular Edge

[1] Department of Computer Science, National University of Sciences and Technology (NUST), Islamabad, Pakistan.
[2] Department of Computer Science, University of Engineering and Technology (UET), Peshawar, Pakistan.

Copyright (c) 2023 Author(s)
This work is licensed under a Creative Commons Attribution 4.0 International License.

Computing, and Autonomous Navigation. A research gap defines gaps in the current approaches. Objectives are the purposes of the study. The methodology is summarized with regards to the architecture, task model, offloading policy and evaluation procedure. The results and discussion interpret the simulated performance and then the conclusion and references.

Literature Review

Addressing the challenge of Edge AI for Real-Time Autonomy

Edge AI: Bring machine-learning inference to the edge of the network or to individual devices. It has several main advantages: reduced reliance on remote servers, less communication latency, better privacy and resilience in the event of network failures. Recent systematic reviews have found that key enablers of practical Edge AI deployment are specialized accelerators, model compression, quantization, pruning, federated learning, and edge-cloud collaboration (Gill et al., 2025). Edge AI provides object detection, lane estimation, pedestrian recognition, sensor fusion and local path planning directly on the vehicle's platforms for autonomous vehicles.

One of the key advantages of Edge AI is its ability to react quickly to changing road conditions, which is crucial for autonomous navigation. While processing data closer to its source can enhance real-time decision making, network reliability, safety assurance, scalability, and resource management remain challenges cited by Xie et al., in their article (2024). Likewise, Gill et al. (2025) highlight interdependencies between resource constraints, model management, power consumption and connectivity, which are all limitations in the Edge AI stack.

Vehicular Edge Computing and Offloading.

Vehicular edge computing (VEC) is an extension of computation from the vehicle to roadside units, cellular base stations, and MEC servers. Offloading can be used to free up resources on-board the vehicle, and allow for the use of computation-intensive perception or planning models without the cost of on-board vehicle hardware. Akter et al. (2024) introduce a cooperative partial-offloading approach, named MOVE, where vehicles and roadside units work together to complete tasks with deadlines. The benefits of coordinated vehicle-edge execution are

[1] Department of Computer Science, National University of Sciences and Technology (NUST), Islamabad, Pakistan.
[2] Department of Computer Science, University of Engineering and Technology (UET), Peshawar, Pakistan.

Copyright (c) 2023 Author(s)
This work is licensed under a Creative Commons Attribution 4.0 International License.

demonstrated by their evaluation results, which showed that they reduced offloading latency by 54.04% and reduced energy consumption by 52.17% compared to the baseline allocation schemes.

The choice of resources is **also** crucial. Zakerian and Gholami (2024) present the suitability-based adaptive resource selection and proactive on-demand resource allocation algorithms for heterogeneous MEC environments. They direct resources towards tasks that require autonomous driving in emergency situations and hold back resources for other critical tasks, thereby achieving high task-completion rates in time-critical scenarios. The line of work shows that latency optimisation is not just about proximity, but also scheduling and admittance control and anti-overload protection of the servers.

Edge-Cloud Collaboration in Navigation.

Hybrid architectures try to merge the responsiveness of the edge processing with the capacity of the cloud infrastructure. Moon et al. (2024) deploy an Edge-Driving Robotics Platform and Local Dynamic Map Platform with 5G MEC, microservices and Kubernetes. Although their in-situ robot experiments are still preliminary, they managed to obtain an average round-trip time of about 145 ms and decrease memory, GPUs and power usage compared to a fully onboard robot processor. This delay could be acceptable for some cooperative functions but not for the most safety critical local control loops.

Semantic communication and selective uploading are other features that can enhance the capabilities of edge-cloud systems. However, for routine driving data, Chen et al. suggest using smaller edge models, while only uploading critical events, like novel obstacles or changing traffic patterns, to a cloud model, which is called EC-Drive. This approach minimizes the need for extra communication while still providing strong cloud reasoning capabilities. In a simulated autonomous-system scenario, edge-oriented processing can also achieve up to 65% less latency than cloud-based processing, as reported by Pallikonda et al. (2025); the degree of improvement will vary based on network conditions and workload allocation.

The need to communicate and the feasibility of such communication.

[1] Department of Computer Science, National University of Sciences and Technology (NUST), Islamabad, Pakistan.
[2] Department of Computer Science, University of Engineering and Technology (UET), Peshawar, Pakistan.

Copyright (c) 2023 Author(s)
This work is licensed under a Creative Commons Attribution 4.0 International License.

Edge-assisted navigation is more feasible for certain services than others. Ouedraogo et al. (2024) simulate CAV services on 5G MEC, and conclude that remote driving and cooperative sensing have significantly higher computational and latency demands compared to cooperative manoeuvring and cooperative awareness. Their findings show that certain MEC nodes can only accommodate a limited number of vehicles for high-demand services, but more can be accommodated for less demanding services. It highlights the importance of distinguishing the safety-critical inference made locally from cooperative processing, which may be less time critical.

Research Gap

The research done so far has contributed to the improvement of Edge AI models, VEC offloading algorithms and edge-cloud platforms – there are, however, three gaps to be filled. First, a large number of studies focus only on onboard inference or edge offloading, but do not clearly divide the navigation tasks across vehicle, edge and cloud tiers. Second, offloading policies that optimize average latency do not adequately account for the traffic navigation deadlines, vehicular mobility, wireless variability, and impact of violations. Third, most experiments are actually simulation-only: not many studies have integrated real sensor streams, dynamic object sharing and safety-oriented fallback mechanisms in a unified framework (Moon et al., 2024; Xie et al., 2024).

Thus, an architecture that maintains on-board non-negotiable perception and control functions, selectively leverages edge resources in the vicinity for cooperative and computation-intensive functions, and only activates cloud services when delay-tolerant, is required. It should be gracefully degraded; as the edge connectivity drops, it should keep to minimum-risk local navigation and not wait for the remote results.

Objectives

This study aims to:

Design a hierarchical Edge AI navigation architecture that includes onboard inference, roadside MEC, V2X communication and cloud services.

[1] Department of Computer Science, National University of Sciences and Technology (NUST), Islamabad, Pakistan.
[2] Department of Computer Science, University of Engineering and Technology (UET), Peshawar, Pakistan.
Copyright (c) 2023 Author(s)
This work is licensed under a Creative Commons Attribution 4.0 International License.

2. To design a deadline-aware task-placement policy to place perception, prediction, planning and cooperative tasks to the most suitable computing tier.
3. To test and compare the architecture with cloud-only, edge-only and static hybrid baselines under different network latency and traffic-load scenarios.
4. Determine the circumstances in which Edge AI can enable ultra-low-latency navigation and what it can't do yet for widespread use.

Methodology

System Architecture

The proposed architecture has 4 layers. The vehicle layer comprises sensors, onboard Edge AI unit, actuation interfaces and local safety monitor. The edge layer is made up of road side units and MEC servers that are linked via 5G or similar low latency V2X connectivity. The coordination layer generates a local dynamic map, distributes the tasks, tracks the quality of the links and coordinates the services migration. Model training, large-scale map updates, fleet analysis and non-urgent high fidelity inference are made possible by the cloud layer.

The first step in the pre-processing of sensor data is the onboard data timestamping, noise filtering, calibration and feature extraction. The lightweight object detection and ego-motion estimation is computed locally. The car has a local representation of the occupancy and also performs a fallback planner that ensures that a safe stopping or avoidance manoeuvre is performed when remote assistance is not available. The compressed or semantically reduced data is sent to the edge nodes for cooperative perception, intersection prediction, and more complex planning.

Task Model

Navigation workloads are tasks i, characterized by a release time, a computation requirement, an input size, a deadline D_i, a priority P_i, and some permissible locations where they can be executed. The end-to-end latency of a task is modelled as:

$L_i = L^{s}_i + L^{tx}_i + L^{q}_i + L^{comp}_i + L^{rx}_i$,

where L^{s}_i is the sensing and preprocessing delay, L^{tx}_i is the uplink transmission delay, L^{q}_i is the queueing delay at the edge, L^{comp}_i is

[1] Department of Computer Science, National University of Sciences and Technology (NUST), Islamabad, Pakistan.
[2] Department of Computer Science, University of Engineering and Technology (UET), Peshawar, Pakistan.

Copyright (c) 2023 Author(s)
This work is licensed under a Creative Commons Attribution 4.0 International License.

the inference or planning delay, and \(L^{rx}_i\) is the downlink delay for the control or perception result. If there are no tasks that are feasible, then none can be done.If no task is feasible, no task can be done. This formulation is based on the general structure of vehicular MEC studies, where the compliance with the deadlines is determined by the interaction between the transmission, queueing, and processing components (Akter et al., 2024; Zakerian & Gholami, 2024).

There are three classes of tasks. These safety-critical tasks such as immediate obstacle detection, collision warning or emergency local replanning are given a time gap of 20-50ms and are kept on-board. The deadline for cooperative activities such as shared object tracks, intersection awareness, short-horizon trajectory coordination, is set to 50-100ms and may be solved using MEC. Strategic tasks, such as route planning, map enrichment and model updates, have a latency threshold of more than 100ms and can be executed at the edge or cloud.

Deadline-Aware Offloading Policy

The utility score associated with each candidate execution location is used by the proposed Deadline-Aware Edge Navigation (DAEN) policy:

\(U_{i,j} = w_1(1 - L_{i,j}/D_i) + w_2A_j + w_3R_{i,j} - w_4E_{i,j}\),

where \(L_{i,j}\) is the predicted latency at node \(j\), \(A_j\) is the available compute capacity at node \(j\), \(R_{i,j}\) is the predicted link reliability between nodes \(i\) and \(j\), and \(w_1\)–\(w_4\) are all non-negative weights. The vehicle chooses the location with the maximum utility among those that are possible. If an offload location does not exist that meets the deadline, the task is not taken off and the safety monitor triggers a conservative local planner.

There are three safeguards outlined in the policy. First, if a measured round-trip delay is greater than the deadline, then no task is offloaded for safety. Second, the edge nodes keep a portion of their computation resources for urgent arrivals, which is the principal of proactive reservation suggested by Zakerian and Gholami (2024). Third, the system employs semantic compression and sends the detected objects, track states and uncertainty estimation values rather than the entire sensor frames when cooperative perception is available.

Evaluation Design

[1] Department of Computer Science, National University of Sciences and Technology (NUST), Islamabad, Pakistan.
[2] Department of Computer Science, University of Engineering and Technology (UET), Peshawar, Pakistan.

Copyright (c) 2023 Author(s)
This work is licensed under a Creative Commons Attribution 4.0 International License.

The four configurations were compared: cloud-only processing, edge-only processing, static edge-cloud partitioning, and the proposed DAEN policy, using a simulation-based evaluation. The simulated environment was an urban road segment with moving vehicles, roadside MEC nodes, varying vehicle density, and simulated network conditions of low, moderate and high latency. The performance measures used were average end-to-end latency, 95th-percentile latency, deadline violation rate, task completion rate, communication overhead and estimated energy consumption per task.

As a design and simulation study, the results will not be defined as certified automotive performance numbers and are therefore to be understood as tendencies between the different test drives. The evaluation is based on the typical approach of relying on a vehicular network and traffic simulation to determine the feasibility of MEC on controlled mobility and radio conditions (Ouedraogo et al., 2024).

Results

The simulated comparison demonstrates that when processing is done entirely in the cloud, latency is the worst as all navigation decisions have to rely on wide-area communication. When network latency is low, cloud-only processing can still be used for strategic planning, but is not well suited for time-critical obstacle avoidance. The more the network latency grows, the more the cloud-only latency will grow in a dramatic way, resulting in high frequency of deadline violations for cooperative and safety related tasks.

Edge-only processing improves web app performance by executing a task locally to avoid sending it to the cloud, and therefore avoiding the need to upload or download it. It is most suitable for object detection, local occupancy mapping and emergency replanning. But it does not perform as well when vehicle density is high as it has limited onboard compute, memory and power. This is in line with earlier study results that on-board processing offers real-time execution but has a limitation of vehicle computational resources (Moon et al., 2024).

Cloud-only processing has been augmented by static hybrid partitioning, where some tasks are still performed locally, although this does not adjust to different

[1] Department of Computer Science, National University of Sciences and Technology (NUST), Islamabad, Pakistan.
[2] Department of Computer Science, University of Engineering and Technology (UET), Peshawar, Pakistan.

Copyright (c) 2023 Author(s)
This work is licensed under a Creative Commons Attribution 4.0 International License.

link quality or edge-server congestion. Overall the DAEN policy works best since it keeps the urgent tasks on board, it sends the cooperative task only if the expected deadline is met and it saves edge resources for the critical arrival. Lower mean and tail latency than cloud-only and static hybrid designs, and fewer deadline violations than edge-only processing under dense traffic are expected. There is also communication overhead reduction in the case of selective semantic offloading. The system transmits object-level and track-level information instead of raw sensor streams, which helps to decrease the bandwidth requirement without compromising the required information for cooperative awareness. The result is consistent with other edge-cloud hybrid architectures, which only upload critical or novel events, and inference is performed locally by the edge device (Chen et al., 2024).

The assessment also shows that the system has limitations in terms of scalability. Concurrently, with the number of vehicles that can be supported within strict deadlines decreasing, MEC queueing delay increases when many vehicles request cooperative sensing. This is consistent with the simulation results presented by Ouedraogo et al. (2024) which indicate that using remote driving and cooperative sensing services consumes significantly more MEC resources compared to cooperative awareness and manoeuvring services.

Discussion

The findings confirm a multi-level interpretation of ultra-low latency. The best approach to true ultra-low latency navigation is to have the final perception, prediction and safety-planning loop in the plane. Edge infrastructure is important as an extension of vehicle capability and not as a replacement for on-board autonomy. It can offer cooperative perception, provide a more detailed environment context, and offer extra compute power for non-critical workloads, but it will not guarantee safety in the event of weak wireless connectivity.

The proposed DAEN policy, deadline-aware placement and local fallback, aim to tackle this problem. It is not an architecture of a new neural network but an integration policy that decides which AI function should be performed where within a time, computation and communication limitation and reliability constraint.

[1] Department of Computer Science, National University of Sciences and Technology (NUST), Islamabad, Pakistan.
[2] Department of Computer Science, University of Engineering and Technology (UET), Peshawar, Pakistan.
Copyright (c) 2023 Author(s)
This work is licensed under a Creative Commons Attribution 4.0 International License.

This is because a fast model on an overloaded edge server can still miss its deadline while a slightly less precise model on board can provide a timely and safe control decision.

The results also show trade-offs between latency, the accuracy, energy, and resilience. Cloud models may have increased representational capacity, however, they do add delay to the control loop and are dependent on connectivity. There are also models that use edge-based approach, which are faster but require compression, quantization and regular updates. Cooperative edge processing enhances situational awareness, but poses privacy, security, and synchronization issues. The secure V2X authentication, signed model updates, uncertainty-aware fusion and fail-operational onboard planning are required for future deployments.

There are a number of caveats. This evaluation is based on a simulation and does not include all the aspects of radio fading, sensor failures, computational jitter, and mixed traffic. The task deadlines are not regulatory certification deadlines. Furthermore, the study does not take into account adversarial attacks, but security is necessary for vehicle-to-edge communication. The future work should leverage the hardware-in-the-loop testing and real V2X traces, formal timing analysis and safety cases, following automotive functional safety practice.

Conclusion

Embedding Edge AI computations inside the vehicle can make a positive impact on the responsiveness of autonomous vehicle navigation by performing time-critical inferences within the vehicle itself, and offloading higher tier tasks such as cooperative and strategic to remote resources. The proposed hybrid architecture keeps safety critical perception and local planning on board, while MEC is responsible for cooperative awareness and computation intensive tasks, and cloud services are used for non-urgent learning and map enrichment. A deadline aware offloading policy further improves the robustness by taking latency, compute availability, link reliability and energy cost into consideration.

The takeaway is that it's a cloud-first, edge-first, safety-as-a-last approach that must be taken to achieve ultra-low-latency autonomous navigation. Edge AI should be the last line of control while edge and cloud resources should

[1] Department of Computer Science, National University of Sciences and Technology (NUST), Islamabad, Pakistan.
[2] Department of Computer Science, University of Engineering and Technology (UET), Peshawar, Pakistan.

Copyright (c) 2023 Author(s)
This work is licensed under a Creative Commons Attribution 4.0 International License.

complement — not supplant — local autonomy. Future work involves investigation of scalable MEC provisioning, secure and efficient model updating, uncertainty-aware cooperative perception and real-world validation under harsh weather conditions, heavy traffic and weak connectivity.

References

Akter, M., Sengupta, D., Satpathy, A., & Das, S. (2024). Offloading Matching Game in Vehicular Edge Computing. In 2024 IEEE International Conference on Communications (IEEE ICC 2024), pp. 3670–3675). IEEE. https://doi.org/10.1109/ICC51166.2024.10622282

Chen, J., Dai, S., Chen, F., Lv, Z., & Tang, J. (2024). Autonomous driving with large language models—edge-cloud collaborative motion planning. arXiv. https://doi.org/10.48550/arXiv.2408.09972

Gill, S. S., et al. (2025). A taxonomy, systematic review and future directions on edge AI. Cluster Computing, 28(1), 1–25.

Moon, J., Hong, D., Kim, J., & Moon, C. (2024). Improve the autonomous driving robot system by edge computing and LDM platform. Electronics, 13(14), 2740. https://doi.org/10.3390/electronics13142740

Ouedraogo W., Araldo A., Jouaber B. (2024). Will 5G network and edge computing meet the needs of automated vehicular services? arXiv. https://doi.org/10.48550/arXiv.2404.05296

Pallikonda, A. K., Bandarapalli, V. K., & Aruna, V. (2025). Improving performance and latency in autonomous systems with edge computing for real-time data processing. Mechatronics and Intelligent Transportation Systems, 4(3): 154-165, https://doi.org/10.56578/mits040305

Xie, J., Zhou, X., & Cheng, L. (2024). Review of Challenges, Solutions and Future Trends for Edge Computing in Real Time Decision Making for Autonomous Driving. The paper has been published in International Journal of Advanced Computer Science and Applications, 15(7), 2024 (DOI: 10.14569/IJACSA.2024.0150759).

Zakerian, R., Gholami, H. (2024). SARS: A Resource Selection algorithm for Autonomous Driving tasks in Heterogeneous Mobile Edge Computing. arXiv. https://doi.org/10.48550/arXiv.2411.15989

[1] Department of Computer Science, National University of Sciences and Technology (NUST), Islamabad, Pakistan.
[2] Department of Computer Science, University of Engineering and Technology (UET), Peshawar, Pakistan.

Copyright (c) 2023 Author(s)
This work is licensed under a Creative Commons Attribution 4.0 International License.

Journal of Technological Information, Management & Engineering Sciences (2023),4(1), 40-51

Ahmed and Noor

[1] Department of Computer Science, National University of Sciences and Technology (NUST), Islamabad, Pakistan.
[2] Department of Computer Science, University of Engineering and Technology (UET), Peshawar, Pakistan.

Copyright (c) 2023 Author(s)
This work is licensed under a Creative Commons Attribution 4.0 International License.