vehicles

Article

FPGA–STM32-Embedded Vision and Control Platform for ADAS Development on a 1:5 Scale Vehicle

Karen Roa-Tort [1], Diego A. Fabila-Bustos [1], Macaria Hernández-Chávez [1], Daniel León-Martínez [1], Adrián Apolonio-Vera [1], Elizama B. Ortega-Gutiérrez [1], Luis Cadena-Martínez [1], Carlos D. Hernández-Lozano [1], César Torres-Pérez [1], David A. Cano-Ibarra [1], J. Alejandro Aguirre-Anaya [2] and Josué D. Rivera-Fernández [1,*]

[1] Laboratorio de Optomecatrónica y Energías, UPIIH Instituto Politécnico Nacional, Distrito de Educación, Salud, Ciencia, Tecnología e Innovación, San Agustín Tlaxiaca 42162, Hidalgo, Mexico; kroat@ipn.mx (K.R.-T.); dfabilab@ipn.mx (D.A.F.-B.); mhernandezch@ipn.mx (M.H.-C.); dleonm1902@alumno.ipn.mx (D.L.-M.); aapoloniov1900@alumno.ipn.mx (A.A.-V.); eortegag1600@alumno.ipn.mx (E.B.O.-G.); lcadenam1602@alumno.ipn.mx (L.C.-M.); chernandezl1601@alumno.ipn.mx (C.D.H.-L.); ctorres1804@alumno.ipn.mx (C.T.-P.); dcanoi2000@alumno.ipn.mx (D.A.C.-I.)

[2] Unidad Profesional Interdisciplinaria de Energía y Movilidad, Instituto Politécnico Nacional, Av. Wilfrido Massieu, Adolfo López Mateos S/N, Nueva Industrial Vallejo, Gustavo A. Madero 07738, Mexico; jaguirrea@ipn.mx

* Correspondence: jdriveraf@ipn.mx a

Abstract

This paper presents the design, development, and experimental validation of a low-cost, modular, and scalable Advanced Driver Assistance System (ADAS) platform intended for research and educational purposes. The system integrates embedded computer vision and electronic control using an FPGA for accelerated real-time image processing and an STM32 microcontroller for sensor data acquisition and actuator management. The YOLOv3-Tiny model is implemented to enable efficient pedestrian and vehicle detection under hardware constraints, while additional vision algorithms are used for lane line detection, ensuring a favorable trade-off between accuracy and processing speed. The platform is deployed on a 1:5 scale gasoline-powered vehicle, offering a safe and cost-effective testbed for validating ADAS functionalities, such as lane tracking, pedestrian and vehicle identification, and semi-autonomous navigation. The methodology includes the integration of a CMOS camera, an FPGA development board, and various sensors (LiDAR, ultrasonic, and Hall-effect), along with synchronized communication protocols to ensure real-time data exchange between vision and control modules. A wireless graphical user interface (GUI) enables remote monitoring and teleoperation. Experimental results show competitive detection accuracy—exceeding 94% in structured environments—and processing latencies below 70 ms per frame, demonstrating the platform's effectiveness for rapid prototyping and applied training. Its modularity and affordability position it as a powerful tool for advancing ADAS research and education, with high potential for future expansion to full-scale autonomous vehicle applications.

Keywords: ADAS; FPGA-based embedded computer vision; autonomous vehicle development; electronic control and sensing system; driver safety technology

check for updates

Academic Editors: Meng Li and Sijing Guo

Received: 26 June 2025
Revised: 25 July 2025
Accepted: 15 August 2025
Published: 17 August 2025

Citation: Roa-Tort, K.; Fabila-Bustos, D.A.; Hernández-Chávez, M.; León-Martínez, D.; Apolonio-Vera, A.; Ortega-Gutiérrez, E.B.; Cadena-Martínez, L.; Hernández-Lozano, C.D.; Torres-Pérez, C.; Cano-Ibarra, D.A.; et al. FPGA–STM32-Embedded Vision and Control Platform for ADAS Development on a 1:5 Scale Vehicle. *Vehicles* **2025**, 7, 84. https://doi.org/10.3390/vehicles7030084

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The development of Advanced Driver Assistance Systems (ADAS) has become a critical objective in automotive research, targeting improvements in road safety, traffic

efficiency, and driver comfort through functionalities such as lane departure warning, obstacle detection, pedestrian recognition, and adaptive cruise control [1,2]. Among the various sensing modalities, vision-based systems are particularly notable for their rich contextual awareness, including lane marking identification, traffic sign detection, and real-time obstacle tracking [2,3]. However, embedding computer vision in automotive systems imposes significant challenges related to computational demand, energy consumption, and latency.

While traditional solutions using GPUs or multicore CPUs deliver significant processing power, they frequently exhibit high power draw and nondeterministic latency, rendering them less suitable for constrained embedded automotive platforms [2,4]. FPGA-based embedded computer vision offers a promising alternative by enabling parallel processing pipelines, deterministic latency, and energy-efficient operation [2,5]. Notably, implementations such as YOLOv3-Tiny on development boards like the Kria KV260 achieve detection accuracies exceeding 90% with low power consumption (~5 W), demonstrating the viability of FPGA-based vision for ADAS applications [5,6].

In parallel, robust ADAS requires multi-sensor integration. Techniques combining LiDAR, ultrasonic, and Hall-effect sensors enhance vehicle perception and control reliability [7]. While full-scale automotive systems typically employ CAN or LIN buses for ECU communication, academic and prototype platforms often favor simpler serial protocols such as I^2C and UART for their ease of implementation and educational value [8,9]. Despite these simpler interfaces, the underlying architecture remains scalable and conceptually consistent with industrial communication standards.

Several projects have advanced FPGA-driven sensing and actuation: neuro-fuzzy ADAS sensors for personalized driving behavior models [10] and eco-driving promoters using real-time, on-vehicle FPGA computing platforms [11]. Similarly, modular FPGA systems have been deployed for pavement defect detection and traffic sign recognition [12,13]. Nevertheless, comprehensive platforms that integrate vision, sensor fusion, real-time control via STM32, and wirelessly interfaced GUIs for teleoperation are still scarce.

To address this gap, this manuscript presents the design, integration, and experimental validation of a hybrid ADAS prototype built on a 1:5 scale gasoline-powered vehicle. It features the following:

- FPGA-based computer vision, utilizing YOLOv3-Tiny for detection and lane tracking.
- STM32-based sensing and actuation, handling LiDAR (via I^2C), ultrasonic, and Hall-effect sensors and Pulse Width Modulation-driven servomotors.
- Synchronized communication via UART, facilitating data transfer between vision and control units.
- Wireless GUI interface, enabling real-time monitoring, teleoperation, and system feedback.

This modular, cost-effective, energy-efficient, and extensible framework positions the platform as a strong candidate for academic training, rapid prototyping, and applied ADAS research while maintaining architectural concepts transferable to industrial environments through future integration with CAN/LIN standards.

The primary purpose of this study is to design and validate a low-cost, modular, and scalable ADAS research platform that enables both academic training and applied development in autonomous vehicle technologies. Unlike previous works focused on single-purpose systems or resource-intensive solutions, this study addresses the lack of integrated, accessible platforms combining real-time vision processing, multi-sensor data acquisition, and electronic control suitable for small-scale prototyping. The main novelty lies in the integration of FPGA-based embedded computer vision with STM32-controlled

sensing and actuation within a unified, synchronized architecture, all implemented on a gasoline-powered 1:5 scale vehicle. This approach offers an energy-efficient, educational, and easily extensible system that bridges the gap between academic research platforms and industrial ADAS architectures. The contributions of this work include the hardware and software integration methodology, the demonstration of system scalability and modularity, and the experimental validation highlighting its potential as a practical tool for both research and education in intelligent mobility systems.

It is important to highlight that the present work focuses on the detection of pedestrians, vehicles, and lane markings, rather than directly addressing obstacle avoidance. This distinction is crucial, as the primary objective of this initial development stage is to establish a perception system capable of supporting additional functionalities in future phases—such as obstacle avoidance, autonomous navigation, dynamic path planning, or other applications for academic or research purposes.

The remainder of this paper is organized as follows: Section 2 details the materials and methods, including the system architecture, hardware components, and the implementation of the computer vision and control modules. Section 3 presents the experimental setup and evaluation procedures used to validate the proposed platform. Section 4 discusses the obtained results in terms of detection accuracy, processing latency, and overall system performance, including an analysis of its practical applicability. Finally, Section 5 summarizes the main conclusions and outlines potential directions for future work, emphasizing the scalability and educational value of the developed ADAS platform.

2. Materials and Methods

The experimental platform consisted of a 1:5 scale vehicle powered by a two-stroke gasoline engine, which served as a testbed for the integration of perception and control subsystems. Key components included a Kria KV260 FPGA-based vision processing board, an STM32 microcontroller for real-time control, a suite of sensors (LiDAR, ultrasonic, and Hall-effect), and servomotors for steering and throttle actuation. The methodology encompassed data acquisition for computer vision training, hardware integration, vibration mitigation strategies, and performance evaluation in a controlled test environment, based on recent works on autonomous vehicle perception systems using SoC FPGA [14,15]. Each subsystem was designed to operate under conditions replicating real-world scenarios, enabling a robust assessment of the system's potential for academic and pre-commercial ADAS research.

2.1. System Integration and Prototype Architecture

The developed system was implemented on a 1:5 scale vehicle and consisted of three main subsystems: the power supply system, the embedded vision system, and the master–slave control system. The architecture is centered around the Kria KV260 Vision AI board, which integrates an ARM processor and an FPGA within a System-on-Chip (SoC). This configuration enables local image processing via a USB camera and wireless transmission of results to a mobile device. This approach follows strategies reported in literature for FPGA platforms in autonomous vehicles [14,16]. The master–slave control system uses a microcontroller-based electronic control unit (ECU) responsible for actuator management, sensor interfacing, and communication with the embedded vision system. This distributed integration allows real-time control and aligns with validated methods in recent studies [15,16]. Control commands and vehicle status are managed through a custom graphical user interface (GUI) running on a mobile device, enabling real-time monitoring and manual override.

2.2. Vision System

The vision subsystem comprises a USB camera responsible for capturing environmental images at the front of the vehicle and a Kria KV260 Vision AI board, which integrates a processor with serial computation capabilities alongside a programmable logic device for image processing and data transmission to the visualization interface. Power is supplied by a Li-Po battery coupled with a voltage regulation and short-circuit protection system to ensure stable operation of the components. The implementation leverages hardware-accelerated FPGA algorithms that significantly reduce latency, compared to CPU-only solutions [14,17]. Processed images are subsequently transmitted via a Wi-Fi module to a mobile device, enabling visualization of lane segmentation, vehicle detection, and pedestrian recognition. The system leverages the FPGA-based Kria KV260 Vision AI board for efficient hardware acceleration. Figure 1 presents a block diagram illustrating the key components of the vision subsystem.

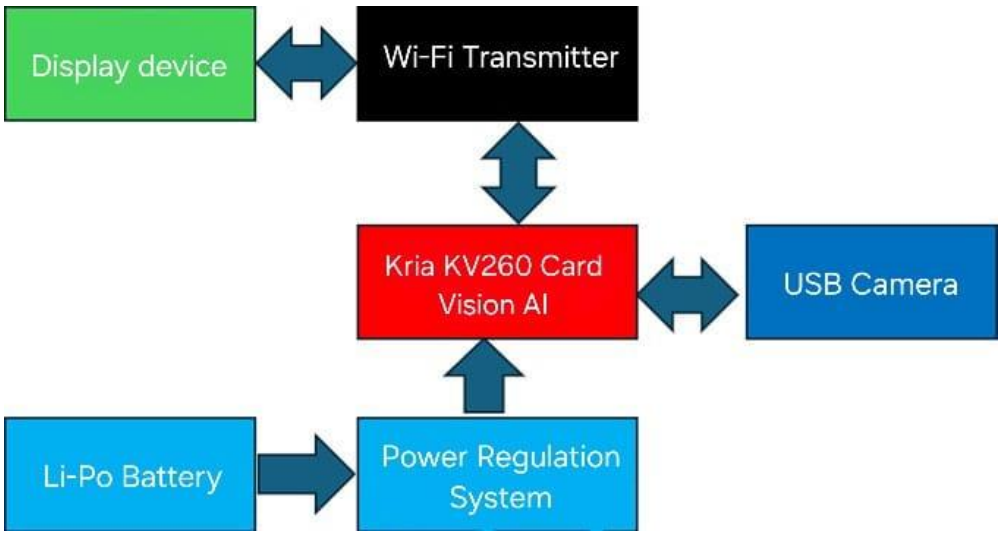


Figure 1. General vision system block diagram.

2.2.1. Power Supply System

The power supply was designed according to the energy requirements of the Kria KV260 board, which operates at 12 V, with a maximum current draw of 3 A. (It is 36 W.) A 14.7 V Li-Po battery regulated by an FTVOGUE B07PXWMNN6 converter was used, capable of delivering up to 8 A. Protection components included a 3 A fuse, control switch, and a voltage monitor with an audible alert when cell voltage drops below 3.3 V. The battery was housed in a dual-case system—plastic inner and metal outer—for impact and fire protection. Components were mounted on a 6 mm acrylic base for structural integration and electrical isolation.

2.2.2. Camera Selection and Integration

Different camera options were evaluated based on frame rate, latency, field of view, and interface compatibility. The Logitech C920 Pro HD was selected due to its 30 fps rate, low latency, 78° field of view, and USB interface, as well as abundant technical documentation and low cost. It was mounted in a weather-protected enclosure with a cooling fan to ensure stable forward-facing vision.

Due to the considerable vibrational stresses induced by the high rotational speed of the two-stroke gasoline engine—capable of reaching up to 19,000 revolutions per minute (RPM)—it was imperative to develop a dedicated mechanical support structure to preserve the stability, alignment, and operational integrity of the embedded vision and control

subsystems. To this end, a custom-engineered aluminum framework was designed and manufactured using high-precision machining techniques to ensure structural rigidity and dimensional consistency. The assembly was firmly anchored to the vehicle chassis using high-strength fastening elements, including grade-rated bolts, self-locking nuts, stainless steel washers, and U-clamps, selected to withstand dynamic loads and mechanical fatigue.

To attenuate the propagation of vibrations toward sensitive electronic components, a multi-stage vibration isolation strategy was implemented. This included elastomeric rubber washers positioned at key contact interfaces, C-profile rubber edge trims for peripheral damping, and polyurethane-based anti-vibration tape applied at structural junctions. Furthermore, all critical joints were secured using dual-locking mechanisms to prevent mechanical loosening under prolonged dynamic excitation. The comprehensive integration of these isolation techniques resulted in a significant reduction in mechanical resonance and vibrational noise, thereby improving the functional reliability and lifespan of the mechatronic subsystems under realistic operating conditions, following common practices and reported techniques to protect sensitive electronic components in embedded systems [18,19]. Additionally, enhanced mechanical stability contributed to a more accurate and consistent acquisition of visual data, improving the quality and precision of image capture for subsequent processing stages.

2.2.3. Implementation of the FPGA-Embedded Vision System

The embedded vision system was deployed on the Xilinx Kria KV260 development board, which integrates an ARM processor and FPGA fabric, enabling hybrid computing. The pynq-dpu library was utilized to accelerate deep learning inference using the FPGA's programmable logic, significantly reducing latency compared to CPU-only solutions. Lane detection was implemented via a sliding window algorithm optimized for real-time execution, while object detection relied on the YOLOv3 architecture adapted through transfer learning. This approach, validated in recent works, achieves real-time processing with high accuracy and energy efficiency [14,20]. Pre-processed images were streamed from the USB camera, analyzed on the FPGA, and the output data—including object classes and bounding boxes—were formatted for wireless transmission. A custom Android application received and visualized the detection results in real time, offering a user-friendly graphical interface for system monitoring.

Neural network training was conducted using Google Colab's cloud-based infrastructure, leveraging a pre-trained YOLOv3 model to accelerate convergence through transfer learning. The training process was executed on a Tesla T4 GPU, optimizing the detection performance on the custom dataset. After 1200 iterations using a batch size of 16 and an initial learning rate of 0.001, the model demonstrated robust performance metrics, with high confidence scores for vehicle and pedestrian detection under varied urban scenes. The trained weights were quantified and converted for deployment on the Kria KV260 platform, ensuring compatibility with the FPGA-based deep learning acceleration pipeline. This approach significantly improved inference speed and energy efficiency, key requirements for embedded automotive applications.

2.3. Electronic Control System

The electronic control system was designed to coordinate the perception data from the embedded vision subsystem with real-time actuation of the vehicle's steering and throttle systems. Central to this architecture is an STM32 microcontroller, selected for its high-performance ARM Cortex-M core and integrated peripherals optimized for embedded control applications. The system integrates multiple sensor modalities, including a LiDAR module for distance measurement, an ultrasonic sensor for cars or persons proximity

detection, and a Hall-effect sensor for rotational speed monitoring, following architecture proposed in current ADAS research [16,21]. These inputs are processed and fused to enhance environmental awareness and decision-making. Actuation is achieved through precise control of servomotors responsible for steering and throttle modulation. Furthermore, wireless communication modules ensure bidirectional data flow with external devices, enabling real-time telemetry and remote system oversight. This configuration provides a robust and scalable platform for implementing and validating advanced control algorithms, serving both research and academic training in autonomous vehicle technologies.

The spatial distribution of the sensors and electronic components was carefully planned to ensure optimal performances of both the perception and control subsystems. The LiDAR sensor was mounted at the front of the vehicle to enable accurate distance measurement, while the ultrasonic sensor was positioned laterally to enhance spatial awareness during low-speed maneuvers. A Hall-effect sensor was installed near the rear axle to provide real-time feedback on wheel rotation, supporting velocity estimation, and closed-loop control. The STM32 microcontroller, responsible for processing sensor data and generating control signals, was centrally located to minimize wiring complexity and signal latency. Servomotors for steering and throttle control were directly coupled to the mechanical linkages of the front axle and the throttle lever, respectively. Additionally, the embedded vision system based on the Kria KV260 FPGA board was mounted within a vibration-isolated enclosure on the upper deck of the chassis. Figure 2 illustrates the detailed layout of these components on the 1:5 scale vehicle, providing a visual reference for the physical integration strategy described.

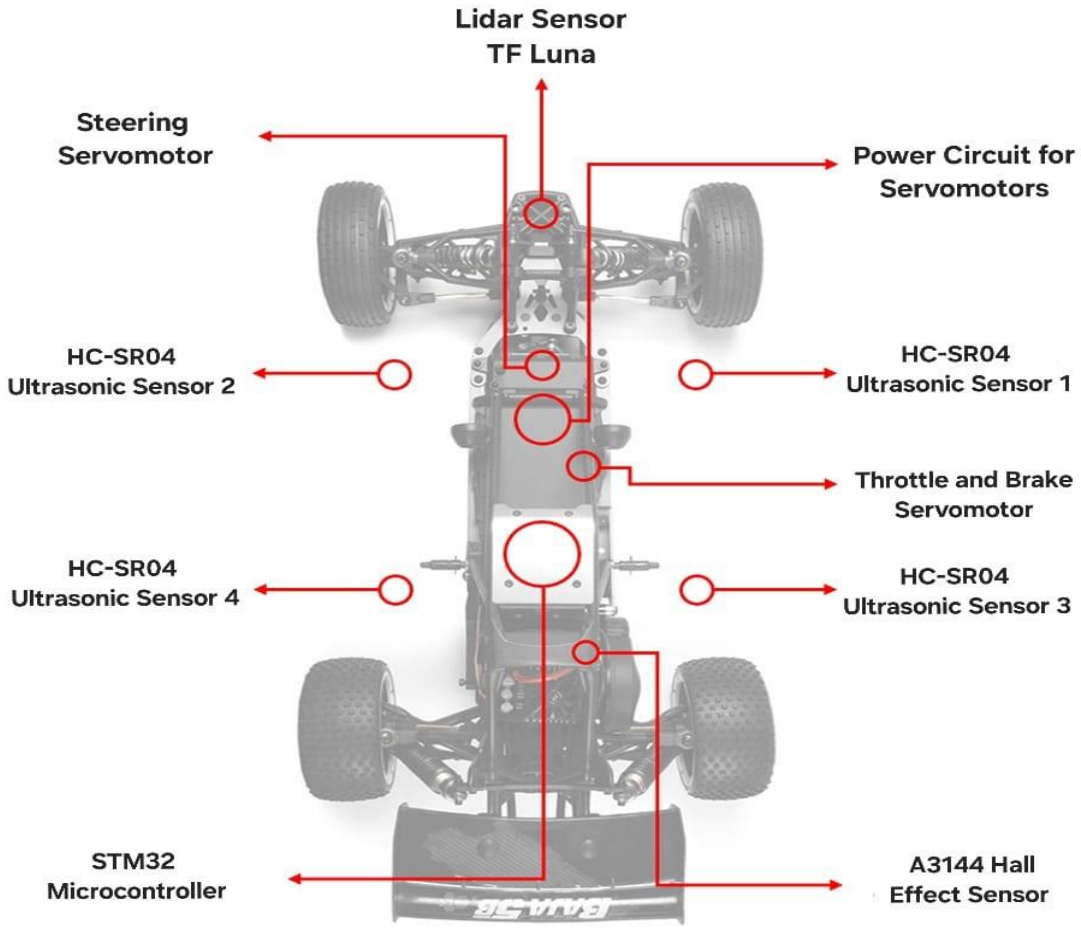


Figure 2. Top-view schematic layout of the 1:5 scale vehicle, illustrating the placement of the main electronic and sensing components.

2.3.1. STM32 Microcontroller

The STM32 NUCLEO-F401RE board, based on a 32-bit ARM Cortex-M4 core, is the central controller. It was selected for its processing power and extensive peripheral support, including USART, I^2C, Pulse Width Modulation timers, and external interruptions. It controls steering and throttle servomotors, a standard practice documented in embedded automotive systems [16,22]. The microcontroller's key functions for this system include:

- Processing signals from the various sensors to estimate environmental conditions and vehicle status.
- Driving two Pulse Width Modulation-controlled servomotors: the KM-3318 metal-gear servomotor for steering and the KM-2013 for throttle and braking, providing reliable mechanical response.
- Implementing asynchronous serial communication via USART at 9600 baud with the FPGA board, which acts as the communication master, enabling bidirectional exchange of sensor data and control commands.

2.3.2. Sensors' Description

For the STM32 NUCLEO-F401RE board, three sensor types were integrated for comprehensive situational awareness:

1. TF-Luna LiDAR sensor: Mounted on the vehicle's front, this infrared laser sensor measures distances with high accuracy and a sampling rate up to 250 Hz. Its narrow detection angle (2° to 5°) provides precise object localization, making it well-suited for frontal detection tasks.
2. HC-SR04 ultrasonic sensors: Four ultrasonic sensors are positioned along the vehicle's sides to detect nearby objects within a 2 to 400 cm range. Operating at 40 kHz, these sensors calculate distances by measuring echo return times, with an effective detection angle less than 15°, ensuring focused and reliable readings.
3. A3144 Hall-effect sensor: Affixed to the transmission shaft, this sensor detects magnetic pulses corresponding to shaft rotation by measuring revolutions per minute (RPM).

It is important to mention that this sensor integration is common in autonomous vehicles and ADAS, optimizing sensor fusion for robust control [21].

2.3.3. USART Communication Protocol

To enable efficient data exchange between the STM32 microcontroller and the FPGA-based vision subsystem, a serial communication link was established using the Universal Asynchronous Receiver-Transmitter (UART) protocol. This protocol was selected due to its simplicity, low hardware overhead, and suitability for real-time embedded application.

A custom data frame structure was defined to ensure consistent and synchronized transmission of sensor data. Each transmitted frame consisted of eight comma-separated fields; to enable data identification, the FPGA recorded the incoming values using the following structured format: direction motor, acceleration motor, ultrasonic sensor 1, ultrasonic sensor 2, ultrasonic sensor 3, ultrasonic sensor 4, LiDAR sensor, and RPM velocity.

3. Results

This section details the design, integration, and experimental validation of the embedded vision and electronic control system for autonomous vehicles. It provides a comprehensive overview of the system architecture and prototype assembly, followed by the characterization of key subsystems, including the combined leverages FPGA-based image processing and STM32 microcontroller-based vehicle control, coordinated through serial

communication and supported by a graphical interface for immediate variables supervision. The results demonstrate the feasibility and performance of the proposed embedded vision solution in a controlled, scaled autonomous vehicle context.

3.1. Vision System Results

The embedded vision system implemented on the Kria KV260 successfully performed real-time lane segmentation and detection of vehicles, pedestrians, and lane markings. Lane markings were identified using a sliding window algorithm, while a YOLOv3 neural network handled the identification of these three classes. The YOLOv3 architecture, based on the Darknet-53 backbone, consists of 53 convolutional layers with skip connections and residual blocks, optimized for speed and accuracy in embedded systems. The network was fine-tuned using transfer learning from pre-trained weights on the COCO (Common Objects in Context) dataset and retrained on a custom dataset using a learning rate of 0.001, batch size of 16, and the Adam optimizer over 100 epochs.

The custom dataset consisted of approximately 4500 annotated images, with 1500 high-resolution images per class (vehicles, pedestrians, and lane markings), captured from real-world urban and semi-urban environments in Pachuca and San Agustín Tlaxiaca, Mexico. Images were resized to 416 $\times$ 416 pixels and annotated using bounding boxes in Roboflow, exported in YOLO format. The dataset was split into training (80%), testing (15%), and validation (5%) sets. Data augmentation techniques such as horizontal flipping, random brightness variation, and image scaling were applied to improve generalization.

This training process enabled the neural network to achieve a mean average precision (mAP) of 87.4% and an inference speed suitable for real-time deployment on FPGA hardware. Wireless transmission of results to a mobile device facilitated system validation and visual monitoring of segmented lanes and detected vehicles, pedestrians, and lane markings, reinforcing the embedded system's applicability to real-world ADAS applications.

To validate the performance of the system in conditions representative of real-world scenarios, a scaled testing environment was implemented. The testbed consisted of a 2 m wide cycle lane, clearly marked with solid and dashed lane delimiters, mimicking urban road geometries. This controlled environment enabled systematic experimentation under repeatable conditions, facilitating the assessment of lane detection accuracy; reliability in recognizing vehicles, pedestrians, and lane markings; and system responsiveness. The enclosed nature of the site also minimized external interference, providing a stable platform for iterative development and performance tuning of the ADAS prototype. In Figure 3, the system's image identification results can be observed.

The dataset size of 4500 images, with 1500 per class, is relatively limited compared to typical large-scale deep learning datasets. To mitigate this limitation, extensive data augmentation techniques were employed to enhance the model's generalization capability. Although the validation set comprised only 5% of the data, it was complemented by a dedicated 15% test set to ensure a comprehensive evaluation of the model's performance. Furthermore, the system's effectiveness was validated in a controlled, full-scale scaled testing environment replicating real-world road conditions, thereby extending the assessment beyond the limitations of the dataset. Future work will focus on expanding and diversifying the dataset with additional real-world samples to further improve the robustness and generalization capacity of the model.

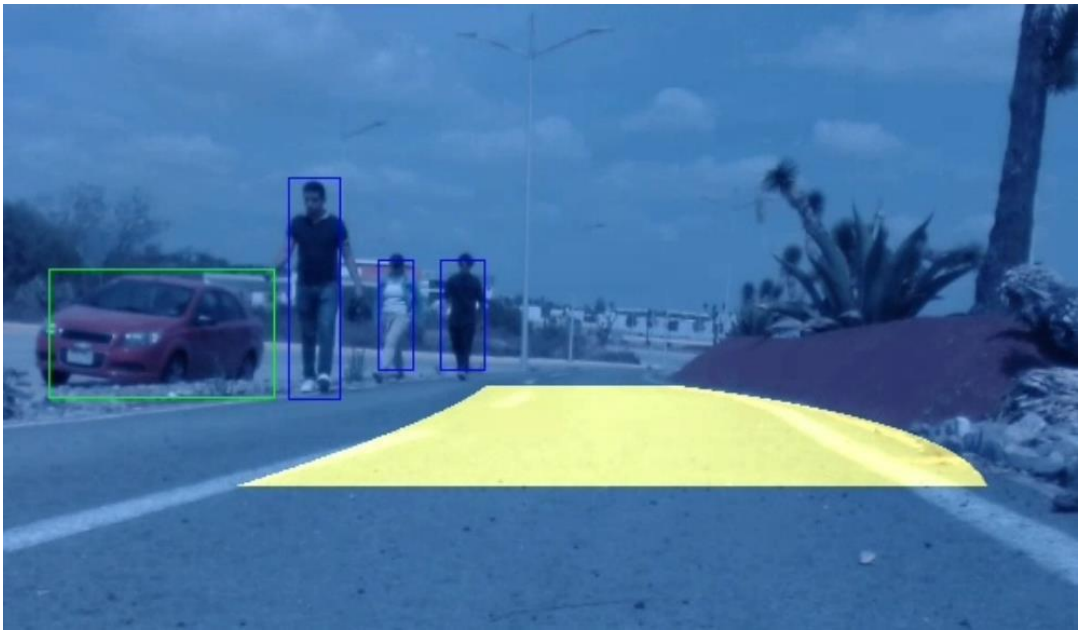


Figure 3. Pattern recognition results for people, cars, and lane lines.

The vision subsystem relies on a CMOS camera coupled to the FPGA for real-time image acquisition and processing. The performance of the detection model was found to be strongly influenced by ambient lighting conditions. While satisfactory detection rates were achieved in well-lit environments, reduced detection accuracy was observed in low-light scenarios, primarily due to the lower contrast of lane markings. To ensure stable illumination during initial testing, the camera was covered to control light exposure and minimize glare effects, enabling reliable image acquisition during daylight conditions, specifically between 9:00 a.m. and 5:00 p.m., under dry weather conditions. In this initial development stage, this strategy allowed consistent detection of lanes, pedestrians, and vehicles under controlled lighting conditions. In future stages, testing will be extended to varied illumination scenarios, including low-light and night-time environments, as well as the evaluation of alternative camera systems with enhanced dynamic range and sensitivity. These improvements are expected to expand the operational capabilities of the vision module and increase its robustness under real-world conditions.

Integration with the STM32-based control unit was verified through UART communication. Based on the detected lane position and object proximity, the FPGA sent structured decision data to the microcontroller to adjust motor speed and steering. This closed-loop interaction demonstrated the feasibility of real-time collaborative control between the vision and electronic control subsystems.

Performance testing on the scaled vehicle platform showed stable operation at speeds up to 25 km/h, with reliable visual detection at distances up to 6 m.

3.1.1. YOLOv3 and Segmentation Performance

To measure the performance of this computer vision system, the neural network and the lane detection algorithm were executed simultaneously for 5 min, recording the FPS (frames per second) value every 5 s to monitor the behavior of the network, as well as to observe its highest, lowest, and average peaks. Figure 4 shows the results obtained, allowing us to observe that, on average, the FPS value of the network remains at 3 FPS, with a minimum of 2.3 and a maximum of 3.7 at certain moments.

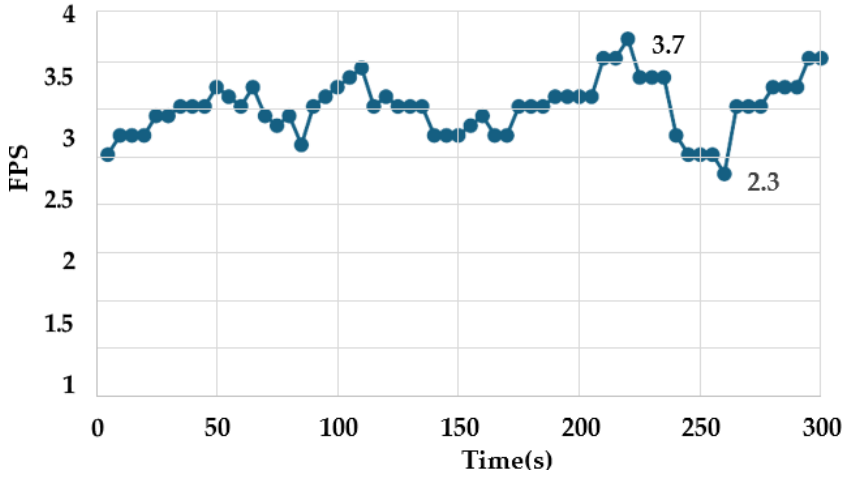


Figure 4. YOLOv3 and segmentation performance.

3.1.2. Segmentation Algorithm Accuracy

To evaluate the accuracy of the segmentation algorithm, two different metrics were used:

- Based on the distance between the edge detected by the segmentation algorithm versus the real edge.
- Based on the segmented area of the algorithm versus the real area.

For the first metric, calculation was performed using 80 images, considering the distance between corners of the segmented area, since the greatest discrepancies between the segmentation algorithm and the ground truth were observed in that region. The mean absolute difference was 33.6 pixels. Given the image resolution (640 × 480 pixels), this is considered an acceptable value for the developed algorithm. Finally, the maximum difference was 75 pixels, and the minimum was 5 pixels. Additionally, the percentage of vertices (Cj) from which the distance to the true segmentation contour is less than 20 pixels was calculated. This threshold was selected based on the mean absolute difference and visual inspection showing that such distance was small.

The results indicate that 16% of the points had a non-significant difference, according to the selected criteria. In Table 1, it is possible to notice the results of the segmentation based on distance.

Table 1. Distance-based segmentation results.

Metric	Value	Units
Mean Absolute Difference (MAD)	31.91	pixels
Maximum Difference (MAXDj)	75.11	pixels
Minimum Difference	5.00	pixels
Points with <20 px difference	16.00	%

Based on the segmented area of the algorithm compared to the actual area, the segmented surface from 80 images was used. Considering the surface segmented by the algorithm and the ground truth surface, the latter is defined as the correct area that follows the road path. The following areas were considered: true positive surface, false positive surface, false negative surface, and true negative surface. Using these values, the true positive rate, true negative rate, false positive rate, and true negative rate were calculated. From these, the metrics were derived, as shown in Table 2.

Table 2. Area-based segmentation results.

Metric	Value
True Positive Rate (TPR/FVP)	81.28%
True Negative Rate (TNR)	84.54%
False Positive Rate (FPR)	15.46%
False Negative Rate (FNR)	18.72%

The segmentation metrics reported in Table 2, derived from a pixel-wise comparison between the algorithm's output and the ground truth lane area over 80 images, demonstrate that the lane detection algorithm reliably identifies relevant lane pixels with a true positive rate (TPR) of 81.28% and a true negative rate (TNR) of 84.54%. This indicates a strong capability to distinguish lane markings from the background, effectively minimizing misclassification. In practical terms, in a real scenario, such segmentation accuracy could support stable lane tracking within the controlled test environment, enabling the vehicle to maintain lane position and execute basic navigational maneuvers. While some misclassifications occur, the system compensates for these through sensor fusion with LiDAR and ultrasonic data, which provide redundancy and contextual information to guide safe vehicle control. Therefore, despite segmentation errors, the combined sensor and control architecture ensures robust decision-making appropriate for the prototype's scale and intended use. These results highlight the platform's utility as a research and educational tool to evaluate and improve ADAS algorithms under realistic but safe operating conditions.

3.2. Electronic Control System Results

The electronic control subsystem was developed using an STM32 NUCLEO-F401RE microcontroller, responsible for managing the acquisition, processing, and transmission of sensor data, as well as the actuation of vehicle dynamics based on control signals. The system integrates five core components: a TF-Luna LiDAR sensor via I^2C for frontal detection, four HC-SR04 ultrasonic sensors for lateral and near-field distance monitoring, an A3144 Hall-effect sensor for wheel rotation measurement and speed estimation, and two servomotors—one dedicated to steering and the other to acceleration and braking.

The current architecture based on the STM32 board can efficiently handle up to four UART-based sensors if using hardware serial ports. However, if additional UART devices are added via software serial port (bit-banging), CPU overhead and timing constraints may introduce data loss or instability. For I^2C sensors like the LiDAR, multiple devices can be supported, as long as they have unique addresses, but the total bus bandwidth (100–400 kHz) can become a bottleneck at high sampling rates. Overall, no major processing bottlenecks are expected unless high-frequency data fusion is performed on a low-end STM32.

The TF-Luna LiDAR module demonstrated stable operation at up to a 250 Hz sampling frequency, effectively detecting in the 0.2 to 8 m range. Its narrow field of view (2°–5°) provided reliable frontal measurements, even under moderate ambient noise. The ultrasonic sensors operated with a precision of ± 0.3 cm within their 2–400 cm range and were calibrated to detect proximity asymmetries on both sides of the vehicle, assisting in lateral stabilization and cars or persons avoidance during motion.

For vehicle actuation, two high-torque servomotors were deployed. The KM-3318 servo, controlling the front axle, maintained directional stability with low angular error. The KM-2013 servo modulated throttle input, allowing the vehicle to shift between five discrete speed levels based on real-time feedback from the Hall-effect sensor. The latter was mounted on the transmission shaft and enabled continuous velocity monitoring through

pulse counting. Under test conditions, the system achieved a velocity resolution sufficient to maintain control accuracy within ± 0.5 km/h across all five speed levels.

All sensor data and actuation states were serialized into an 8-field structured UART frame, which was transmitted from STM32 to the Kria KV260 FPGA. The FPGA, functioning as the decision-making unit, parsed the incoming data to synchronize it with the visual information captured by the onboard camera system. This low-latency communication architecture ensured tight coupling between perception and control, enabling smooth transitions and responsive navigation behaviors in the scaled test vehicle.

Regarding sensor noise and signal interference, adjustments were made to the sensor acquisition timing to minimize measurement errors, particularly in the ultrasonic modules. Due to their working principle based on acoustic wave propagation, ultrasonic sensors are susceptible to signal overlaps and environmental reflections, especially when multiple sensors operate simultaneously. To address this, response times were staggered, and sequential triggering was implemented, ensuring that individual measurements did not interfere with each other. In contrast, LiDAR and Hall-effect sensors did not present significant interference issues, given their different physical measurement principles.

However, in the case of the Hall-effect sensor, mechanical vibrations from the vehicle chassis initially affected reading stability. This issue was mitigated through the installation of physical dampers and mounting isolation elements, which reduced signal fluctuations caused by vibrations during movement. These strategies contributed to improving the reliability of sensor data acquisition under operating conditions.

To enhance the usability and monitoring capabilities of the system, a custom graphical interface was developed. This interface receives the same UART frame, decodes it in real time, and presents both sensor data and actuator states to the user.

3.3. Graphical User Interface

To facilitate visualization and interaction with the embedded control system, a custom graphical user interface (GUI) was developed. This interface serves as both a real-time monitoring dashboard and a bidirectional control panel, enabling the user to visualize sensor data and simultaneously send actuation commands to the vehicle.

The GUI communicates wirelessly with the Kria KV260 FPGA via a dedicated Wi-Fi module. The FPGA, acting as an intermediary, forwards sensor data received from the STM32 microcontroller through UART and dispatches user-generated control commands back to the STM32 for real-time actuation. This bidirectional communication loop ensures synchronous coordination between visual processing, sensor monitoring, and user control. The interface decodes a structured data frame composed of eight comma-separated fields, with the order shown in Table 3.

Table 3. Description of the transmitted data frame with respect to its position within the structure.

Position	1	2	3	4	5	6	7	8
Data	Direction servo angle	A3144 Hall-effect Sensor	Ultrasonic 1	Ultrasonic 2	Ultrasonic 3	Ultrasonic 4	LiDAR	Velocity servo angle

The incoming frame is decoded using a custom parser within the FPGA, which extracts each field using the comma delimiter to identify start and end boundaries. These values are then transmitted to the GUI and displayed numerically and graphically, allowing the user to interpret the current state of the vehicle, including its orientation, motor throttle level, object proximity on all sides, frontal distance, and estimated speed in kilometers per hour.

In addition to visualization, the GUI includes control elements—such as sliders and directional inputs—that allow the user to modify the steering angle and throttle speed. These commands are encoded in a compatible format and sent from the GUI to the FPGA and then forwarded to the STM32 for execution via Pulse Width Modulation control of the corresponding servomotors.

The GUI enhances the operational usability of the system by providing an intuitive and efficient interface that bridges perception and actuation layers. It also supports debugging and performance evaluation by logging data during test runs. Figure 5 presents the developed interface, highlighting the layout of sensor readings, control and communication indicators, and real-time visualization for the vision system monitoring.

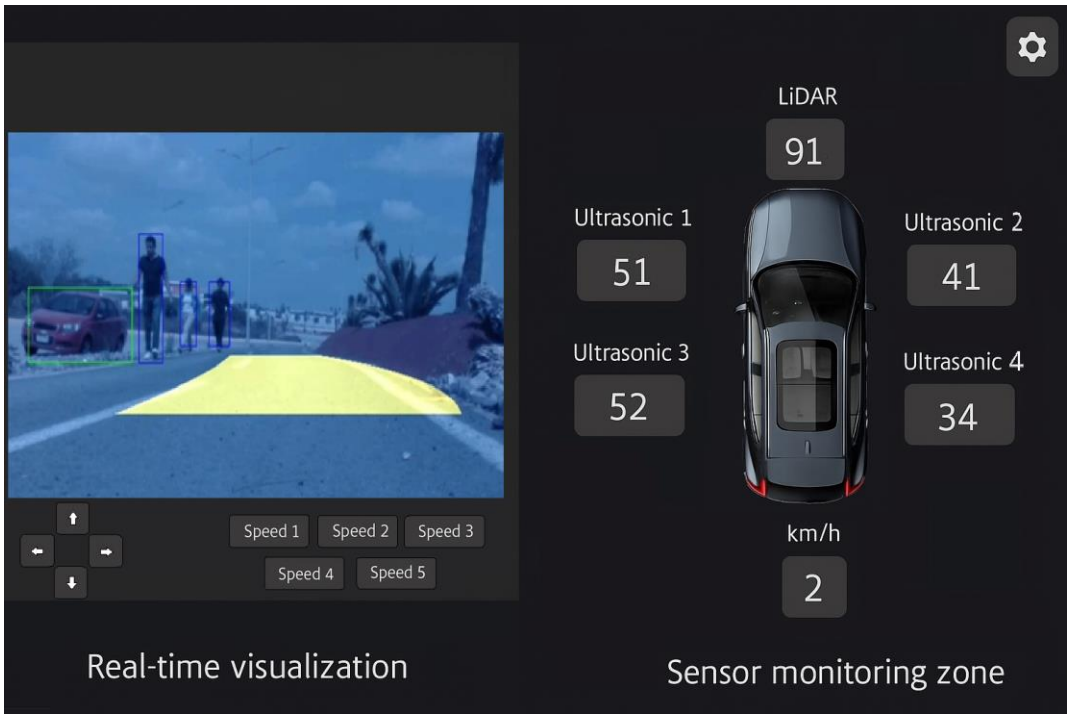


Figure 5. Graphical user interface frontend.

In the current development stage, system safety is primarily managed through operation within controlled environments and the availability of manual override via the graphical user interface (GUI). When inconsistent or unreliable sensor data are detected—such as communication loss or erratic readings—the system triggers an alert informing the operator of the anomaly and prompting a system reset if communication does not re-establish automatically. This basic safety mechanism is suitable for the current prototype phase, where testing occurs under supervised conditions. Future development will focus on implementing automated fault detection and recovery protocols, such as continuous monitoring of sensor status, data consistency checks, and automatic fallback strategies, to ensure safe operation without requiring human intervention, especially in more autonomous or unsupervised scenarios.

Figure 6 presents the final physical implementation of the system titled FPGA–STM32-Embedded Vision and Control Platform for ADAS Development on a 1:5 Scale Vehicle, in accordance with the functional design described in Figure 2. This configuration highlights the main components: the camera for image acquisition, the FPGA–STM32-based processing unit for real-time data analysis through the GUI, and the arrangement of sensors and actuators responsible for vehicle control. This physical representation demonstrates the integration of the various modules and subsystems that comprise the experimental

platform, developed for educational and research applications in the field of advanced driver assistance systems (ADAS).

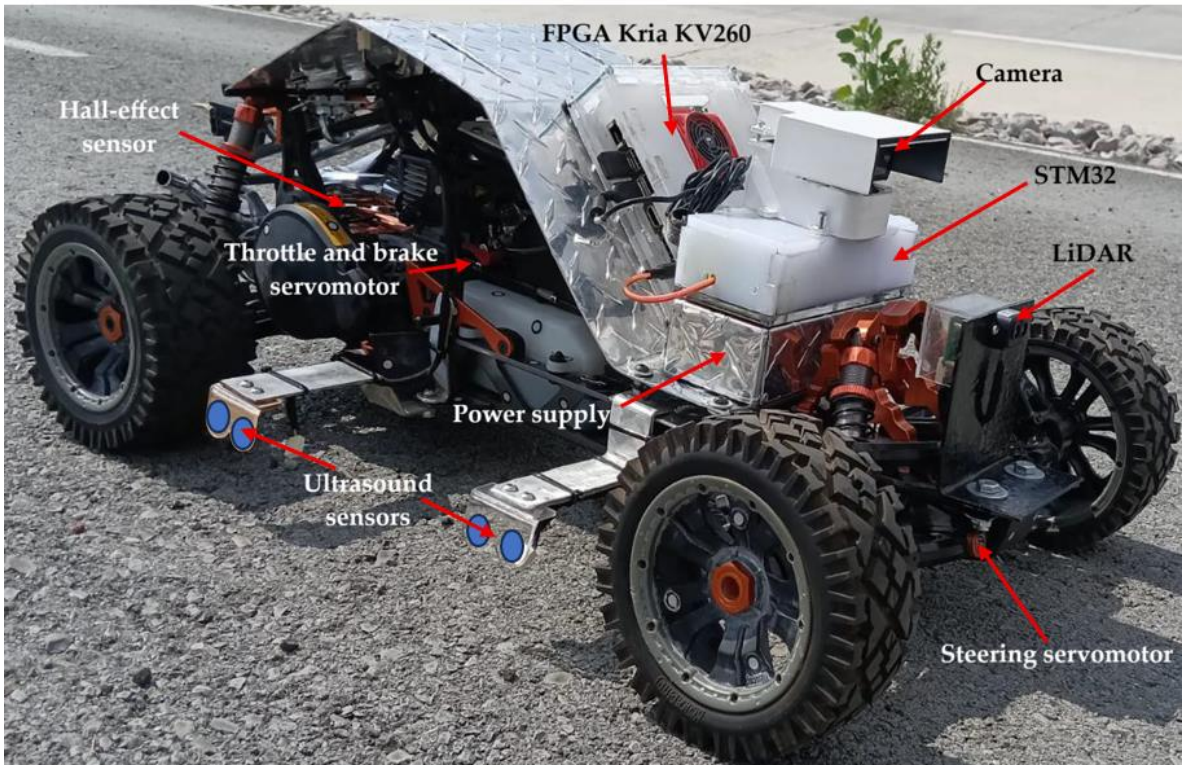


Figure 6. Final physical implementation of the FPGA–STM32-Embedded Vision and Control Platform for ADAS Development on a 1:5 Scale Vehicle.

4. Discussion

The embedded ADAS prototype presented in this work was comparatively analyzed against recent FPGA-based object detection systems reported in literature. These studies adopt diverse methodologies, including quantization, hardware parallelism, streaming architectures, and model compression techniques. Such a comparative framework allows for a robust evaluation of the technical advantages and trade-offs of the proposed system.

Implementations such as those in [23–25] employ optimized versions of YOLOv3 with quantization strategies like INT8 and 4W4A, achieving high inference rates up to 45 frames per second (FPS). However, these designs are primarily constrained to benchmarking detection models and do not incorporate control strategies or real-world actuator feedback. In contrast, the present work integrates object detection, lane segmentation, and vehicular actuation in a unified embedded platform, providing both perception and control capabilities in real-time.

Architectures described in [26–28] utilize modular streaming accelerators and dataflow-oriented pipelines, demonstrating high energy efficiency and scalability. Nonetheless, they often operate in static test environments with preprocessed datasets, limiting their validation under dynamic, real-world conditions. The system proposed herein includes on-board sensory feedback (ultrasonic, LiDAR, and Hall-effect sensors) and is validated through experimental deployment on a gasoline-powered 1:5 scale vehicle, highlighting its functional robustness in outdoor scenarios.

Long-term experiments were not conducted during this initial stage of system validation, as the primary objective was to establish a functional prototype and evaluate its performance under controlled conditions. That said, extended operational testing is planned as a key component of future work, focusing on system durability, sensor stability

over time, and component-level reliability under continuous operation. With respect to sensor drift, noise, and potential hardware failures, the current system architecture incorporates basic signal filtering and sequential acquisition strategies to minimize interference and improve data stability, particularly for ultrasonic modules. Sensor fusion using complementary sensors (LiDAR, ultrasonic, and Hall-effect) also contributes to mitigating individual sensor errors. Still, advanced fault-tolerant strategies, such as redundancy, sensor health monitoring, or adaptive recalibration methods, will be explored in subsequent development phases to enhance system robustness and long-term operational reliability.

Several studies [29–31] explore resource-efficient inference through weight quantization, filter pruning, and reuse of convolutional modules. While these methods yield significant improvements in throughput and silicon utilization, they may reduce detection accuracy in multi-class scenarios or variable lighting and occlusion conditions. In contrast, this work leverages a YOLOv3-Tiny model trained with a custom dataset acquired from local environments, thus ensuring higher relevance and adaptability to urban traffic contexts.

Although newer YOLO architectures such as YOLOv5, YOLOv7, and YOLOv8 offer improved accuracy and model efficiency, their integration within this study was constrained by hardware compatibility limitations. Specifically, the Xilinx Kria KV260 platform used for vision processing in this project supports pre-optimized hardware acceleration pipelines for a limited set of models. Among these, YOLOv3-Tiny is one of the few officially supported and fully deployable models with existing FPGA IP blocks and reference designs. This constraint strongly influenced the model selection process.

Despite being an earlier version, YOLOv3-Tiny remains highly relevant for embedded and real-time applications due to its lightweight architecture, low latency, and efficient hardware footprint, which are critical for resource-constrained systems. In our implementation, YOLOv3-Tiny enabled reliable people, car, and lane detection within the timing and power consumption requirements of the platform, validating its suitability for the project's objectives.

Nevertheless, the authors acknowledge that newer architectures optimized for edge devices (e.g., NanoDet, YOLOv5-N, MobileNet-SSD) offer promising improvements and should be explored in future works. Future versions of the platform will consider migrating to custom-accelerated pipelines or deploying on platforms with broader deep learning framework support to allow experimentation with more recent, high-performance models.

High-performance systems employing newer FPGA platforms and detection models—such as those described in [32–34]—demonstrate superior detection metrics and real-time throughput. However, their reliance on costly hardware (e.g., Xilinx ZCU104, Virtex-7) and proprietary IP cores increases development complexity and hinders replicability in academic settings. The current system, by comparison, prioritizes accessibility, using the Xilinx Kria KV260 board, sourced from AMD (formerly Xilinx, Inc.), headquartered in San José, California, United States, and STM32 microcontroller, sourced from STMicroelectronics, headquartered in Geneva, Switzerland to maintain a favorable cost–performance ratio suited for educational use and prototype development.

Concerning scalability for deploying newer iterations of YOLO, the current hardware platform, based on the Xilinx Kria KV260, presents specific limitations. The pre-optimized hardware acceleration pipelines available for this FPGA platform support only a restricted set of neural network architectures, including YOLOv3-Tiny. While the FPGA fabric theoretically offers computational resources to implement more advanced models, significant reconfiguration at the hardware description level would be necessary, which falls outside the intended scope of rapid prototyping and educational use. Consequently, migrating to alternative processing platforms or hybrid FPGA–CPU systems will be considered in

future development stages to support the deployment of more recent and computationally demanding detection architectures.

Additional contributions, such as those in [35–37], propose advanced techniques, including shared memory schemes, low-power neural execution, and reconfigurable processing pipelines. While these strategies optimize power consumption and silicon efficiency, they typically omit system-level integration with actuators and teleoperation interfaces. The proposed system addresses this gap by incorporating UART-based communication for sensor–actuator synchronization and offering a graphical user interface (GUI) for command issuance and data visualization.

Although the present system employs UART and I^2C communication protocols for data transfer between modules, which are relatively simple compared to the CAN (Controller Area Network) and LIN (Local Interconnect Network) buses used in full-scale automotive systems, this design decision enhances modularity and learning accessibility. The architecture remains scalable and conceptually consistent with industry standards, allowing students and researchers to understand and replicate critical interactions in vehicular communication systems. As such, the prototype is especially well-suited for academic environments where the emphasis is on foundational concepts, control strategies, and perception–actuation integration.

In comparison to existing FPGA-based ADAS implementations, this work stands out due to its focus on educational and prototype-level deployment using low-cost, modular architecture. For instance, the ADAS platform described in Sensors (2024) integrates a Xilinx ZCU104 FPGA to accelerate YOLOv3-based pavement defect detection, transmitting results via UART-to-CAN conversion for real-time communication [9]. While that study targets infrastructure inspection, its combination of FPGA vision processing and automotive interface exemplifies comparable integration strategies. Similarly, a study presented a dedicated SSD pedestrian detection accelerator implemented on a Zynq device, demonstrating high inference speed through network compression and hardware optimization [38]. In addition, a road segmentation system using LiDAR data and FPGA-based CNN inference achieved processing latencies under 17 ms for each scan, evidencing the viability of embedded LiDAR–CNN solutions [39]. The current platform differs by unifying computer vision (pedestrians, vehicles, and lanes), sensor fusion, and actuator control on a small-scale vehicle using Kria KV260 and STM32, thus bridging the gap between accessible academic prototypes and industrial ADAS frameworks.

In the work developed by Yecheng Lyu et al. [40], an optimized system is presented for real-time road segmentation using LiDAR data processed through convolutional neural networks (CNNs) implemented on an FPGA, achieving a latency of only 16.9 ms per scan and high accuracy validated with the KITTI dataset. In contrast, the system proposed in this paper introduces a modular, scalable, and low-cost ADAS platform that integrates computer vision, heterogeneous sensors, and embedded control, all implemented on a 1:5 scale vehicle with real-time monitoring and teleoperation capabilities. While the approach by Lyu et al. focuses on a specific task with high computational efficiency, the system presented here offers greater flexibility for experimentation and validation of multiple ADAS functionalities. Thus, both approaches are complementary: one prioritizes performance in a critical process for vehicle autonomy, and the other provides a versatile environment for applied research and practical training.

The work presented by Javier García López et al. [41] focuses on adapting a deep neural network based on VoxelNet for vehicle detection in LiDAR point clouds, leveraging FPGA technology to achieve real-time performance and high accuracy on complex datasets. In contrast, the proposed research develops a modular, scalable educational ADAS platform implemented on a 1:5 scale vehicle, integrating FPGA and microcontroller-based acceler-

ated processing and control, employing YOLOv3-Tiny for image-based object detection alongside multiple sensors. While the former emphasizes optimizing 3D detection on FPGA hardware, the latter offers a comprehensive hardware–software integration for rapid prototyping and practical training, albeit with inherent limitations due to the reduced scale of the prototype.

Compared with other FPGA-based detection systems reported in literature, the proposed platform prioritizes functionality integration over inference throughput. While it does not surpass high-end architecture in terms of speed or quantization efficiency, it offers significant advantages in terms of cost-effectiveness, didactic utility, and full-system integration. As such, the prototype serves as an effective foundation for ADAS experimentation, embedded systems training, and future research into autonomous mobility technologies.

In contrast to recent developments in autonomous vehicle technologies, the proposed system offers a modular and low-cost integration of real-time perception and control using an STM32 microcontroller and a Kria KV260 FPGA. Unlike the approach presented by Ahmad et al. [42], which focuses on the detection of small, biased GPS spoofing attacks using time-series analysis and inertial sensor fusion, our system does not rely on GNSS signals and is therefore inherently resilient to such threats. However, this design choice also limits its applicability in large-scale outdoor environments where satellite-based navigation is necessary. On the other hand, the review conducted by Abdollahi et al. [43] highlights the role of autonomous vehicles in pavement condition assessment and urban infrastructure monitoring. While their work provides a comprehensive overview of sensing technologies and analytical tools, it lacks the hardware-level integration and real-world validation demonstrated in our system. Accordingly, the proposed architecture bridges a relevant gap by delivering an experimentally validated solution that combines vision-based object detection, multisensor data acquisition, and actuation control. This system serves not only as a platform for ADAS experimentation but also as a scalable foundation for future autonomous applications. Nevertheless, future iterations may benefit from integrating secure positioning mechanisms and expanding sensing capabilities for infrastructure-oriented tasks.

Recent studies have demonstrated the potential of embedded devices optimized for real-time inference, such as the DNN model developed by Park et al. [44], which highlights the importance of reducing computational complexity to achieve efficiency without sacrificing accuracy. In this regard, the implementation of YOLOv3-Tiny on the platform achieves competitive precision with a mean average precision (mAP) of 87.4%, although the average frame rate (FPS) of 3 indicates room for improvement compared to systems targeting higher frequencies for real-time applications [45].

Regarding the specific functionality of lane and object detection and segmentation, the method based on sliding window segmentation and YOLOv3-Tiny detection shows robust results under controlled conditions, comparable to the effectiveness of commercial and infrastructure-based lane-centering technologies, as described by Kadav et al. [46]. Although infrastructure-based solutions may offer greater robustness under adverse conditions, the flexibility and lower cost of embedded and vision-based systems represent a significant advantage for research and educational scenarios. Additionally, the creation of a custom dataset with manual annotations from real urban environments strengthens the system's adaptability to specific contexts, a feature less emphasized in other works relying on standardized datasets [47].

Finally, the integration of bidirectional wireless communication and a graphical user interface (GUI) for real-time monitoring and teleoperation constitutes a practical and educational contribution that complements technical architecture. Unlike systems focusing

exclusively on hardware–software optimization for real-time detection, such as those presented by Zaharia et al. [48] and Sarvajcz et al. [45], the proposed approach incorporates a comprehensive framework that facilitates supervision and control in physical test platforms, a key requirement for training and experimental development in ADAS. However, sensitivity to varying lighting conditions and the low FPS suggest that future works should consider improvements in the visual acquisition system and algorithmic optimizations to extend operability in real-world and higher-speed scenarios, common challenges in current literature [45,47].

While the current validation was performed using a 1:5 scale vehicle under controlled experimental conditions, this approach inherently limits the generalizability of the results to real-world automotive environments. The use of a scaled platform was intended to provide a safe, low-cost, and flexible testing environment during the early stages of development. However, it is acknowledged that the current experimental validation is limited. The primary goal at this stage was to establish a low-cost, modular platform suitable for academic and prototype-level development, where safety and cost constraints necessitate initial testing in a simplified, scaled environment.

Nevertheless, extensive field testing under diverse environmental conditions—such as varying lighting, weather, and road textures—is necessary for a comprehensive assessment of the system's robustness and reliability. Expanding the testing scenarios to include both advanced simulation environments and real-world field conditions will allow for evaluation of the platform's performance in dynamic and uncontrolled settings.

Moreover, transitioning from scaled vehicles to full-scale platforms is identified as a key next step, where the modular and scalable nature of the proposed system will facilitate adaptation to larger testbeds. This progression will enable the assessment of the platform's applicability to industrial-level ADAS development, addressing current limitations in environmental diversity and operational complexity.

In summary, the proposed system distinguishes itself through its multi-layer integration of perception, control, and human interaction, facilitated by a hybrid FPGA–STM32 architecture and complemented by a wireless graphical interface. While it may not match specialized industrial systems in communication complexity or raw inference speed, it provides a versatile, low-cost, and scalable platform ideal for academic instruction, rapid prototyping, and experimental ADAS research. Its communication architecture, while simplified, is aligned with core automotive principles, making it a didactic tool that bridges conceptual understanding with practical system integration. Future development may incorporate higher-speed or automotive-standard buses, further quantization optimization, and adoption of newer detection models to enhance real-time performance and functionality.

5. Conclusions

This work presented the design, integration, and validation of an embedded Advanced Driver Assistance System (ADAS) implemented on a 1:5 scale gasoline-powered autonomous vehicle. The system combined an FPGA-based vision module with an STM32-based electronic control subsystem, supported by real-time sensor fusion, wireless communication, and a graphical interface for teleoperation and data monitoring.

The embedded vision system, implemented on the Kria KV260 board, successfully executed lane segmentation and object detection in real-time using a YOLOv3-Tiny model accelerated through hardware. Simultaneously, the STM32 microcontroller managed sensor acquisition—including ultrasonic, LiDAR, and Hall-effect modules—and actuation via Pulse Width Modulation-controlled servomotors. The communication between

both platforms was structured using a UART protocol, allowing synchronized perception and control.

Although the system utilizes UART and I^2C protocols, which differ from the CAN or LIN buses adopted in full-scale vehicles, its modularity and accessibility provide a valuable academic testbed that facilitates understanding of embedded vision, sensor fusion, and actuator control principles within a scalable and reconfigurable platform. Importantly, future development will focus on integrating automotive-standard communication protocols (CAN and LIN) to enhance compatibility with commercial vehicle electronic control units (ECUs) and enable seamless transition to full-scale applications.

Experimental results demonstrated the system's ability to process visual information, monitor the vehicle's environment, and perform basic autonomous maneuvers. The graphical user interface further enhanced usability, offering a real-time visualization of sensor values and actuator commands, as well as manual override capabilities for teleoperation.

Future work may focus on optimizing inference speed and reducing computational load through model pruning and quantization techniques, as well as deploying newer versions of YOLO or other state-of-the-art lightweight detection models better suited for edge devices. Additionally, integrating higher-speed and industry-standard communication protocols will be prioritized to better align with automotive requirements. These enhancements will support the platform's evolution from an academic prototype to a more robust and scalable system suitable for real-world ADAS applications.

Author Contributions: Conceptualization, J.D.R.-F., K.R.-T., D.A.F.-B., and M.H.-C.; software and data curation, J.D.R.-F., K.R.-T.,, D.A.F.-B., D.L.-M., A.A.-V., E.B.O.-G., L.C.-M., C.D.H.-L., C.T.-P., M.H.-C., D.A.C.-I. and J.A.A.-A.; formal analysis, J.D.R.-F., K.R.-T., D.A.F.-B., and M.H.-C.; investigation, J.D.R.-F., K.R.-T., D.A.F.-B., D.L.-M., A.A.-V., E.B.O.-G., L.C.-M., C.D.H.-L., C.T.-P., M.H.-C.; D.A.C.-I. and J.A.A.-A.; methodology, J.D.R.-F., K.R.-T., D.A.F.-B., and M.H.-C.; project administration, J.D.R.-F., K.R.-T., D.A.F.-B., and M.H.-C.; resources, J.D.R.-F., K.R.-T., D.A.F.-B., and M.H.-C.; supervision, J.D.R.-F., K.R.-T., D.A.F.-B., and M.H.-C.; validation, J.D.R.-F., K.R.-T., D.A.F.-B., D.A.C.-I. and J.A.A.-A.; visualization, J.D.R.-F., K.R.-T., M.H.-C., D.A.F.-B., A.A.-V., E.B.O.-G., L.C.-M., C.D.H.-L., C.T.-P., M.H.-C., D.A.C.-I. and J.A.A.-A.; writing—original draft, J.D.R.-F., K.R.-T., D.A.F.-B., and M.H.-C.; writing—review and editing, J.D.R.-F., K.R.-T., D.A.F.-B., D.L.-M., A.A.-V., E.B.O.-G., L.C.-M., C.D.H.-L., C.T.-P., M.H.-C., D.A.C.-I. and J.A.A.-A.; funding acquisition, K.R.-T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by grants from Iniciación en investigación a través de la ejecución de Proyectos de investigación científica y desarrollo tecnológico (SIP-IPN, 20254116) to Karen Ro-Tort.

Data Availability Statement: The original contributions presented in the study are included in the article; further inquiries can be directed at the corresponding authors.

Acknowledgments: We gratefully acknowledge the support of the Instituto Politécnico Nacional of Mexico for enabling the realization of this work. We also extend our sincere thanks to all individuals involved in the development of this project, as well as to the reviewers for their valuable comments and time dedicated to improving the manuscript. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ADAS	Advanced Driver Assistance System
CAN	Controller Area Network
FPGA	Field-Programmable Gate Array
LIN	Local Interconnect Network

LiDAR	Light Detection and Ranging
UART	Universal Asynchronous Receiver-Transmitter
YOLO	You Only Look Once

References

1. Ball, J.E.; Tang, B. Machine Learning and Embedded Computing in Advanced Driver Assistance Systems (ADAS). *Electronics* **2019**, *8*, 748. [CrossRef]
2. Wei, P.; Cagle, L.; Reza, T.; Ball, J.; Gafford, J. LiDAR and Camera Detection Fusion in a Real-Time Industrial Multi-Sensor Collision Avoidance System. *Electronics* **2018**, *7*, 84. [CrossRef]
3. Murendeni, R.; Mwanza, A.; Obagbuwa, I.C. Using a YOLO Deep Learning Algorithm to Improve the Accuracy of 3D Object Detection by Autonomous Vehicles. *World Electr. Veh. J.* **2025**, *16*, 9. [CrossRef]
4. Fang, N.; Li, L.; Zhou, X.; Zhang, W.; Chen, F. An FPGA-Based Hybrid Overlapping Acceleration Architecture for Small-Target Remote Sensing Detection. *Remote Sens.* **2025**, *17*, 494. [CrossRef]
5. Al Amin, R.; Hasan, M.; Wiese, V.; Obermaisser, R. FPGA-Based Real-Time Object Detection and Classification System Using YOLO for Edge Computing. *IEEE Access* **2024**, *12*, 102345–102357. [CrossRef]
6. He, Z.; Huang, W.; Gao, J.; Liu, Y.; Huang, Y.; Zhou, Y. Design and Implementation of a YOLOv4-tiny Hardware Accelerator with PYNQ-Z1. In Proceedings of the 2025 6th International Conference on Electrical, Electronic Information and Communication Engineering (EEICE), Shenzhen, China, 18–20 April 2025. [CrossRef]
7. Pérez Fernández, J.; Alcázar Vargas, M.; Velasco García, J.M.; Cabrera Carrillo, J.A.; Castillo Aguilar, J.J. Low-Cost FPGA-Based Electronic Control Unit for Vehicle Control Systems. *Sensors* **2019**, *19*, 1834. [CrossRef]
8. Velez, G.; Cortés, A.; Nieto, M.; Vélez, I.; Otaegui, O. A Reconfigurable Embedded Vision System for Advanced Driver Assistance. *J. Real-Time Image Process.* **2015**, *10*, 725–739. [CrossRef]
9. Chi, T.-K.; Chen, T.-Y.; Lin, Y.-C.; Lin, T.-L.; Zhang, J.-T.; Lu, C.-L.; Chen, S.-L.; Li, K.-C.; Abu, P.A.R. An Edge Computing System with AMD Xilinx FPGA AI Customer Platform for Advanced Driver Assistance System. *Sensors* **2024**, *24*, 3098. [CrossRef]
10. Mata-Carballeira, Ó.; Gutiérrez-Zaballa, J.; del Campo, I.; Martínez, V. An FPGA-Based Neuro-Fuzzy Sensor for Personalized Driving Assistance. *Sensors* **2019**, *19*, 4011. [CrossRef]
11. Dendaluce Jahnke, M.; Cosco, F.; Novickis, R.; Pérez Rastelli, J.; Gomez-Garay, V. Efficient Neural Network Implementations on Parallel Embedded Platforms Applied to Real-Time Torque-Vectoring Optimization Using Predictions for Multi-Motor Electric Vehicles. *Electronics* **2019**, *8*, 250. [CrossRef]
12. Singh, S.; Shekhar, C.; Vohra, A. FPGA-Based Real-Time Motion Detection for Automated Video Surveillance Systems. *Electronics* **2016**, *5*, 10. [CrossRef]
13. Vasile, C.-E.; Ulmămei, A.-A.; Bîră, C. Image Processing Hardware Acceleration—A Review of Operations Involved and Current Hardware Approaches. *J. Imaging* **2024**, *10*, 298. [CrossRef]
14. Zhai, J.; Li, B.; Lv, S.; Zhou, Q. FPGA-Based Vehicle Detection and Tracking Accelerator. *Sensors* **2023**, *23*, 2208. [CrossRef]
15. Biglari, A.; Tang, W. A Review of Embedded Machine Learning Based on Hardware, Application, and Sensing Scheme. *Sensors* **2023**, *23*, 2131. [CrossRef]
16. Tseng, Y.-T.; Ferng, H.-W. An Improved Traffic Rerouting Strategy Using Real-Time Traffic Information and Decisive Weights. *IEEE Trans. Veh. Technol.* **2021**, *70*, 9741–9751. [CrossRef]
17. Liu, J.Y.; Sun, S.Y. Design and Implementation of Servo Motor Speed Control System Based on STM32. *Appl. Mech. Mater.* **2013**, *427–429*, 999–1002. [CrossRef]
18. Lyu, Y.; Bai, L.; Huang, X. ChipNet: Real-Time LiDAR Processing for Drivable Region Segmentation on an FPGA. *arXiv* **2018**, arXiv:1808.03506. [CrossRef]
19. Nguyen, D.-D.; Nguyen, D.-T.; Le, M.-T.; Nguyen, Q.-C. FPGA-SoC Implementation of YOLOv4 for Flying-Object Detection. *J. Real-Time Image Process.* **2024**, *21*, 63. [CrossRef]
20. Atibi, M.; Benrabh, M.; Atouf, I.; Boussaa, M.; Bennis, A. Implementation of a Vehicle Detection System in the FPGA Embedded Platform. *J. Theor. Appl. Inf. Technol.* **2017**, *95*, 5746–5755.
21. Xie, X.; Wei, H.; Yang, Y. Real-Time LiDAR Point-Cloud Moving Object Segmentation for Autonomous Driving. *Sensors* **2023**, *23*, 547. [CrossRef]
22. Pamula, W. Vehicle Detection Algorithm for FPGA Based Implementation. In *Computer Recognition Systems 3*; Kurzynski, M., Wozniak, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; Volume 57, pp. 681–690. [CrossRef]
23. Zhao, D.; Liu, S.; Zhang, Z.; Zhang, Z.; Tang, L. Research on ZYNQ Neural Network Acceleration Method for Aluminum Surface Microdefects. *Digit. Signal Process.* **2025**, *157*, 104900. [CrossRef]
24. Yang, X.; Zhuang, C.; Feng, W.; Yang, Z.; Wang, Q. FPGA Implementation of a Deep Learning Acceleration Core Architecture for Image Target Detection. *Appl. Sci.* **2023**, *13*, 4144. [CrossRef]

25. Adiono, T.; Putra, A.; Sutisna, N.; Syafalni, I. Low Latency YOLOv3-Tiny Accelerator for Low-Cost FPGA Using General Matrix Multiplication Principle. *IEEE Access* **2021**, *9*, 130102–130114. [CrossRef]
26. Kumar, M.; Niharika, A.; Reddy, K.A.; Gupta, H.; Pushpalatha, K.N. Advanced Driver Assistance System (ADAS) on FPGA. SSRG Int. *J. VLSI Signal Process.* **2023**, *10*, 22–26. [CrossRef]
27. Günay, B.; Okcu, S.B.; Bilge, H.Ş. LPYOLO: Low Precision YOLO for Face Detection on FPGA. In Proceedings of the 2022 International Conference on Reconfigurable Computing and FPGAs (ReConFig), Prague, Czech Republic, 28–30 July 2022. [CrossRef]
28. Montgomerie-Corcoran, A.; Toupas, P.; Yu, Z.; Bouganis, C.-S. SATAY: A Streaming Architecture Toolflow for Accelerating YOLO Models on FPGA Devices. *arXiv* **2023**, arXiv:2309.01587. [CrossRef]
29. Yu, L.; Zhu, J.; Zhao, Q.; Wang, Z. An Efficient YOLO Algorithm with an Attention Mechanism for Vision-Based Defect Inspection Deployed on FPGA. *Micromachines* **2022**, *13*, 1058. [CrossRef]
30. Ding, C.; Wang, S.; Liu, N.; Xu, K.; Wang, Y.; Liang, Y. REQ-YOLO: A Resource-Aware Efficient Quantization Framework for Object Detection on FPGAs. *arXiv* **2019**, arXiv:1909.00745. [CrossRef]
31. Sharma, S.; Guleria, K.; Tiwari, S.; Kumar, S. An Enhanced DarkNet53-based YOLOv3 Feature Pyramid Network for Real-Time Object Detection. In Proceedings of the 2024 International Conference on Computational Intelligence and Computing Applications (ICCICA), Samalkha, India, 23–24 May 2024; pp. 148–153. [CrossRef]
32. Zhang, F.; Zhu, X.; Wang, H.; Li, Z. Apply YOLOv4-Tiny on an FPGA-Based Accelerator of CNN for Object Detection. *J. Phys. Conf. Ser.* **2022**, *2303*, 012032. [CrossRef]
33. Wang, Y.; Han, Y.; Chen, J.; Wang, Z.; Zhong, Y. An FPGA-Based Hardware Low-Cost, Low-Consumption Target-Recognition and Sorting System. *World Electr. Veh. J.* **2023**, *14*, 245. [CrossRef]
34. Sha, X.; Yanagisawa, M.; Shi, Y. An FPGA-Based YOLOv6 Accelerator for High-Throughput and Energy-Efficient Object Detection. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **2024**, *E108.A*, 392–400. [CrossRef]
35. Wang, Y.; Liao, Y.; Yang, J.; Wang, H.; Zhao, Y.; Zhang, C.; Xiao, B.; Xu, F.; Gao, Y.; Xu, M.; et al. An FPGA-Based Online Reconfigurable CNN Edge Computing Device for Object Detection. *Microelectron. J.* **2023**, *137*, 105805. [CrossRef]
36. Zhang, S.; Cao, J.; Zhang, Q.; Zhang, Q.; Zhang, Y.; Wang, Y. An FPGA-Based Reconfigurable CNN Accelerator for YOLO. In Proceedings of the IEEE International Conference on Electronics Technology, Chengdu, China, 8–12 May 2020; pp. 108–113. [CrossRef]
37. Lin, J.; Yu, W.; Yang, X.; Zhao, P.; Zhang, H.; Zhao, W. An Edge Computing Based Public Vehicle System for Smart Transportation. *IEEE Trans. Veh. Technol.* **2020**, *69*, 12635–12651. [CrossRef]
38. Falaschetti, L.; Manoni, L.; Palma, L.; Pierleoni, P.; Turchetti, C. Embedded Real-Time Vehicle and Pedestrian Detection Using a Compressed Tiny YOLO v3 Architecture. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 19399–19409. [CrossRef]
39. Chiang, C.-Y.; Chen, Y.-L.; Ke, K.-C.; Yuan, S.-M. Real-Time Pedestrian Detection Technique for Embedded Driver Assistance Systems. In Proceedings of the 2015 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 9–12 January 2015; pp. 206–207. [CrossRef]
40. Lyu, Y.; Bai, L.; Huang, X. Real-Time Road Segmentation Using LiDAR Data Processing on an FPGA. In Proceedings of the 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy, 27–30 May 2018; pp. 1–5.
41. García López, J.; Agudo, A.; Moreno-Noguer, F. 3D Vehicle Detection on an FPGA from LiDAR Point Clouds. In Proceedings of the 2019 2nd International Conference on Watermarking and Image Processing (ICWIP 2019), Marseille, France, 18–20 September 2019; pp. 21–26. [CrossRef]
42. Ahmad, M.; Youssef, A.; Islam, R.; Satam, P. Detection of Multiple Small Biased GPS Spoofing Attacks on Autonomous Vehicles Using Time-Series Analysis. In Proceedings of the IEEE 4th International Conference on AI in Cybersecurity (ICAIC), Houston, TX, USA, 5–7 February 2025; pp. 12–19. [CrossRef]
43. Abdollahi, C.; Mollajafari, M.; Golroo, A.; Moridpour, S.; Wang, H. A Review on Pavement Data Acquisition and Analytics Tools Using Autonomous Vehicles. *Road Mater. Pavement Des.* **2024**, *25*, 914–940. [CrossRef]
44. Park, J.; Aryal, P.; Mandumula, S.R.; Asolkar, R.P. An Optimized DNN Model for Real-Time Inferencing on an Embedded Device. *Sensors* **2023**, *23*, 3992. [CrossRef] [PubMed]
45. Sarvajcz, K.; Ari, L.; Menyhart, J. AI on the Road: NVIDIA Jetson Nano-Powered Computer Vision-Based System for Real-Time Pedestrian and Priority Sign Detection. *Appl. Sci.* **2024**, *14*, 1440. [CrossRef]
46. Kadav, P.; Sharma, S.; Fanas Rojas, J.; Patil, P.; Wang, C.R.; Ekti, A.R.; Meyer, R.T.; Asher, Z.D. Automated Lane Centering: An Off-the-Shelf Computer Vision Product vs. Infrastructure-Based Chip-Enabled Raised Pavement Markers. *Sensors* **2024**, *24*, 2327. [CrossRef]

47. Chuprov, S.; Belyaev, P.; Gataullin, R.; Reznik, L.; Neverov, E.; Viksnin, I. Robust Autonomous Vehicle Computer-Vision-Based Localization in Challenging Environmental Conditions. *Appl. Sci.* **2023**, *13*, 5735. [CrossRef]
48. Zaharia, C.; Popescu, V.; Sandu, F. Hardware–Software Partitioning for Real-Time Object Detection Using Dynamic Parameter Optimization. *Sensors* **2023**, *23*, 4894. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.