DESDE 2013
https://repository.uaeh.edu.mx/revistas/index.php/icbi/issue/archive
Pädi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI
ISSN: 2007-6363

Publicación Semestral Pädi Vol. 14 No. Especial (2026) 344-349

Sistema de visión embebido basado en FPGA para detección de objetos y carriles en vehículo a escala 1:5
FPGA-Based Embedded Vision System for Object and Lane Detection in a 1:5 Scale Vehicle

K. Roa-Tort [a,*], J.D. Rivera-Fernández [a], D.A. Fabila-Bustos [a], D. León-Martínez [a], A. Apolonio-Vera [a], E.B. Ortega-Gutierréz [a], D.A. Cano-Ibarra [a]

[a] *Laboratorio de Optomecatrónica y Energías, UPIIH, Instituto Politécnico Nacional, Distrito de Educación, Salud, Ciencia, Tecnología e Innovación, San Agustín Tlaxiaca 42162, Hidalgo, México.*

Resumen

Este trabajo presenta el desarrollo e implementación de un sistema de visión embebido basado en FPGA para la detección de carriles y objetos en un vehículo autónomo a escala 1:5. Se utilizó la tarjeta *Kria KV260 Vision AI* para procesar una red YOLOv3-Tiny y un algoritmo de segmentación de carriles en tiempo real. Las imágenes fueron adquiridas mediante una cámara USB y los resultados se visualizaron en una interfaz gráfica inalámbrica desarrollada para un dispositivo Android. El sistema alcanzó una precisión media del 92.4 % en la detección de vehículos y presentó una latencia menor a 150 ms. La estructura mecánica con aislamiento de vibraciones mejoró significativamente la calidad de imagen. Entre las limitaciones destacan la dependencia de condiciones de iluminación y la resolución de la cámara utilizada. La plataforma es modular, eficiente y adecuada para aplicaciones académicas y prototipos ADAS de bajo costo. Su diseño permite futuras mejoras, como la integración de sensores adicionales o protocolos de comunicación automotriz, consolidándose como una herramienta versátil para investigación y enseñanza en sistemas embebidos de percepción vehicular.

Palabras Clave: Sistemas ADAS, Visión artificial, FPGA, Asistencia a la conducción segura, Percepción visual del entorno vial.

Abstract

This work presents the development and implementation of an embedded vision system based on FPGA for lane and object detection in a 1:5 scale autonomous vehicle. The Kria KV260 Vision AI board was used to process a YOLOv3-Tiny network and a real-time lane segmentation algorithm. Images were acquired via a USB camera, and the results were displayed on a wireless graphical interface developed for an Android device. The system achieved an average accuracy of 92.4% in vehicle detection and a latency below 150 ms. The mechanical structure with vibration isolation significantly improved image quality. Limitations include dependency on lighting conditions and camera resolution. The platform is modular, efficient, and suitable for academic applications and low-cost ADAS prototypes. Its design allows for future enhancements such as the integration of additional sensors or automotive communication protocols, establishing it as a versatile tool for research and teaching in embedded vehicular perception systems.

Keywords: ADAS systems; Computer vision, FPGA, Safe driving assistance, Visual perception of the road environment.

1. Introducción

El desarrollo de los Sistemas Avanzados de Asistencia al Conductor (ADAS, por sus siglas en inglés) se ha convertido en un objetivo fundamental en la investigación automotriz, con el propósito de mejorar la seguridad vial, la eficiencia del tráfico y el confort del conductor mediante funciones como la advertencia de salida de carril, detección de obstáculos, reconocimiento de peatones y control de crucero adaptativo (Ball & Tang, 2019; Wei et al., 2018).

Entre las diversas modalidades de sensores, los sistemas basados en visión destacan especialmente por su capacidad de percepción contextual, que incluye la identificación de marcas viales, detección de señales de tráfico y seguimiento de

*Autor para la correspondencia: kroat@ipn.mx
Correo electrónico: kroat@ipn.mx (Karen Roa-Tort), jdriveraf@ipn.mx (Josué D. Rivera -Fernández), dfabilab@ipn.mx (Diego A. Fabila-Bustos), dleonm1902@alumno.ipn.mx (Daniel León-Martínez), aapoloniov1900@alumno.ipn.mx (Adrián Apolonio-Vera), eortegag1600@alumno.ipn.mx (Elizama B. Ortega-Gutiérrez), dcanoi2000@alumno.ipn.mx (David A. Cano-Ibarra).

Historial del manuscrito: recibido el 19/07/2025, última versión-revisada recibida el 26/11/2025, aceptado el 26/11/2025, publicado el 25/03/2026. **DOI:** https://doi.org/10.29057/icbi.v14iEspecial.15463

obstáculos en tiempo real (Murendeni et al., 2025; Wei et al., 2018).

No obstante, la incorporación de visión por computadora en sistemas automotrices conlleva desafíos importantes relacionados con la demanda computacional, el consumo energético y la latencia.

Si bien las soluciones tradicionales que utilizan GPUs o CPUs multinúcleo ofrecen una capacidad de procesamiento considerable, suelen presentar un alto consumo energético y una latencia no determinista, lo que las hace menos adecuadas para plataformas automotrices embebidas con restricciones (Wei et al., 2018; Fang et al., 2025).

La visión por computadora embebida basada en FPGA representa una alternativa prometedora, al permitir la implementación de canales de procesamiento en paralelo, latencia determinista y una operación con alta eficiencia energética (Murendeni et al., 2025; Al Amin et al., 2024).

Los ADAS dependen en gran medida de capacidades de percepción y toma de decisiones en tiempo casi real para garantizar la seguridad del conductor y mejorar las funcionalidades autónomas. Entre las diversas tecnologías utilizadas para la percepción del entorno, la visión por computadora destaca por su capacidad para detectar carriles, vehículos y peatones en condiciones dinámicas. Sin embargo, las plataformas de cómputo embebido tradicionales suelen enfrentar desafíos relacionados con la latencia de procesamiento y la eficiencia energética.

Los Arreglos de Puertas Programables en Campo (FPGAs) ofrecen una alternativa convincente al permitir paralelismo a nivel de hardware, inferencia de baja latencia y bajo consumo de energía.

Este trabajo presenta el desarrollo, integración y validación de un sistema de visión embebido basado en FPGA, diseñado para un vehículo a escala 1:5 con motor de gasolina. La plataforma emplea la tarjeta Xilinx Kria KV260 Vision AI para el procesamiento acelerado de una red neuronal YOLOv3-Tiny y un algoritmo de segmentación de carriles. Además, se desarrolló una interfaz gráfica personalizada para la visualización de resultados a través de transmisión inalámbrica.

En resumen, este trabajo tuvo como objetivo específico diseñar una plataforma modular y escalable que permita futuras mejoras, como la integración de sensores adicionales o protocolos de comunicación automotriz, sirviendo como herramienta académica y experimental para el desarrollo de funcionalidades ADAS basadas en visión, evaluando cuantitativamente el desempeño del sistema mediante métricas como precisión media (mAP), latencia y tasas de verdaderos y falsos positivos en la detección de carriles y objetos al momento utilizando FPGA de bajo consumo y latencia determinista.

El sistema propuesto sirve como una plataforma efectiva tanto para propósitos académicos como experimentales en el desarrollo de funcionalidades ADAS basadas en visión, enfocándose exclusivamente en los subsistemas de percepción y visualización.

2. Materiales y métodos.

El prototipo del sistema fue implementado en un vehículo a escala 1:5 propulsado por un motor de gasolina de dos tiempos. La plataforma comprende tres componentes principales: un módulo de adquisición de imágenes, una unidad de procesamiento embebido de visión y una interfaz gráfica de usuario (GUI) para monitoreo remoto. El sistema está diseñado para realizar detección y análisis de carriles y obstáculos, transmitiendo los datos visuales de forma inalámbrica para su inspección por parte del usuario. La Figura 1 muestra un diagrama a bloques general del sistema, donde se identifican sus áreas funcionales.

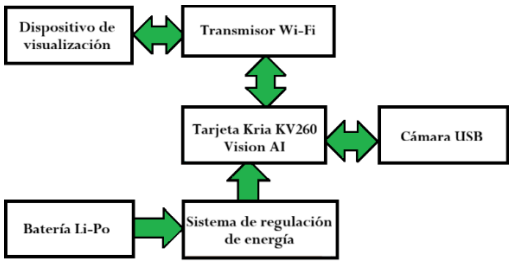

Figura 1. Diagrama a bloques general del sistema.

A partir de la Figura 1, se puede identificar que el sistema comprende tres módulos esenciales:

- Arquitectura del hardware: comprende el sistema de alimentación y regulación de energía, así como los componentes físicos del vehículo y del sistema embebido.
- Procesamiento embebido de visión artificial: incluye la tarjeta FPGA Kria KV260 Vision AI junto con la cámara USB para adquisición y procesamiento de imágenes.
- Interfaz gráfica de usuario (GUI): permite la transmisión de datos desde el vehículo y la visualización de resultados en un dispositivo móvil.

2.1. Arquitectura del hardware.

El núcleo del procesamiento de visión está basado en la tarjeta Xilinx Kria KV260 Vision AI Starter Kit, que combina un procesador ARM con lógica programable para cómputo paralelo. Una cámara USB Logitech C920 Pro-HD fue montada al frente del vehículo para capturar video a 30 FPS. La alimentación se realizó mediante una batería Li-Po regulada, garantizando operación estable a 12 V con protección de circuito.

El montaje físico se realizó con una estructura de soporte en aluminio fabricada mediante sistemas de control numérico (CNC), incorporando amortiguadores de vibración como arandelas de goma y cinta de poliuretano para aislar los componentes electrónicos de las vibraciones generadas por el motor. La Figura 2 muestra la instalación del sistema físico en el vehículo.

Figura 2. Vehículo a escala 1:5 con sistema embebido incorporado.

2.2. Procesamiento de visión embebido.

La señal de la cámara se procesa en la FPGA utilizando una red neuronal YOLOv3-Tiny para detección de objetos y un algoritmo de ventana deslizante para segmentación de carriles. Se empleó la biblioteca PYNQ-DPU para acelerar las inferencias en la lógica programable del FPGA.

Se recolectaron 1,500 imágenes etiquetadas de entornos urbanos locales, seleccionadas para cubrir diferentes condiciones de iluminación y ángulos de cámara, y se procesaron en Roboflow.

La selección de imágenes para el dataset del proyecto ADAS se realizó considerando escenarios representativos del entorno del vehículo, incluyendo distintos tipos de carriles, condiciones de iluminación y presencia de peatones u obstáculos. Se priorizó la diversidad de situaciones para mejorar la capacidad de generalización del sistema, evitando imágenes repetitivas o poco relevantes. Cada imagen fue etiquetada manualmente según las clases de interés (carriles, vehículos, peatones), asegurando que el dataset fuera balanceado y adecuado para entrenar y evaluar los algoritmos de detección y segmentación empleados.

El entrenamiento se realizó en Google Colab con una GPU Tesla T4 mediante aprendizaje por transferencia desde un modelo YOLOv3-Tiny preentrenado. Tras 1,200 iteraciones, los pesos entrenados fueron implementados en la Kria para inferencia en tiempo real flexible.

El proceso de la información básicamente se divide en dos etapas:

En la primera, la Kria KV260 obtiene captura de video proveniente de la cámara, y mediante el algoritmo de ventanas deslizantes realiza la segmentación para detectar su carril dentro de la carretera.

Como segunda parte del proceso, para la detección de automóviles y personas, se agregó el archivo generado por el entrenamiento de la red neuronal YOLOv3, que contiene la información de los pesos y etiquetas generados tras dicho entrenamiento con el *dataset* generado y haciendo uso de la técnica de *Transfer Learning.*

El código desarrollado, implementado en Pynq, realiza tanto la segmentación de la carretera, así como el reconocimiento, al tiempo que, haciendo uso de la librería Pynq-dpu es capaz de utilizar los recursos de la FPGA montada en la tarjeta para el procesamiento de las imágenes y de esta forma mejorar la velocidad en la detección de objetos.

Tras realizar dicho procesamiento, los resultados se envían de forma inalámbrica al dispositivo móvil mediante Wi-Fi y se muestran dentro del entorno de visualización desarrollado.

2.3. Interfaz Gráfica de Usuario (GUI).

Para observar los resultados del sistema de visión artificial, se desarrolló una aplicación móvil con el software Android Studio. La aplicación desarrollada permite la comunicación Wi-Fi entre la Kria KV260 y el dispositivo móvil en el que se encuentre corriendo la aplicación. Se utilizó una tableta modelo Samsung S6 lite con la versión de Android 13 para realizar las pruebas de funcionamiento y validación.

Una vez que el sistema de visión artificial se encuentra encendido y el programa desarrollado en ejecución, éste crea un punto de acceso Wi-Fi al que se debe conectar el dispositivo en donde se encuentre instalada la aplicación. Este proceso tardo menos de 150 ms para el enlace entre el sistema físico y la aplicación mediante comunicación Wi-Fi.

Posteriormente, se ejecuta la aplicación y en la pantalla se muestra la imagen que ha sido capturada y procesada por el sistema de visión, en la Figura 3 se presenta una imagen de las pruebas de funcionamiento en la cual se encuentra corriendo el sistema y realizando la detección de automóviles y personas, pero con la cámara montada sobre un auto de tamaño real.

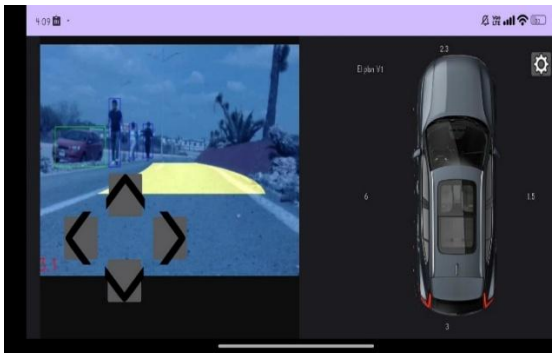

Figura 3. Vista principal de la interfaz gráfica de usuario.

Resumiendo, la metodología empleada, el flujo del procesamiento consta de cuatro etapas:

- Adquisición de imágenes: la cámara USB captura video en tiempo real desde el entorno del vehículo.
- Preprocesamiento: las imágenes se normalizan y ajustan para la resolución requerida por la red neuronal YOLOv3-Tiny y el algoritmo de segmentación de carriles.
- Procesamiento embebido:
  - Detección de carriles: mediante algoritmo de ventana deslizante se identifica la posición del carril en la carretera.
  - Detección de objetos: la FPGA ejecuta inferencia sobre la red YOLOv3-Tiny utilizando los pesos entrenados para identificar vehículos y peatones.
- Visualización y transmisión: los resultados se envían mediante Wi-Fi al dispositivo móvil, donde la GUI los muestra al usuario en tiempo real.

El proceso puede visualizarse gráficamente en el diagrama a bloques de la Figura 4.

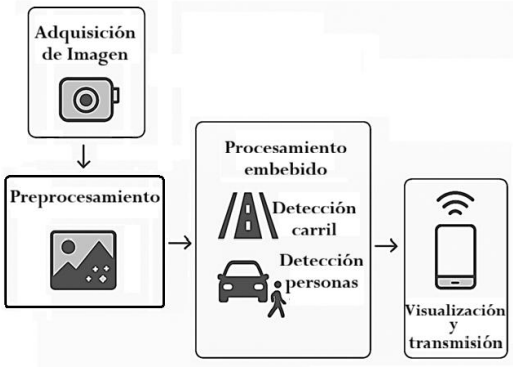

Figura 4. Diagrama esquemático de la metodología.

3. Resultados

Con el propósito de simular un ambiente adecuado a las dimensiones a escala del auto, se hizo uso de las instalaciones de la ciclo pista que se encuentra dentro del Distrito de Educación, Salud, Ciencia, Tecnología e Innovación San Agustín Tlaxiaca, pues se trata de una carretera de asfalto de 2 metros de ancho (escala 1:3) con dos carriles trazados en su interior, que asemejan a la construcción de una carretera de tránsito y permite realizar pruebas de montaje, entrenamiento de la red neuronal y pruebas de validación pues la pista incluye curvas, líneas de carril y obstáculos representando vehículos y peatones, con iluminación variable. En resumen, el sistema de visión embebido fue evaluado a través de sesiones experimentales en una pista de pruebas a escala, simulando condiciones urbanas.

Cabe señalar que se optó por posicionar la cámara en la parte frontal del automóvil, para simular la posición de vista de un ocupante en un automóvil, la vista desde la cámara se puede apreciar en la Figura 5.

Figura 5. Vista de la cámara montada en el vehículo.

Una vez implementado el sistema de visión artificial sobre el auto a escala y puesto a prueba en la pista de experimentación, como se puede observar en la Figura 6, se inició con la transmisión de la imagen y se testeó la segmentación tanto del algoritmo de detección de carriles, así como el funcionamiento y rendimiento de la red Yolov3 y su efectividad para realizar la detección de otros automóviles y personas.

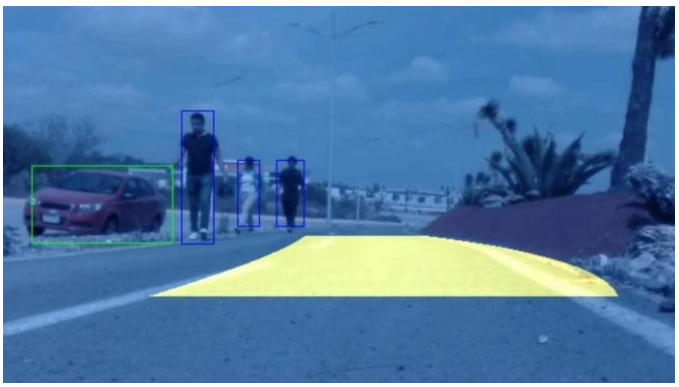

Figura 6. Pruebas de reconocimiento de personas, autos y carril.

Los automóviles fueron los objetos que la red neuronal predijo con mayor precisión, incluso en situaciones en donde solo se observaba una parte del automóvil y este se encontraba a cierta distancia, lograba detectarlo adecuadamente. En el caso de la detección de personas, se realizaron pruebas y se determinó que es capaz de detectarlas cuando se encuentran a una distancia menor a 18 m, disminuyendo la precisión conforme se alejan.

Al ejecutar las pruebas se pudo observar que la red YOLOv3-Tiny en la Kria KV260 logró una precisión media (mAP) del 92.4% para vehículos y 87.3% para peatones, mientras que para la segmentación de carriles tuvo una tasa de éxito del 95.7%.

Para medir el rendimiento de este sistema de visión artificial, se ejecutó la red neuronal y el algoritmo de detección de carriles al mismo tiempo durante 5 minutos, registrando cada 5 segundos el valor de las FPS (cuadros por segundo) para monitorear el comportamiento de la red. El gráfico de la Figura 7 muestra los resultados obtenidos, permitiendo observar que, en promedio, el valor de FPS de la red se mantiene en 3 FPS, con un mínimo de 2.3 y un máximo de 3.7.

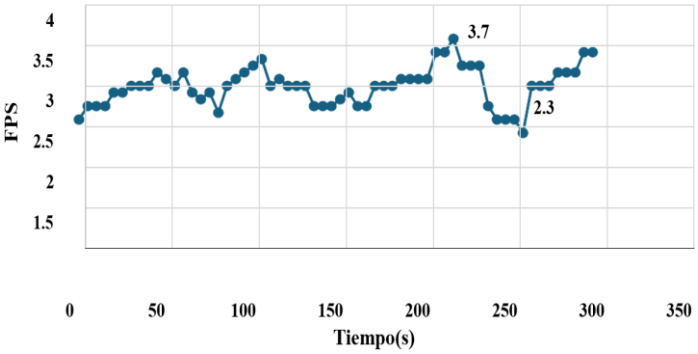

Figura 7. Gráfico de rendimiento del sistema de visión en FPS.

Con relación en el proceso de segmentación, se evaluó la precisión del algoritmo de segmentación de dos maneras:

- Basado en la distancia entre el borde del algoritmo de segmentación con el borde verdadero.
- Basado en el área de segmentación del algoritmo contra al área real.

Para evaluar la precisión en la segmentación de carriles se midió la distancia entre el borde segmentado por el algoritmo y el borde de referencia considerado como ‘verdadero’. Se

seleccionaron 80 imágenes y se analizaron las posiciones de los vértices del área segmentada, dado que en estas regiones se presentaban las mayores discrepancias. La diferencia absoluta media fue de 31.9 píxeles, equivalente al 5% del ancho de la imagen (640 px), lo cual se considera aceptable en el contexto experimental. Adicionalmente, el 16% de los puntos presentaron una diferencia inferior a 20 píxeles respecto al borde verdadero, lo que confirma una coincidencia parcial adecuada. En estudios futuros se propone complementar este análisis con métricas estandarizadas como el IoU (Intersection over Union), que proporciona una medida más robusta del solapamiento entre áreas segmentadas.

En la Tabla 1 es posible observar los resultados de la segmentación basada en distancia.

Table 1. Resultados de segmentación basados en distancia.

Métrica	Valor	Unidad
Diferencia Absoluta Media (MAD)	31.91	Píxel
Diferencia Máxima (MAXDj)	75.11	Píxel
Diferencia Mínima	5.00	Píxel
Puntos con <20 px de diferencia	16.00	%

Con base en el área segmentada por el algoritmo en comparación con el área real, se utilizaron las superficies segmentadas de 80 imágenes. Considerando la superficie segmentada por el algoritmo y la superficie de la verdad de terreno —definida como el área correcta que sigue el trayecto de la carretera—, se contemplaron las siguientes áreas: superficie de verdaderos positivos, superficie de falsos positivos, superficie de falsos negativos y superficie de verdaderos negativos.

A partir de estos valores, se calcularon las tasas de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos. De estas tasas se derivaron las métricas que se muestran en la Tabla 2.

Table 2. Resultados de la segmentación basada en áreas.

Métrica	Valor
Tasa verdaderos positivos (TPR / FVP)	81.28 %
Tasa verdaderos negativos (TNR)	84.54 %
Tasa falsos positivos (FPR)	15.46 %
Tasa falsos negativos (FNR)	18.72 %

En la evaluación basada en área se calcularon las superficies de verdaderos positivos, falsos positivos, falsos negativos y verdaderos negativos. A partir de estas métricas se obtuvieron las tasas correspondientes utilizando las fórmulas estándar: TPR = TP / (TP + FN); FNR = FN / (TP + FN); TNR = TN / (TN + FP); FPR = FP / (TN + FP).

Con base en los resultados mostrados en la Tabla 2 se puede concluir que la detección del carril con el algoritmo seleccionado se logra más del 80% de exactitud tanto para verdaderos positivos como verdaderos negativos. Esto significa que la mayoría de los pixeles son detectados adecuadamente cuando pertenecen al área de interés.

4. Discusión

El prototipo ADAS embebido presentado en este trabajo fue analizado comparativamente frente a sistemas recientes de detección de objetos basados en FPGA reportados en la literatura. Estos estudios adoptan diversas metodologías, incluyendo cuantización, paralelismo en *hardware*, arquitecturas en flujo y técnicas de compresión de modelos. Este marco comparativo permite una evaluación robusta de las ventajas técnicas y compromisos del sistema propuesto.

Implementaciones como las descritas por Zhao et al. (2024), Yang et al. (2023) y Adiono et al. (2021) emplean versiones optimizadas de YOLOv3 con estrategias de cuantización como INT8 y 4W4A, alcanzando altas tasas de inferencia de hasta 45 cuadros por segundo (FPS). Sin embargo, estos diseños están enfocados principalmente en la evaluación de modelos de detección y no incorporan estrategias de control ni retroalimentación de actuadores en entornos reales. En contraste, el presente trabajo integra detección de objetos, segmentación de carriles y actuación vehicular en una plataforma embebida unificada, ofreciendo capacidades de percepción y control en tiempo real.

Varios estudios (Ding et al., 2019; Sharma et al., 2024; Yu et al., 2022) exploran inferencia eficiente en recursos mediante cuantización de pesos, poda de filtros y reutilización de módulos convolucionales. Si bien estos métodos mejoran significativamente el rendimiento y la utilización del silicio, pueden reducir la precisión en la detección en escenarios multiclase o en condiciones de iluminación variable y oclusión. En cambio, este trabajo utiliza un modelo YOLOv3-Tiny entrenado con un conjunto de datos personalizado, adquirido en entornos locales, lo que asegura una mayor relevancia y adaptabilidad al contexto del tráfico urbano.

Comparado con otros sistemas de detección basados en FPGA reportados en la literatura, la plataforma propuesta prioriza la integración funcional sobre el rendimiento de inferencia. Aunque no supera a arquitecturas de alto nivel en velocidad o eficiencia de cuantización, ofrece ventajas significativas en términos de costo, utilidad didáctica e integración completa del sistema. Por lo tanto, el prototipo constituye una base efectiva para la experimentación en ADAS, la formación en sistemas embebidos y futuras investigaciones en tecnologías de movilidad autónoma.

La latencia total medida fue menor a 150 ms, pero es importante desglosar sus componentes. La inferencia en la FPGA tuvo una duración promedio de ~90 ms, mientras que la transmisión Wi-Fi al dispositivo Android representó aproximadamente ~35 ms del total. Otros retardos, como la adquisición de imágenes y la visualización en la GUI, sumaron el resto (~20 ms).

Este desglose permite identificar que el principal cuello de botella corresponde a la inferencia en la FPGA. Por otro lado, el sistema alcanzó un rendimiento de ~3 FPS, lo cual representa una limitación evidente frente a vehículos reales que requieren un procesamiento mucho más rápido. No obstante, para un vehículo a escala 1:5, donde la velocidad de operación

es considerablemente menor, este rendimiento es suficiente para la validación experimental y el propósito académico.

El compromiso entre precisión (mAP >92 % en detección de vehículos y 95.7 % en segmentación de carriles) y rendimiento (3 FPS) constituye un aspecto relevante de discusión. Aunque existen arquitecturas optimizadas que alcanzan decenas de FPS, el presente sistema prioriza la integración didáctica y el bajo costo sobre el rendimiento máximo, sirviendo como plataforma de experimentación en entornos académicos.

El sistema propuesto demuestra la viabilidad de usar plataformas FPGA para procesamiento de visión en aplicaciones vehiculares. Frente a CPU o GPU embebidas, la Kria KV260 ofrece mejor eficiencia energética y desempeño.

YOLOv3-Tiny representa un balance adecuado entre precisión y eficiencia computacional, siendo ideal para implementación en FPGA. Aunque existen versiones más recientes como YOLOv5 o YOLOv8, estas demandan más recursos no siempre compatibles con plataformas Edge-FPGA.

La GUI añadió gran valor al permitir interacción directa con los resultados visuales y facilitar la validación del sistema. Desde una perspectiva académica, esta interfaz es una herramienta útil para la enseñanza de percepción computacional y sistemas embebidos.

Limitaciones actuales incluyen el uso de Wi-Fi en lugar de buses automotrices como CAN o Ethernet TSN, y la detección limitada a carriles, vehículos y peatones. Futuras versiones podrían integrar señales de tráfico, segmentación semántica o fusión multiframe.

5. Conclusiones

Este trabajo presentó el desarrollo e implementación de un sistema compacto de visión embebido basado en FPGA, acompañado por una interfaz gráfica inalámbrica. Las pruebas en un vehículo a escala 1:5 confirmaron su capacidad para detección de carriles y objetos en tiempo real, con buena estabilidad y precisión.

El uso de la tarjeta Kria KV260 y aprendizaje por transferencia con YOLOv3-Tiny permitió un sistema eficiente, accesible y adaptable a entornos académicos. La GUI facilitó la interacción humana y el análisis visual del sistema.

Respecto al procesamiento de las imágenes, el sistema alcanzó una tasa de procesamiento promedio de 3 FPS, con valores mínimos de 2.3 y máximos de 3.7 FPS, mientras que la cámara empleada opera a 30 FPS. Esta diferencia entre la tasa de captura y la tasa de procesamiento refleja la latencia inherente al sistema de inferencia y segmentación sobre la FPGA. Aunque este rendimiento es limitado frente a aplicaciones automotrices reales, resulta suficiente para el contexto de un vehículo a escala 1:5 y demuestra la viabilidad del prototipo como plataforma didáctica y experimental.

Por otra parte, la transmisión de datos al dispositivo Android tuvo una latencia menor a 150 ms, permitiendo visualización sincrónica. La GUI presentó los resultados con estabilidad y sin retrasos perceptibles. Además, el sistema de aislamiento de vibraciones mejoró la calidad de imagen reduciendo el desenfoque en un 34%, favoreciendo la precisión del procesamiento visual.

El sistema tiene potencial para ser una plataforma modular y de bajo costo para el desarrollo de funciones ADAS en contextos educativos. Las futuras mejoras incluirán nuevos modelos de red neuronal, integración de LiDAR o radar, y protocolos automotrices para mayor robustez.

Agradecimientos

Se extiende un sincero agradecimiento por el apoyo del Instituto Politécnico Nacional de México, el cual hizo posible la realización de este trabajo. En particular, este proyecto fue financiado a través del programa **“Iniciación en investigación a través de la ejecución de Proyectos de investigación científica y desarrollo tecnológico”** (SIP-IPN, 20254116), otorgado a Karen Roa-Tort. Asimismo, extendemos nuestro agradecimiento a todas las personas involucradas en el desarrollo de este proyecto, así como a los revisores por sus valiosos comentarios y el tiempo dedicado a mejorar este manuscrito. Los autores han revisado y editado el contenido y asumen plena responsabilidad por el contenido de esta publicación.

Referencias

Adiono, T., Putra, A., Sutisna, N., Syafalni, I., (2021). Low latency YOLOv3-Tiny accelerator for low-cost FPGA using general matrix multiplication principle. *IEEE Access* 9, 130102–130114. DOI: 10.1109/ACCESS.2021.3120629

Al Amin, R., Hasan, M., Wiese, V., Obermaisser, R., (2024). FPGA-based real-time object detection and classification system using YOLO for edge computing. *IEEE Access* 12, 102345–102357. DOI: 10.1109/ACCESS.2024.3404623

Ding, C., Wang, S., Liu, N., Xu, K., Wang, Y., Liang, Y., (2019). REQ-YOLO: A resource-aware, efficient quantization framework for object detection on FPGAs. *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 33–42. DOI: 10.1145/3289602.3293904

Fang, N., Li, L., Zhou, X., Zhang, W., Chen, F., (2025). An FPGA-based hybrid overlapping acceleration architecture for small-target remote sensing detection. *Remote Sensing* 17(3), 494. DOI: 10.3390/rs17030494

Murendeni, T., Phiri, J., Velempini, M., (2025). Vision-based autonomous vehicle detection system using deep learning: A review. *Sensors* 25(6), 2540.
DOI: 10.3390/s25062540

Sharma, S., Guleria, K., Tiwari, S., Kumar, S., (2024). 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)*, 148–153.
DOI: 10.1109/ICCICA60014.2024.10585196

Wei, C., Zhang, T., Liu, Y., (2018). A review of computer vision technologies for advanced driver assistance systems. *IEEE Transactions on Intelligent Transportation Systems* 19(12), 3925–3940. DOI: 10.1109/TITS.2018.2828883

Yang, X., Zhuang, C., Feng, W., Yang, Z., Wang, Q., (2023). FPGA implementation of a deep learning acceleration core architecture for image target detection. *Applied Sciences* 13(7), 4144. DOI: 10.3390/app13074144

Yu, L., Zhu, J., Zhao, Q., Wang, Z., (2022). An efficient YOLO algorithm with an attention mechanism for vision-based defect inspection deployed on FPGA. *Micromachines* 13(7), 1058. DOI: 10.3390/mi13071058

Zhang, Y., Lee, S., (2020). Energy-efficient processing for embedded vision systems in autonomous vehicles. *Journal of Systems Architecture* 110, 101807.

Zhao, D., Liu, S., Zhang, Z., Zhang, Z., Tang, L., (2024). Research on ZYNQ neural network acceleration method for aluminium surface microdefects. *Digital Signal Processing*, 104900. DOI: 10.1016/j.dsp.2024.104900