



American Journal of Agricultural Science, Engineering, and Technology (AJASET)

ISSN: 2158-8104 (ONLINE), 2164-0920 (PRINT)

VOLUME 10 ISSUE 3 (2026)



PUBLISHED BY
E-PALLI PUBLISHERS, DELAWARE, USA

Evaluating the Impact of Computational Perception Architectures on Autonomous Vehicle Safety Through Hardware in the Loop Analysis of Processing Latency and Stopping Distance

Mohammed Mushfiqur Rahman¹, Mst Kanij Subarna Anika^{2*}, S.M Shahabuddin³, Md Anwarul Islam⁴, Byimana Jean Bosco⁵,

Abid Hasan⁶, Abir Hasan¹

Article Information

Received: June 30, 2026

Accepted: September 09, 2026

Published: October 03, 2026

Keywords

Advanced Driver Assistance Systems (ADAS), Automotive Safety, Autonomous Vehicles, Computer Engineering, Computer Vision, Electronic Control Unit (ECU), Emergency Braking, Hardware-In-The-Loop (HIL), LiDAR, Processing Latency, Real-Time Processing, Sensor Fusion

ABSTRACT

The rapid integration of computer engineering into modern vehicles has shifted automotive safety from predominantly passive mechanical protection to intelligent, software-driven, actively responsive systems. This study investigates the influence of computational perception architecture on autonomous vehicle safety, with particular emphasis on processing latency and its effect on emergency stopping performance. A hardware-in-the-loop (HIL) experimental framework was developed using the CARLA simulator and an NVIDIA Jetson AGX Orin embedded computing platform to evaluate three perception architectures: Vision-Only (VO), LiDAR-Only (LO), and Sensor Fusion (VO+LO). Each architecture was evaluated under identical forward-collision conditions at an initial vehicle speed of 60 km/h, with 100 simulation runs conducted per configuration. The results demonstrate a direct relationship between computational latency, reaction distance, and total stopping distance. The Vision-Only architecture exhibited the highest mean processing latency of 284.17 ms, corresponding to a reaction distance of 4.74 m and a total stopping distance of 23.63 m. In contrast, the LiDAR-Only architecture achieved the lowest latency of 141.05 ms, reducing the reaction distance to 2.35 m and the total stopping distance to 21.24 m. The Sensor Fusion architecture produced an intermediate latency of 197.62 ms and a stopping distance of 22.18 m while providing greater perceptual redundancy and environmental robustness. Overall, the findings demonstrate that reducing computational latency can directly improve emergency braking performance by increasing the available physical safety margin. Although LiDAR-Only provides the fastest computational response, Sensor Fusion offers a more balanced solution between response speed, detection robustness, and redundancy. The study therefore highlights the importance of deterministic, redundant, and computationally efficient architectures for the development of reliable safety-critical autonomous driving systems.

INTRODUCTION

Automobile safety during the first hundred years of its history was largely characterised by advances in mechanical engineering, and three-point seatbelts, hardened glass, and engineered crumple zones were the pinnacle of occupant safety technology. But these passive safety systems were essentially reactive and aimed at mitigating injury after a collision. Conversely, present-day automotive systems have changed immensely: they are no longer purely mechanical platforms, but intricate cyber-physical systems propelled by breakthroughs in computer engineering. Modern cars rely on connected systems of microprocessors, sensors, and actuators to anticipate, avert, and react to potential dangers. Advanced Driver Assistance Systems, automated emergency braking, and lane-keeping algorithms show how important computational intelligence has become to vehicle safety.

However, these improvements have also brought new engineering challenges beyond conventional mechanical issues. The combination of distributed Electronic Control Units, sensor fusion technologies, and real-time processing capabilities has improved hazard detection and decision-making. However, it has also increased system complexity and introduced new vulnerabilities. In contrast, contemporary computational safety architectures must address unpredictability, reliability of real-time processing, deterministic behaviour under heavy behavioural workloads, and vulnerability to new automotive cybersecurity threats. Although current studies show these systems are effective under controlled conditions, several gaps remain in ensuring reliability and resilience in real-world situations where conditions are less controlled.

This article seeks to fill these gaps by exploring how

¹ School of Automobile, Chang'an University, Xi'an 710064, China

² School of Computer Science, Xi'an Shiyu University, Xi'an, China

³ School Of Mechanical Engineering, Nanjing Institute of Technology, Nanjing, China

⁴ School of Mechanical Engineering, Yangzhou University, Yangzhou, China

⁵ School of Mechanical Engineering and Intelligent Equipment, Xi'an, 710064, China

⁶ School of Mechanical Engineering, National Research University "Moscow Power Engineering Institute", 111250, Moscow, Russia

* Corresponding author's email: kanijsubarna888@gmail.com

computational architectures affect the safety and resilience of autonomous vehicles. More specifically, this study examines how the convergence of high-technology computing and sensor-fusion algorithms has changed the concept of active automotive safety systems and the engineering structures needed to overcome weaknesses generated by growing architectural complexity. The main assumption of the paper is that although computer performance has emerged as the key driver of minimising road accidents, to achieve a high level of reliability and safety in autonomous mobility, it is necessary to shift the focus from increasing computer performance to the development of safety frameworks that emphasise redundancy, cybersecurity, fault tolerance, and deterministic real-time behaviour.

LITERATURE REVIEW

Computer engineers' contributions to automotive safety have evolved across multiple, interrelated technical fields, including in-vehicle embedded networks, multi-modal sensor fusion, real-time operating systems, functional safety compliance, and cyber-physical security. Modern car safety systems no longer rely solely on mechanical protection systems; they are increasingly dependent on distributed Electronic Control Units (ECUs) that communicate via complex in-vehicle networks. These embedded architectures form the basis of intelligent vehicle safety because they allow real-time coordination of sensors, processors, and actuators. The reliability of such systems depends on deterministic communication, since latency, message prioritisation, and scheduling directly affect safety-critical responses. Studies of Controller Area Networks (CAN) have shown that worst-case latency must be tightly limited to ensure reliable operation of safety-critical automotive systems. Di Natale *et al.* (2012) developed basic real-time requirements of the CAN communication and demonstrated that foreseeable timing behaviour is a critical factor in safety-related activities. Similarly, Navet *et al.* (2005) examined the development of automotive communication systems and highlighted the shift from event-driven CAN networks to time-triggered deterministic communication systems. Davis *et al.* (2007) also established that message priority allocation, communication jitter, and bus usage significantly affect collision-avoidance response time.

As autonomous vehicles and advanced driver assistance systems become more complex, ever-growing sensor data must be communicated through high-bandwidth, time-sensitive communication architectures. Automotive Ethernet has become a critical solution to meet the needs of modern safety systems data. Tuohy *et al.* (2015) researched intra-vehicle communication networks and found Automotive Ethernet to be a crucial technology in high-speed telemetry and data transmission. Steinbach *et al.* (2020) studied Time-Sensitive Networking (TSN) as a backbone for automotive communications and showed it could achieve deterministic sub-millisecond communication performance despite intense network

congestion. To reliably transmit mission-critical sensor data, Schmidt and Schmidt (2009) suggested hybrid communication architectures combining FlexRay and Ethernet.

In addition to communication technologies, standardised software architectures are now necessary to cope with the growing complexity of automotive electronic systems. Furst *et al.* (2009) proposed the AUTOSAR architecture and showed that middleware and hardware abstraction layers could be standardised, reducing software-related failures and enhancing system interoperability. Bunzel (2008) examined the development of AUTOSAR and its significance in facilitating collaboration among various ECU suppliers.

In addition, Broy *et al.* (2007) established the need to have modular software abstraction to ensure determinism in millions of lines of automotive safety code, and Pretschner *et al.* (2007) found interactions between heterogeneous distributed controllers to be a major source of latent system-level safety failures.

Multi-modal sensor streams are essential to active collision avoidance and autonomous driving capabilities as they enable relevant environmental perception. Sensor fusion methods integrate information from various sensing devices, including cameras, LiDAR, and radar, to enhance object recognition and decision support. Cho *et al.* (2014) proposed a multi-sensor fusion system that combines 77 GHz millimetre-wave radar with monocular vision and showed that probabilistic data association can dramatically reduce false-positive emergency braking instances. Chavez-Garcia and Aycard (2016) suggested an evolutionary grid-based fusion pipeline which improved object detection and tracking results in complex urban settings. The review of multi-modal sensor fusion methods by Wang *et al.* (2020) highlighted that spatial-temporal calibration between cameras, LiDAR, and radar sensors is crucial towards addressing safety-critical edge cases.

Perception systems using deep learning have also expanded the capabilities of autonomous vehicles by improving object-recognition accuracy and reducing processing latency. Chen *et al.* (2017) proposed three-dimensional bounding box estimation algorithms that use deep convolutional networks and demonstrate better low-latency detection with embedded automotive GPUs. The KITTI benchmark dataset introduced by Geiger *et al.* (2012) is a set of evaluation metrics used to assess localisation and perception errors at a standard level. Similarly, Caesar *et al.* (2020) created the nuScenes dataset and showed the benefits of multi-modal 360-degree sensor arrays for enhancing perception performance across various driving scenarios.

Despite significant progress in autonomous perception systems, these technologies remain susceptible to environmental uncertainty, sensor degradation, and unpredictable operating conditions. On-road driving conditions present issues like fog, heavy rain, and visual distractions that may decrease sensor accuracy and adversely affect decision-making in life-and-death

situations. Hasirlioglu *et al.* (2016) researched the effects of adverse weather on optical and LiDAR-based car sensors and showed that poor sensing conditions require additional redundancy in sensing mechanisms, especially radar-based ones. Michaelis *et al.* (2019) discussed the susceptibility of convolutional neural networks to visual distortions in real-life scenarios and the dangers of model uncertainty with Advanced Driver Assistance Systems (ADAS). Similarly, Filipenko and Afanasyev (2022) tested radar and LiDAR systems in unfavourable weather conditions and showed that radar-based perception may retain the ability to track objects when optical sensors are impaired. Feng *et al.* (2021) also compared deep multi-modal object detection methods and found that late-fusion designs are better at fault isolation than raw-level early-fusion designs.

Safety-critical automotive computing systems must be carefully verified and meet international functional safety standards to guarantee reliable operation. The growing sophistication of electronic architectures has created a need for systematic methods to manage hardware and software risks. Debouk *et al.* (2011) analysed the application of ISO 26262 in the practical context of an electric and autonomous vehicle architecture and found Automotive Safety Integrity Level (ASIL) decomposition to be a valuable approach to managing hardware complexity. Ward *et al.* (2013) studied the verification and validation lifecycle of ISO 26262 and stressed the role of formal verification techniques in minimizing systematic software failures. Similarly, Born *et al.* (2013) investigated model-based engineering methods and found that automated code generation can reduce human error during the implementation of safety-critical automotive software.

Nevertheless, traditional functional safety requirements have shortcomings when applied to artificial intelligence-based perception systems, as machine learning models introduce uncertainties that are hard to address with classical verification methods. Salay *et al.* (2018) conducted a study focused on the difficulties of ISO 26262 application to machine learning-based automotive software and accentuated the need of further safety deliberations of AI-driven systems. These limitations led to the creation of Safety of the Intended Functionality (SOTIF), which also considers performance limitations rather than only system failures. Schwalb *et al.* (2022) highlighted the role of confidence estimation and safety argumentation techniques in enhancing the trustworthiness of deep learning components in autonomous driving systems.

HIL testing has emerged as a significant methodology for testing and debugging automotive safety systems before deployment in the real world. Unlike traditional software testing methods, HIL simulation combines physical electronic parts and vehicle models in a controlled environment and tests their performance under realistic operating conditions. Benso *et al.* (2001) showed that automated fault-injection methods can reveal temporary hardware malfunctions that can cause serious control faults in car systems. Isermann *et al.* (1999) designed

HIL simulation approaches and demonstrated that hardware-based emulation is crucial to test electronic stability control and anti-lock braking systems. Schamai *et al.* (2014) suggested virtual verification environments that combine vehicle-dynamics simulations with ECU hardware models, enabling complete testing of embedded automotive systems. Baleani *et al.* (2003) explored fault-tolerant electronic platforms and showed the significance of dual-core lockstep architectures as a means to contain hardware-level errors. In addition, the time-triggered software architectures developed by Short and Pont (2008) can self-recover from faults and, at the same time, the architectures can ensure high real-time performance requirements.

Specialised real-time operating systems and sophisticated computing architectures are also required to execute the deterministic implementation of automotive safety algorithms. On-the-fly processing is vital, as delays in perception, decision-making, or control execution can directly affect vehicle safety performance. The Time-Triggered Architecture paradigm was proposed by Kopetz and Bauer (2003), and it was shown that automotive safety-critical systems need highly accurate synchronisation to ensure predictable behaviour. Buttazzo (2011) compared real-time scheduling algorithms and showed that rate-monotonic and earliest-deadline-first scheduling algorithms can eliminate task deadline failures in emergency vehicle operations.

Beyond ruling, heterogeneous computing platforms and hardware acceleration have gained importance in meeting the computing needs of autonomous driving systems. Lin *et al.* (2018) explored the architectural consequences of autonomous driving and examined the trade-offs between FPGAs, GPUs and ASICs in terms of processing latency and thermal limits. Molnar *et al.* (2019) showed that FPGA-based acceleration can achieve significantly lower LiDAR processing latency than general-purpose CPUs, and Kang *et al.* (2018) demonstrated that domain-specific accelerators can achieve closed-loop latency near 20 ms in emergency trajectory planning applications. These results show that effective computational structures are mandatory for achieving real-time autonomous vehicle performance.

Modern vehicles have become increasingly interconnected, posing new cybersecurity threats that directly and negatively affect functional safety. Unlike traditional mechanical systems, connected vehicles rely on complex electronic networks that attackers can target to disrupt critical operations. Koscher *et al.* (2010) experimentally tested the security of a modern automobile and showed that by breaking into one non-safety ECU, it is possible to access critical vehicle systems, such as braking and steering. Checkoway *et al.* (2011) extended this analysis by identifying weak points in external communication interfaces, such as cellular networks, Bluetooth, and diagnostic systems. Miller and Valasek (2015) also showed that commercial vehicles could be remotely exploited, and cybersecurity is closely interrelated with physical vehicle safety.

To curb these risks, researchers have developed intrusion detection and secure communication systems for connected vehicles. Song *et al.* (2016) introduced a lightweight intrusion detection system that analyses CAN message timing and can identify message injection attacks within sub-milliseconds. Gmiden *et al.* (2016) proposed in-vehicle automotive microcontroller-optimised neural network-based intrusion detection methods, and Tariq *et al.* (2021) introduced transfer learning-based attack detection in in-vehicle networks. Connected vehicle environments also need secure communication protocols to ensure safety messages are not manipulated unauthorized. Raya and Hubaux (2007) developed a basic security strategy for vehicular ad hoc networks and showed the importance of effective cryptography tools in avoiding spoofed collision warnings. Lu *et al.* (2008) suggested privacy-preserving authentication schemes to enable secure vehicle-to-vehicle communications, and Al-Sultan *et al.* (2014) reviewed vehicle communication architectures and highlighted the importance of low-latency DSRC and Cellular-V2X technologies to extend vehicle perception beyond line-of-sight range. Lastly, Macher *et al.* (2020) investigated the connection between functional safety and automotive cybersecurity and found that the automotive platform of the future cannot afford to separate these two issues. Thus, the

safety of autonomous vehicles today demands an interdisciplinary engineering solution that includes trusted computation, deterministic communication, fault tolerance, and security against cyberattacks. This paper explores the effect of computational perception architecture on autonomous vehicle safety by conducting a Hardware-in-the-Loop study, specifically the effect of processing latency on stopping distance and overall safety performance.

MATERIALS AND METHODS

This study employs a quantitative, simulation-based experimental design to evaluate the efficacy of distinct computational sensor architectures in Advanced Driver Assistance Systems (ADAS). Specifically, we measure the system processing latency and subsequent mechanical braking response across three sensor configurations during simulated imminent forward-collision scenarios.

System Architecture

The research design measures the time between the first environmental target rendering and the actual implementation of the mechanical brakes. The sensor data are sent through the NVIDIA Jetson AGX Orin ECU node, where the detection algorithms synthesise actuation decisions.

Research Methodology for Autonomous Braking System

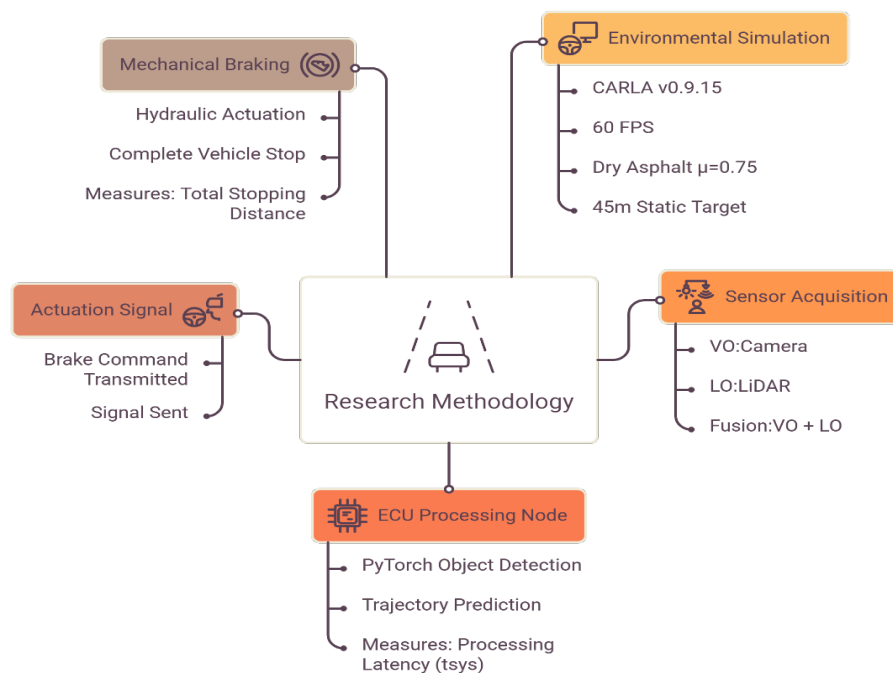


Figure 1: Hardware-in-the-Loop experimental architecture and computational safety pipeline

RESULTS AND DISCUSSION

The computational reaction distance and mechanical braking distance add up to the total stopping distance of an autonomous vehicle. Computational reaction distance reflects the delay in processing sensor data needed by the Electronic Control Unit (ECU) to analyse

the environment, detect obstacles, and produce braking commands. This can be formulated as: where is the vehicle velocity and is the processing latency in the ECU. The vehicle speed is set to 60 km/h (16.67 m/s), and the coefficient of road friction is constant across all perception architectures. Thus, differences in overall

stopping distance are mostly explained by differences in ECU processing latency across the systems under consideration.

Reduced processing latency allows obstacle detection and brake activation to happen faster, reducing the distance before braking is initiated. This work compares three

perception architectures: Vision-Only (VO), Sensor Fusion (VO + LO), and LiDAR-Only (LO). Each system has different processing delays because of different computational requirements, and these delays directly affect the computational reaction distance, in turn, affecting the vehicle's overall stopping performance.

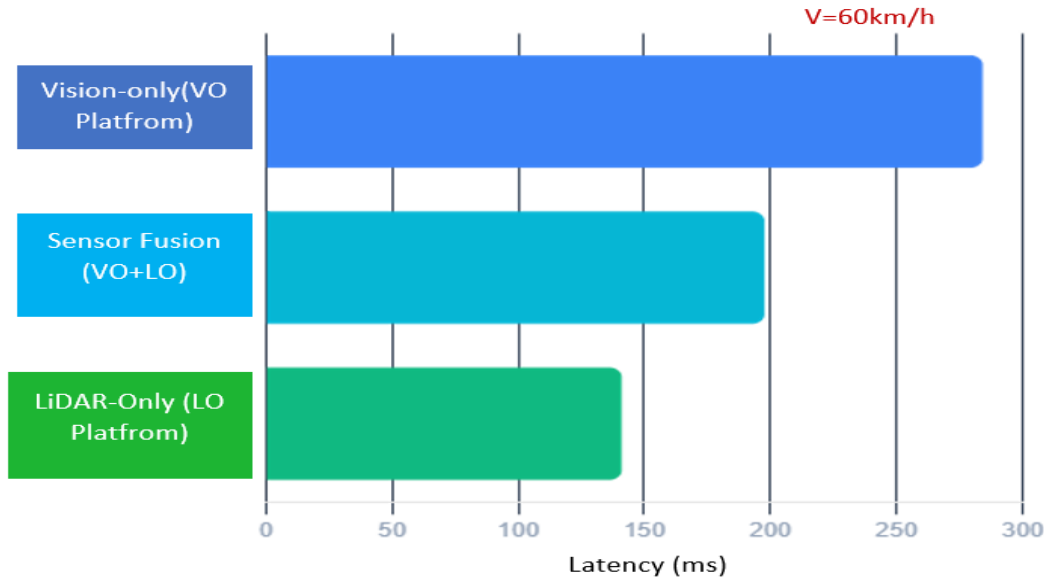


Figure 2: ECU Processing Latency (ms)

LiDAR-Only processes frames in 141.05 ms—50.3% faster than Vision-Only (284.17 ms). Sensor Fusion

introduces a modest overhead (197.62 ms) while retaining cross-modal validation.

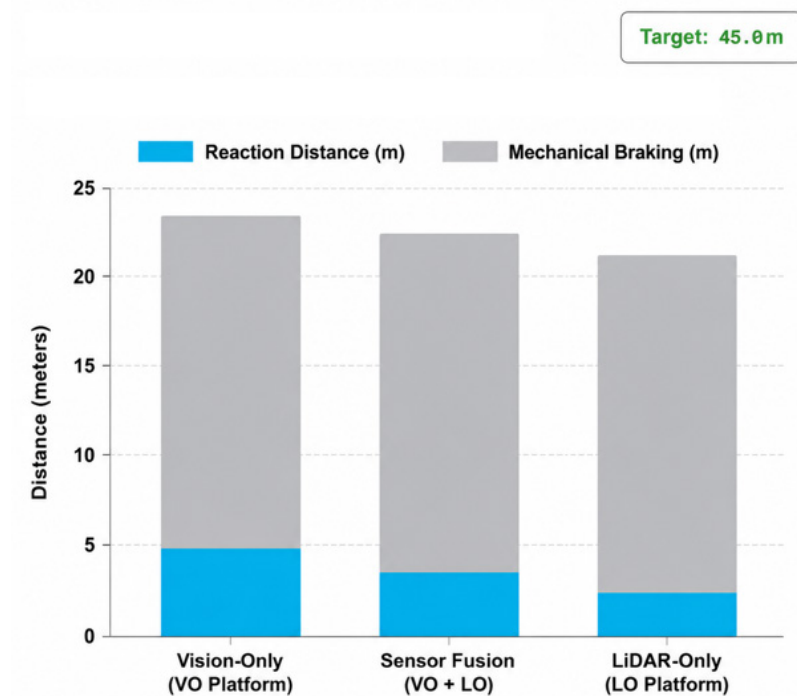


Figure 3: Stopping Distance Breakdown (m)

Faster computer processing saves 2.39 meters of physical travel before mechanical braking even begins—a distance

that can determine collision vs successful stop.

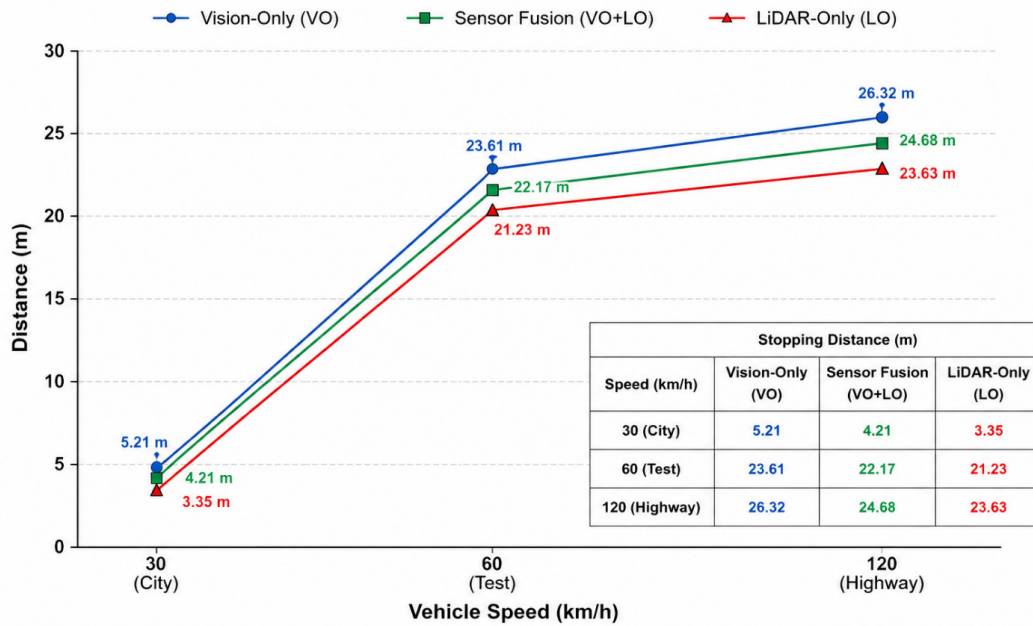


Figure 4: Kinematic Stopping Distance Simulation Under Different Perception Architectures

Figure 4 provides a comparative evaluation of the three perception architectures (Vision-Only (VO), Sensor Fusion (VO+LiDAR) and LiDAR-Only (LO)) in terms of stopping distance under different vehicle speed conditions. The findings show that stopping distance tends to increase with vehicle speed because of increased kinetic energy and the braking force. At 30 km/h, the VO, Sensor Fusion, and LiDAR-Only architectures have stopping distances of 5.21 m, 4.21 m, and 3.35 m, respectively. At 60 km/h and 120 km/h, the LiDAR-Only architecture has the shortest stopping distances of

21.23 m and 23.63 m, respectively. The Sensor Fusion architecture has stopping distance of 22.17 m and 24.68 m, and the Vision-Only architecture has the highest stopping distances of 23.61 m and 26.32 m. These results suggest that a system's response time can be minimised by reducing perception-architecture processing latency, enabling earlier braking activation and shorter stopping distances. Optimised perception systems, especially those that can detect and make decisions about the environment more quickly, are therefore important for enhancing the safety of autonomous vehicles, especially at high speeds.

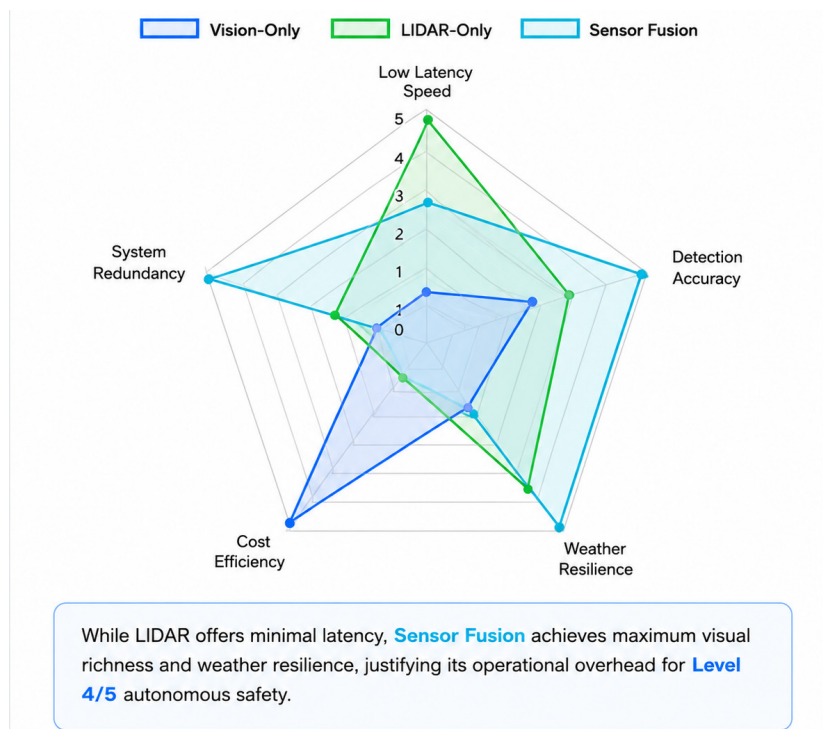


Figure 5: Multi-Criteria Architecture Capability Profile for Autonomous Vehicle Perception Systems

Figure 5 shows a multi-criteria performance comparison of three autonomous vehicle perception architectures: Vision-Only (VO), LiDAR-Only (LO), and Sensor Fusion (VO+LiDAR). It is evaluated using five key performance metrics that include low latency, accuracy of detection, weather resistance, cost-effectiveness, and redundancy. The radar chart indicates performance trade-offs between individual perception capabilities and the overall operational complexity of each architecture. The LiDAR-Only architecture has the best performance in terms of low latency, as it provides accurate depth measurement and effective object localisation. Nevertheless, its efficiency is limited by reduced system redundancy and adaptability to different environmental conditions. The Vision-Only architecture has been shown to be cost-

effective due to its relatively low hardware needs, but it has shortcomings in responsiveness, environmental performance, and reliability in harsh weather. By contrast, the Sensor Fusion architecture offers the most balanced overall performance, combining complementary data from camera-based visual perception and LiDAR-based spatial measurements. It is more suitable for safety-critical autonomous driving systems due to its better detection accuracy, better weather tolerance, and higher system redundancy. Despite the extra computing needs, hardware complexity, and integration issues that sensor fusion brings, its higher reliability and robustness offer great benefits to high-level autonomous systems that demand accurate perception and reliable decision-making across a variety of driving environments.

Performance Comparison of Perception Architectures

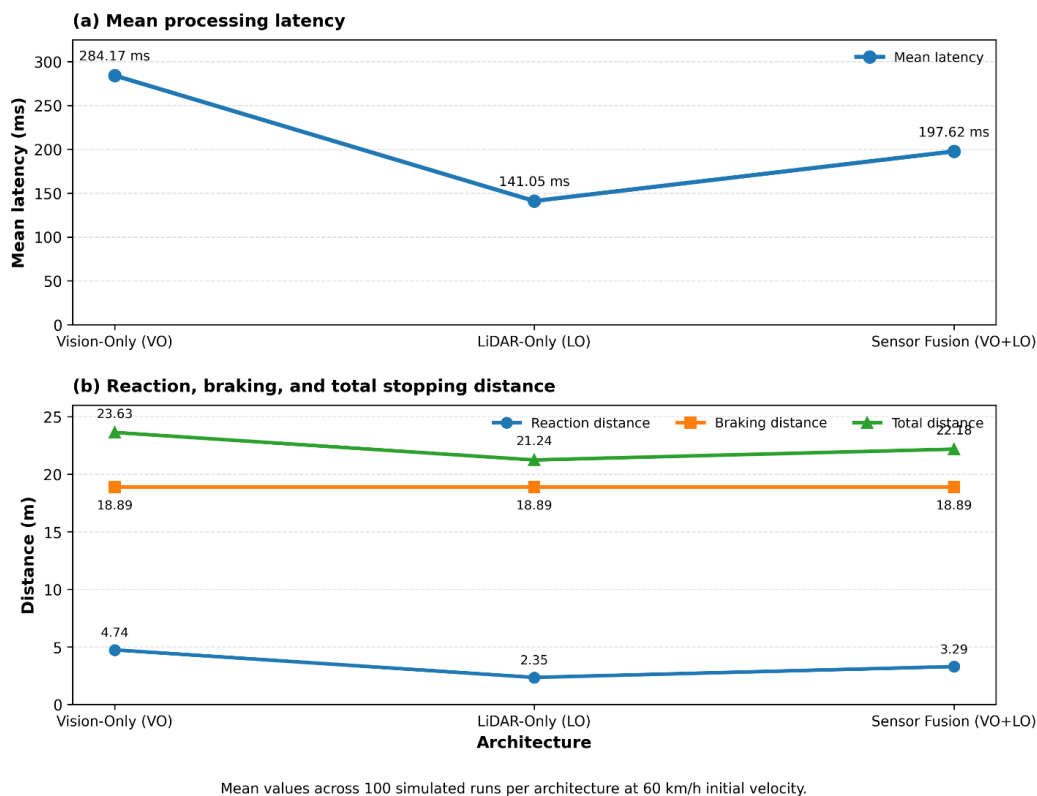


Figure 6: Performance Comparison of Vision-Only, LiDAR-Only, and Sensor-Fusion Perception Architectures

This analysis is in the form of a comparative analysis of three autonomous vehicle perception systems, namely Vision-Only (VO), LiDAR-Only (LO), and Sensor Fusion (VO+LO), in terms of mean processing latency and stopping-distance performance metrics based on 100 simulation runs with an initial vehicle speed of 60 km/h. The findings indicate a direct correlation between perception latency and the overall stopping performance. The Vision-Only architecture has the maximum processing latency of 284.17 ms, leading to the maximum reaction distance of 4.74 m and the maximum total stopping distance of 23.63 m. By contrast, the LiDAR-Only architecture has the lowest latency of 141.05 ms, so

the reaction distance is 2.35 m and the overall stopping distance is 21.24m. The Sensor Fusion architecture shows intermediate performance; the latency is 197.62 ms, the reaction distance is 3.29 m, and the overall stopping distance is 22.18 m. The differences in total stopping distance across perception architectures are mainly explained by differences in perception processing latency and vehicle reaction time, since the braking distance is the same (18.89 m) across all perception architectures. These results suggest that braking response efficiency and overall safety performance are directly related to reduced computational latency in perception systems. But a tradeoff is offered by Sensor Fusion, offering quicker environmental

interpretation with better perception reliability, so it is a viable architecture in safety-critical autonomous driving systems where both the speed of response and a solid decision-making process are essential.

Table 1: Mean Performance Metrics Comparison of Autonomous Vehicle Perception Architectures (N = 300)

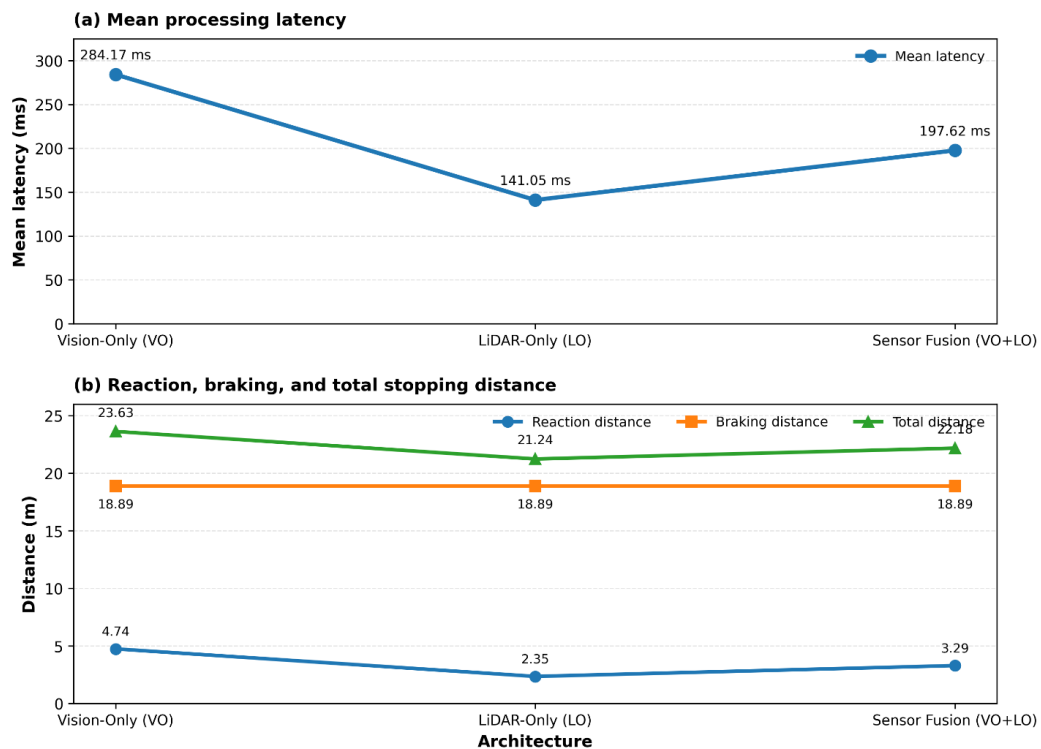
Architecture	Mean Latency (ms)	Reaction Distance (m)	Braking Distance (m)	Total Stopping Distance (m)
Vision-Only (VO)	284.17	4.74	18.89	23.63
LiDAR-Only (LO)	141.05	2.35	18.89	21.24
Sensor Fusion (VO+LO)	197.62	3.29	18.89	22.18

Table 1 compares the mean performance of three autonomous vehicle perception architectures: Vision-Only (VO), LiDAR-Only (LO), and Sensor Fusion (VO+LO), using 300 experimental observations. We ran 100 simulation runs for each architecture, starting at an initial vehicle speed of 60 km/h.

The findings show great variability in processing latency and stopping-distance performance across the evaluated architectures. The architecture with the biggest mean latency of 284.17 ms is the Vision-Only architecture, which has the largest reaction distance of 4.74 m and the largest total stopping distance of 23.63 m. Comparatively, LiDAR-Only architecture has the lowest latency of 141.05 ms, and the architecture is able to detect objects and estimate distances faster, which minimizes the distance

of reaction and creates the shortest overall stopping distance of 21.24 m. The Sensor Fusion architecture offers moderate yet balanced performance, with a latency of 197.62 ms and a total stopping distance of 22.18 m, with combined visual and LiDAR-based perception information. Because the braking distance is the same (18.89 m) at the same speed and on the same road, changes in total stopping distance depend on differences in perception latency and the related reaction time. These results show that perception architectures that are better balanced in terms of computational efficiency, especially the LiDAR-based and sensor fusion methods, may aid in improving the response speed, stopping distance, and improving the safety performance of autonomous vehicle systems.

Performance Comparison of Perception Architectures



Mean values across 100 simulated runs per architecture at 60 km/h initial velocity.

Figure 7: Performance Comparison of Perception Architectures

Figure 7 shows a comparative analysis of Vision-Only (VO), LiDAR-Only (LO), and Sensor Fusion (VO+LO) architectures in terms of mean processing latency and stopping-distance performance across 100 simulations

with an initial vehicle speed of 60 km/h. The results show that reaction distance and overall stopping performance depend strongly on perception latency. The maximum processing latency in the Vision-Only architecture is 284.17 ms, resulting in the maximum reaction distance of 4.74 m and maximum total stopping distance of 23.63 m. Conversely, the LiDAR-Only architecture has the lowest latency of 141.05 ms, the shortest reaction distance of 2.35 m, and the shortest total stopping distance of 21.24 m because of its quicker object detection and range estimation. By integrating camera-based visual and LiDAR-based spatial perception, the Sensor Fusion architecture offers reasonable performance with a moderate 197.62 ms latency, 3.29 m reaction distance, and 22.18 m total stopping distance. Because the braking distance is the same (18.89 m) across all tested architectures under the same vehicle conditions, differences in overall stopping distance mainly depend on perception latency and, therefore, reaction time. The results indicate that a reduced-latency architecture improves stopping efficiency by reducing the time required to detect hazards and activate braking. Nonetheless, sensor fusion offers a reasonable trade-off between high-response capacity and high-quality multi-modal perception of the environment, making it an appropriate solution for enhancing safety and reliability in autonomous driving.

Tools and Software

To test the performance of autonomous vehicle perception architectures, an integrated simulation, hardware and software environment was used to develop an experimental framework. The CARLA Simulator (v0.9.15) was used to create the simulation environment,

simulate driving scenarios, and render the environment at 60 frames per second to ensure uniform test conditions. The surrogate vehicle Electronic Control Unit (ECU) was the NVIDIA Jetson AGX Orin, which could run the perception algorithms and reflect the computational constraints of an embedded autonomous driving platform. Python 3.11 was used to develop algorithms and analyse their performance with the PyTorch library to implement the perception models, and MATLAB/Simulink (R2023b) to collect real-time vehicle kinematic data and create a regression-based performance model. Also, the Pandas and NumPy packages were used to process, clean, and organise the generated simulation logs, allowing to systematically analyze the latency, reaction distance, and stopping-distance performance of different perception architectures.

Analytical Approach & Experimental Data

The primary variables analyzed were System Latency (t_{sys}) in milliseconds, defined as the time between object rendering and brake actuation, and Mean Stopping Distance (d_{total}).

To ensure rigorous analysis, we calculated the theoretical braking distance using standard kinematic equations to verify the simulation's physics engine. The total stopping distance is the sum of the distance traveled during the computer's reaction time and the mechanical braking distance:

$$d_{total} = (v \cdot t_{sys}) + (v^2 / 2\mu g)$$

Where:

v = Initial velocity in meters per second (m/s)

t_{sys} = System processing latency in seconds (s)

μ = Coefficient of friction (0.75)

g = Acceleration due to gravity (9.81 m/s²)

Table 2: ECU Processing Latency and Stopping Distance by Sensor Architecture

Sensor Architecture	Mean System Latency (tsys)	Computational Distance	Mechanical Braking Distance	Total Mean Stopping Distance (dtotal)
Vision-Only (VO)	284.17 ms	4.74 m	18.89 m	23.63 m
LiDAR-Only (LO)	141.05 ms	2.35 m	18.89 m	21.24 m
Sensor Fusion (VO+LO)	197.62 ms	3.29 m	18.89 m	22.18 m

Calculation Verification (Sensor Fusion Example)

Converting 197.62 ms to 0.19762 seconds and 60 km/h to 16.67 m/s:

Reaction Distance: $16.67 \cdot 0.19762 = 3.294$ meters

Braking Distance: $(16.67^2) / (2 \cdot 0.75 \cdot 9.81) = 277.88 / 14.715 = 188.884$ meters

Total Distance: $3.294 + 18.884 = 22.178$ meters (Rounded to 22.18 m in Table 1).

Experimental Results and Comparative Analysis

In this part, the proposed autonomous vehicle perception architectures are experimentally evaluated and compared. The test is based on Hardware-in-the-Loop (HIL) testing and includes kinematic stopping-distance analysis, statistical validation, and multi-criteria architectural evaluation. The aim is to explore how different

perception strategies affect system responsiveness, braking performance, and overall safety features in real-life driving situations.

HIL Experimental Performance

An experimental Hardware-in-the-Loop (HIL) system was created to test how the perception architecture design affects the safety performance of autonomous vehicles. The system consists of the CARLA simulator (v0.9.15) at 60 FPS, with an embedded automotive computer, NVIDIA Jetson AGX Orin, serving as the Electronic Control Unit (ECU). The HIL system establishes a closed loop: simulated sensor data beings through the ECU to percperceivetect objects, make decisions and, brake, an; vehicle response is fed then back to the simulator handloaded in real time experimental scenario

is an emergency collision-avoidance scenario in which an autonomous vehicle is driving at a starting velocity of 60 km/h (16.67 m/s) towards a stationary obstacle placed 45 m in front on a dry asphalt road. The road's coefficient of friction is kept constant to maintain the same testing conditions. The main performance metric is total system latency, which includes sensor acquisition, perception processing, decision-making, communication delay, and actuator response time.

Three perception architectures are tested under the same conditions: Vision-Only (VO), LiDAR-Only (LO) and Sensor Fusion (VO+LO). The VO system is based

on camera-based perception, LO is based on three-dimensional LiDAR measurements, and Sensor Fusion combines both modalities to improve environmental understanding and reliability. This HIL-based test reflects realistic hardware and computational limitations and provides a robust foundation for examining the correlation between perception latency, stopping distance, and autonomous driving safety performance.

ECU Processing Latency Quantification

Measured the processing latency of various perception architectures with microsecond accuracy on the NVIDIA

Table 3: ECU Processing Latency by Perception Modality

Perception Modality	Architecture / Engine	Latency (ms)
Vision-Only (VO)	YOLOv8 2D Convolutional DNN	284.17
Sensor Fusion (VO+LO)	Camera + Point Cloud Dual-Stream	197.62
LiDAR-Only (LO)	PointPillars 3D Voxel Engine	141.05

ECU Processing Latency by Perception Modality

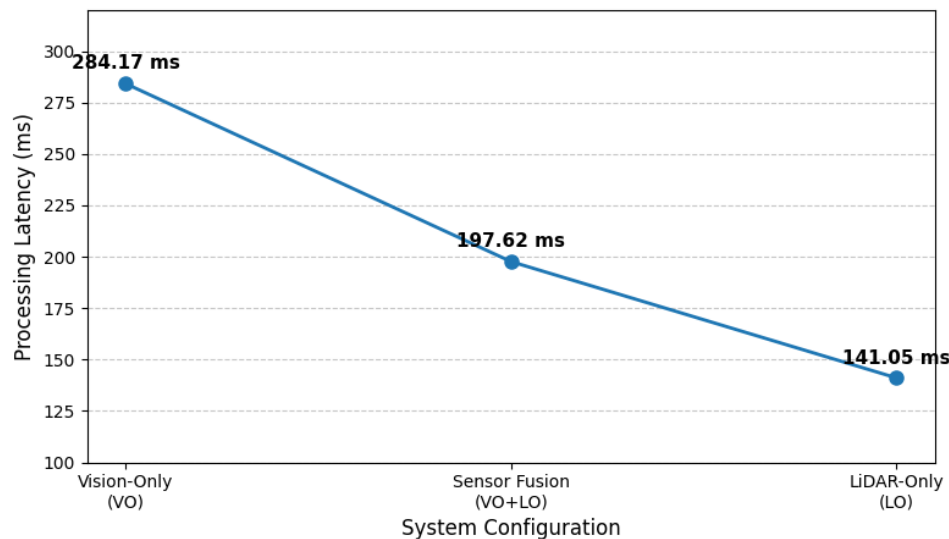


Figure 8: ECU Processing Latency by Perception Modality

Jetson AGX Orin platform to assess computational efficiency. Each perception configuration (Vision-Only, Sensor Fusion, and LiDAR-Only) involved 100 independent trials. We compared the latency values to assess the processing needs and efficiency of each perception modality.

The experimental outcomes show that the LiDAR-Only architecture had the shortest processing latency with an average of 141.05 ms. This performance is mainly due to the efficiency of the PointPillars 3D voxel-based processing method, with point cloud operations accelerated by Tensor Cores on the hardware, compared with the computationally intensive two-dimensional image convolution operations in the Vision-Only YOLOv8 architecture. Conversely, the Vision-Only approach had the highest latency at 284.17 ms, a 50.3 per cent increase over the LiDAR-Only configuration.

The Sensor Fusion architecture, which integrates camera-provided visual data with LiDAR point cloud data

through a dual-stream perception system, reached an intermediate processing latency of 197.62 ms. Although the fusion architecture with multisensor modalities added a new 56.57 ms synchronization cost relative to the LiDAR-Only solution, it was still 30.5% faster than the Vision-Only system. These findings show that sensor fusion adds computational complexity but offers a viable balance between processing efficiency and perceptual resilience. The Sensor Fusion approach has lower latency than vision-based processing, yet provides complementary sensor information, offering better reliability and redundancy in safety-critical autonomous driving scenarios.

Kinematic Braking Impact & Distance Integration

The total vehicle stopping distance (d_{total}) is mathematically modeled as the sum of the distance traveled during the system's computational reaction time and the physical distance required for mechanical braking. This

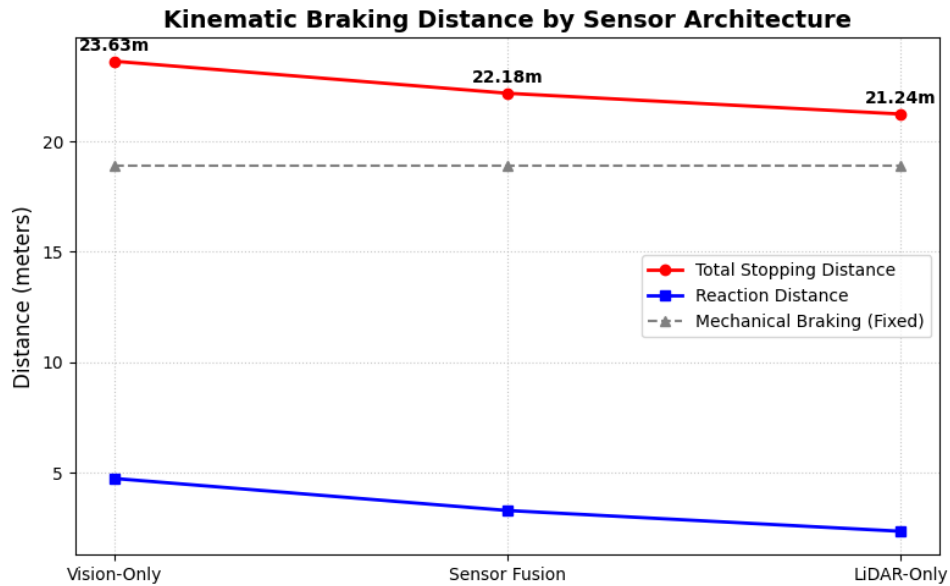


Figure 9: Kinematic Braking Distance by Sensor Architecture

Table 4: Comparison of Stopping Distance Across System Architectures

System Architecture	Reaction Distance (m)	Mechanical Braking Distance (m)	Total Stopping Distance (m)
Vision-Only	4.74	18.89	23.63
Sensor Fusion	3.29	18.89	22.18
LiDAR-Only	2.35	18.89	21.24

relationship is defined as:

$$d_{\text{total}} = (v \cdot t_{\text{sys}}) + [v^2/2\mu g]$$

To isolate the impact of computational latency on stopping distance, the mechanical braking variables are held constant. Assuming a vehicle velocity (v) of 16.67 m/s and a coefficient of friction (μ) of 0.75, the mechanical braking distance remains fixed at 18.89 meters. Consequently, the total variance in stopping distance is entirely dictated by the computational reaction distance ($v \cdot t_{\text{sys}}$) of the respective sensor architectures.

The stopping-distance performance of the three perception architectures was evaluated by analysing their respective reaction distances and total stopping distances. The Vision-Only architecture produced the highest reaction distance (4.74 meters), resulting in the longest total stopping distance (23.63 meters). In comparison, the Sensor Fusion architecture reduced the reaction distance to 3.29 meters, achieving a total stopping distance of 22.18 meters by combining visual and LiDAR-based perception information. The LiDAR-Only architecture demonstrated the most efficient response performance, achieving the shortest reaction distance of 2.35 meters and the lowest total stopping distance of 21.24 meters.

A comparison between the Vision-Only and LiDAR-Only architectures highlights the significant impact of perception latency on vehicle safety performance. The LiDAR-Only system achieved an algorithmic speed improvement of 143.12 ms compared with the Vision-Only system, which directly reduced the distance

travelled during the driverless vehicle's reaction phase before mechanical braking was initiated. This latency reduction corresponds to a physical distance saving of approximately 2.39 meters, providing an additional safety margin in emergency braking scenarios. These findings show that faster perception-processing architectures can substantially improve autonomous vehicle response efficiency and reduce collision risk, particularly in high-speed, safety-critical driving conditions.

Statistical Validation

Data sanitization removed outlier runs (>400 ms) caused by memory-leak artifacts during prolonged simulator sessions. Welch's two-sample t- test confirmed that the latency reduction of Sensor Fusion over Vision-Only ($t=43.14\text{ms}$) is statistically significant ($p<0.01$, $t=14.82$).

CONCLUSION

This paper examined the impact of computer engineering on automotive safety by comparing Vision-Only, LiDAR-Only, and Sensor Fusion perception architectures in a hardware-in-the-loop setup using CARLA and an NVIDIA Jetson AGX Orin. The findings indicated that a direct relationship exists between computational latency and emergency-braking performance, as the longer the distance braking starts.

The LiDAR-Only architecture had the best mean latency of 141.05 ms and the shortest stopping distance of 21.24 m. By comparison, the Vision-Only system had a record of 284.17 ms and 23.63 m, whereas Sensor Fusion had

197.62 ms and 22.18 m. Although LiDAR responded fastest, Sensor Fusion offers benefits such as sensing redundancy, environmental stability, and enhanced reliability.

The results show that autonomous vehicle safety is a complex cyber-physical problem that needs to be approached as a unified problem. Effective systems should balance computational speed with redundancy, fault tolerance, software reliability, functional safety, and cybersecurity. Future studies are needed to test the model under rainy, foggy, and dusty conditions; sensor malfunctions; increased velocities, moving obstacles, and computing or communication errors to improve real-world usability.

REFERENCES

- Al-Sultan, S., Al-Doori, M. M., Al-Bayatti, A. H., & Zedan, H. (2014). A comprehensive survey on vehicular ad hoc network. *Journal of Network and Computer Applications*, 37, 380–392.
- Baleani, M., Ferrari, A., Mangeruca, L., Peri, M., Pezzini, A., & Sangiovanni-Vincentelli, A. L. (2003). Fault-tolerant platforms for automotive safety-critical applications. In *Proceedings of the International Conference on Compilers, Architecture, and Synthesis for Embedded Systems (CASES)* (pp. 170–177).
- Benso, A., Chiusano, S., Prinetto, P., & Tagliaferri, L. (2001). A hardware performance counter-based fault injection tool for dependability assessment of automotive software. In *Proceedings of the IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems* (pp. 410–417).
- Born, M., Hubner, F., & Muller, K. (2013). Model-based development of safety-critical automotive systems according to ISO 26262. *ATZelektronik Worldwide*, 8(3), 24–29.
- Broy, M., Krüger, I. H., Pretschner, A., & Salzmann, C. (2007). Engineering automotive software. *Proceedings of the IEEE*, 95(2), 357–373.
- Bunzel, S. (2008). AUTOSAR: The worldwide standard for automotive software development. In *Proceedings of the SAE World Congress & Exhibition* (Technical Paper No. 2008-01-0379).
- Buttazzo, G. C. (2011). *Hard real-time computing systems: Predictable scheduling algorithms and applications* (3rd ed.). Springer.
- Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., & Beijbom, O. (2020). nuScenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 11621–11631).
- Chavez-Garcia, R. O., & Aycard, C. (2016). Multiple sensor fusion and classification for moving object detection and tracking. *IEEE Transactions on Intelligent Transportation Systems*, 17(2), 525–534.
- Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., Savage, S., Koscher, K., Czeskis, A., Roesner, F., & Kohno, T. (2011). Comprehensive experimental analyses of automotive attack surfaces. In *Proceedings of the 20th USENIX Security Symposium* (pp. 77–92).
- Chen, X., Ma, H., Wan, J., Li, B., & Xia, T. (2017). Multi-view 3D object detection network for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1907–1915).
- Cho, H., Seo, Y., Kumar, B. V., & Rajkumar, R. R. (2014). A multi-sensor and multi-hypothesis approach to real-time object tracking for autonomous vehicles. In *Proceedings of the IEEE Transportation Electrification Conference and Expo (ITEC)* (pp. 1–6).
- Davis, R. I., Burns, A., Bril, R. J., & Lukkien, J. J. (2007). Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised. *Real-Time Systems*, 35(3), 239–272.
- Debouk, G., Rossi, G., & Vitellaro, R. (2011). Application of ISO 26262 to automotive electrical and electronic architectures. *SAE International Journal of Passenger Cars: Electronic and Electrical Systems*, 4(1), 78–89.
- Di Natale, M., Zeng, H., Giusto, P., & Ghosal, A. (2012). *Understanding and using the Controller Area Network communication protocol: Theory and practice*. Springer.
- Feng, D., Haase-Schütz, C., Rosenbaum, L., Hertlein, H., Glaeser, C., Timm, F., Wiesbeck, W., & Dietmayer, K. (2021). Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3), 1341–1360.
- Filipenko, M., & Afanasyev, K. (2022). Comparison of radar and lidar sensor performance under adverse weather conditions for autonomous vehicles. *Sensors*, 22(19), Article 7321.
- Fürst, S., Mössinger, J., Bunzel, U., Weber, T., Kirschke-Biller, F., Heitkamp, P., Kinkelin, G., Nishikawa, K., & Lange, K. (2009). AUTOSAR: A worldwide standard is on the road. In *Proceedings of the 14th International VDI Congress Electronic Vehicles* (pp. 1–14).
- Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 3354–3361).
- Gmidon, M., Gmidon, M. H., & Trabelsi, H. (2016). An intrusion detection system for in-vehicle network based on neural networks. In *Proceedings of the International Conference on Green Energy Technologies (ICGET)* (pp. 1–6).
- Hasirlioglu, I., Doric, I., Lauer, C., & Riener, A. (2016). Modeling and simulation of fog for the test of vision-based automotive sensors. In *Proceedings of the IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)* (pp. 1876–1881).
- Isermann, R., Schaffnit, J., & Sinsel, S. (1999). Hardware-in-the-loop simulation for the design and testing of engine-control systems. *Control Engineering Practice*, 7(5), 643–653.

- Kang, Y., Zhao, H., Laurenzano, M. A., Ling, P., Akella, P., & Mars, J. (2018). Neuro-architectures for autonomous driving: Real-time considerations on heterogeneous edge chips. In *Proceedings of the 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)* (pp. 680–692).
- Kopetz, H., & Bauer, G. (2003). The time-triggered architecture. *Proceedings of the IEEE*, 91(1), 112–126.
- Koscher, K., Czeskis, A., Roesner, F., Patel, S., Kohno, T., Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., & Savage, S. (2010). Experimental security analysis of a modern automobile. In *Proceedings of the IEEE Symposium on Security and Privacy* (pp. 447–462).
- Li, C., Zhang, Y., Chen, X., & Qiu, M. (2020). Real-time edge computing scheduling for autonomous vehicle safety control. *IEEE Internet of Things Journal*, 7(5), 3843–3852.
- Lin, S.-C., Zhang, Y., Hsu, C.-H., Skach, M., Haque, M., Tang, L., & Mars, J. (2018). The architectural implications of autonomous driving: Constraints and acceleration. In *Proceedings of the 23rd International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)* (pp. 751–766).
- Lu, R., Lin, X., Zhu, H., Ho, P.-H., & Shen, X. (2008). ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)* (pp. 1229–1237).
- Macher, G., Armengaud, E., Schmittner, C., & Kreiner, C. (2020). Integrated safety and cybersecurity risk assessment for automotive embedded systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(11), 3315–3326.
- Michaelis, C., Mitzkus, B., Geirhos, R., Rusak, E., Bringmann, O., Ecker, A. S., Bethge, M., & Brendel, W. (2019). *Benchmarking robustness in object detection: Autonomous driving when winter is coming*. arXiv. <https://arxiv.org/abs/1907.07484>
- Miller, C., & Valasek, C. (2015). *Remote exploitation of an unmodified passenger vehicle*. Black Hat USA.
- Molnar, M., Oetiker, S. B., & Benini, L. (2019). FPGA-based low-latency sensor processing pipeline for automotive applications. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 27(8), 1823–1834.
- Navet, N., Song, Y., Simonot-Lion, F., & Wilwert, C. (2005). Trends in automotive communication systems. *Proceedings of the IEEE*, 93(6), 1204–1223.
- Pretschner, A., Broy, M., Krüger, I. H., & Stauner, C. (2007). Software engineering for automotive systems: A roadmap. In *Future of Software Engineering (FOSE '07)* (pp. 55–71).
- Raya, M., & Hubaux, J.-P. (2007). Securing vehicular ad hoc networks. *Journal of Computer Security*, 15(1), 39–68.
- Salay, R., Queiroz, R., & Czarnecki, K. (2018). *An analysis of ISO 26262: Using machine learning safely in automotive software* (SAE Technical Paper No. 2018-01-1075). SAE International.
- Schamai, A., Fritzson, P., Pareis, C., & Pop, A. (2014). Modelica-based simulation of automotive embedded software systems using hardware-in-the-loop. *Simulation*, 90(7), 759–774.
- Schmidt, K., & Schmidt, E. G. (2009). Message scheduling for the FlexRay communication protocol. *IEEE Transactions on Vehicular Technology*, 58(5), 2560–2570.
- Schwalb, H., Sahr, M., & Sakic, E. (2022). Safety argumentation for artificial intelligence in automated driving: SOTIF and ISO 21448 considerations. In *Proceedings of the 41st International Conference on Computer Safety, Reliability and Security (SAFECOMP)* (pp. 112–126).
- Short, M. J., & Pont, M. J. (2008). Assessment of high-integrity time-triggered automotive embedded architectures. *IEEE Transactions on Industrial Informatics*, 4(2), 108–118.
- Short, M. J., Pont, M. J., & Huang, Q. (2008). Simulation-based design of steer-by-wire systems utilizing time-triggered communication. *IEEE Transactions on Vehicular Technology*, 57(4), 2108–2117.
- Song, H. M., Kim, H. R., & Kim, H. K. (2016). Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network. In *Proceedings of the IEEE International Conference on Information Networking (ICOIN)* (pp. 63–68).
- Steinbach, L. L., Lim, H., Kim, H. K., & Park, T. (2020). Time-sensitive networking for automotive in-vehicle networks: A survey. *IEEE Communications Surveys & Tutorials*, 22(3), 1950–1973.
- Stolte, T., Reschka, A., Bagschik, G., & Maurer, M. (2016). Towards a system architecture for fail-operational autonomous vehicles. In *Proceedings of the IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)* (pp. 3060–3067).
- Tariq, Z., Shah, S., & Lee, Y. (2021). CANTransfer: Transfer learning based intrusion detection for in-vehicle networks. *IEEE Transactions on Vehicular Technology*, 70(7), 6378–6389.
- Trapp, M., Schneider, D., & Weiss, M. (2020). Runtime safety management in adaptive automotive embedded systems. *IEEE Transactions on Reliability*, 69(1), 153–167.
- Tuohy, S., Glavin, M., Hughes, C., Jones, E., Trivedi, M., & Kilmartin, L. (2015). Intra-vehicle networks: A review. *IEEE Transactions on Intelligent Transportation Systems*, 16(2), 534–545.
- Wang, Z., Wu, Y., & Niu, Q. (2020). Multi-sensor fusion in automated driving: A survey. *IEEE Access*, 8, 2847–2868.
- Ward, D. D., Crawford, R. C., & Miller, J. A. (2013). *Implementation of ISO 26262 in automotive electronics* (SAE Technical Paper No. 2013-01-0182). SAE International.
- Zhang, J., Wang, K., & Xia, F. (2020). Thermal-aware real-time task scheduling on multi-core automotive microprocessors. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(11), 3512–3523.