International Journal of Information Science and Technological Applications-UAS
IJISTA
ISSN (en trámite)
https://revistas.uas.edu.mx/index.php/IJISTA

Mayo 2026
Vol. II.
Número. I

Análisis Comparativo del Rendimiento de Inferencia de MobileNetV2 en CPU, GPU y TPU para la Clasificación de Objetos Agrícolas en Sistemas Embebidos

Comparative Analysis of MobileNetV2 Inference Performance on CPU, GPU and TPU for Agricultural Object Classification in Embedded Systems

J.A. Becerra-Jimenez[1], J.R. Millan-Almaraz[2]

[1]Facultad de Informática / Universidad Autónoma de Sinaloa, México.
[2]Facultad de ciencias Fisicomatemáticas / Universidad Autónoma de Sinaloa, México.
J.A. Becerra-Jimenez, jaimeandres@uas.edu.mx , ORCID: 0009-0007-9816-8010
J.R. Millan-Almaraz, jrmillan@uas.edu.mx , ORCID: 0000-0002-3800-3712

Recibido: abril 2026, **Aceptado:** abril 2026, **Publicado:** mayo 2026

Resumen:
El avance de la agricultura inteligente requiere soluciones de visión por computadora eficientes que operen en dispositivos de bajo costo y consumo energético. Este artículo presenta un análisis comparativo del rendimiento de inferencia de MobileNetV2 en diferentes arquitecturas de hardware: una CPU de escritorio (Ryzen 5 5600G), una GPU (NVIDIA GTX 1070) y un sistema embebido con TPU (Google Coral Dev Board). El estudio evaluó el tiempo de inferencia y los cuadros por segundo (FPS) en modos por lotes y en línea. Los experimentos se realizaron con bases de datos de insectos agrícolas, como la mosquita blanca.
Los resultados muestran que la GPU es la más rápida en inferencia por lotes, alcanzando hasta 683 FPS con 97.93% de precisión. En contraste, la TPU Coral es más eficiente en la inferencia en línea, logrando hasta 348 FPS con 91.76% de precisión. La CPU presenta un rendimiento intermedio y menor eficiencia energética. Estos hallazgos confirman la viabilidad de MobileNetV2 para aplicaciones agrícolas en dispositivos embebidos y demuestran que la elección del hardware depende de la aplicación: la GPU es ideal para el procesamiento masivo, mientras que la TPU es óptima para aplicaciones en tiempo real en el campo.

Palabras Clave:
Agricultura de Precisión, MobileNetV2, TPU, GPU, CPU, Sistemas Embebidos.

Abstract:
The advancement of smart agriculture requires efficient computer vision solutions that can operate on low-cost, low-power devices. This study presents a comparative analysis of the inference performance of the MobileNetV2 model across different hardware architectures: a desktop CPU (Ryzen 5 5600G), GPU (NVIDIA GTX 1070), and embedded system equipped with a TPU (Google Coral Dev Board). The study evaluated inference time and frames per second (FPS) in both batch and online processing modes using datasets of agricultural insects, such as the whitefly. The results demonstrate that the GPU is the fastest architecture for batch inference, reaching up to 683.28 FPS with a 97.93% accuracy rate. In contrast, the Coral TPU proved to be the most efficient for online inference, achieving up to 348.96 FPS with 91.76% accuracy, whereas the CPU exhibited intermediate performance and lower energy efficiency. These findings confirm the viability of deploying MobileNetV2 for agricultural applications on embedded devices, highlighting that hardware selection depends strictly on the application: GPUs are ideal for massive data processing, while TPUs are optimal for real-time, edge-computing deployments in the field.

Keywords:
Precision agriculture, MobileNetV2, TPU, GPU, CPU, Embedded systems.

1. Introducción

La agricultura continúa siendo uno de los pilares fundamentales de la economía de México y de América Latina, y su sostenibilidad depende, en gran medida, de la capacidad del sector para detectar y controlar oportunamente plagas y enfermedades. La mosquita blanca (Bemisia tabaci Gennadius) representa uno de los mayores desafíos fitosanitarios a nivel mundial: este insecto chupador, capaz de colonizar más de 600 especies de plantas hospederas, provoca pérdidas económicas de miles de millones de dólares anuales mediante daño directo por succión de savia y, de forma indirecta, actuando como vector de más de 150 virus vegetales, principalmente del género Begomovirus [1]. Los impactos documentados son cuantiosos: en el estado de Georgia (EE.UU.), por ejemplo, brotes de B. tabaci generaron pérdidas de 132.3 y 161.2 millones de dólares en 2016 y 2017, respectivamente [1] Su presencia en cultivos hortícolas de la región noroeste de México donde el tomate, el pepino y el chile de agricultura protegida tienen gran relevancia económica convierte el monitoreo temprano y automatizado de esta plaga en una necesidad prioritaria para el sector agroalimentario.

Frente a este escenario, la inteligencia artificial (IA), y en particular el aprendizaje profundo (deep learning), ha emergido como una herramienta de alto potencial para automatizar el reconocimiento de plagas mediante visión por computadora. Su integración en sistemas de Agri-IoT ofrece oportunidades para optimizar la toma de decisiones, mejorar la gestión de recursos fitosanitarios y aumentar la productividad de manera sostenible [2]. Modelos de clasificación de imágenes entrenados con datos agrícolas han demostrado ser capaces de identificar especies de insectos dañinos con alta precisión, incluso bajo condiciones de campo complejas como variaciones de iluminación, fondos no controlados o solapamiento de especímenes [3]. La aplicación de estas técnicas puede transformar el monitoreo tradicional que depende de inspecciones manuales periódicas en sistemas de vigilancia continua y en tiempo real, reduciendo significativamente el tiempo de respuesta ante la aparición de infestaciones.

Sin embargo, el despliegue de estos modelos en entornos agrícolas reales impone restricciones que van más allá de la precisión del modelo. La baja conectividad a Internet en zonas rurales y periurbanas, la limitación en el consumo energético de los dispositivos de campo, y la necesidad de procesamiento en tiempo real hacen que las soluciones basadas en la nube sean frecuentemente inviables o insuficientes [4]. Este contexto ha impulsado el paradigma del cómputo en el borde (edge computing), en el cual el procesamiento ocurre directamente en el dispositivo ubicado en el campo, sin necesidad de transmitir datos a servidores remotos. Esta arquitectura ofrece ventajas claras en cuanto a latencia, privacidad de los datos y autonomía operativa [5]. No obstante, su implementación plantea una pregunta de diseño crítica: ¿cuál plataforma de hardware embebido ofrece el mejor equilibrio entre velocidad de inferencia, precisión del modelo y viabilidad de despliegue para aplicaciones agrícolas específicas? Responder esta pregunta requiere una evaluación experimental sistemática, dado que los dispositivos disponibles desde CPU de propósito general hasta aceleradores de IA especializados presentan perfiles de rendimiento y consumo muy distintos entre sí [6].

En este contexto, los modelos de clasificación ligeros han demostrado ser los candidatos naturales para el despliegue en sistemas embebidos. Entre ellos, MobileNetV2, propuesto por Sandler et al. [7], destaca por su arquitectura basada en residuales invertidos y cuellos de botella lineales (inverted residuals and linear bottlenecks), que permite operar con aproximadamente 300 millones de operaciones de punto flotante (FLOPs) y tan solo 3.4 millones de parámetros, manteniendo una precisión competitiva en tareas de clasificación de imágenes. A diferencia de otras alternativas ligeras, como MobileNetV3 o EfficientNet que, si bien son más precisas en ciertos benchmarks, requieren la función de activación hard-swish que resulta incompatible con el compilador de la Edge TPU de Google Coral [6], MobileNetV2 ofrece compatibilidad nativa y completa con dicha plataforma de hardware, permitiendo que todas sus operaciones sean ejecutadas íntegramente en el acelerador sin necesidad de recurrir a la CPU del sistema embebido. Estudios comparativos recientes han confirmado que, en escenarios de recursos limitados, MobileNetV2 demuestra una superior compatibilidad con hardware de borde y velocidades de inferencia más consistentes frente a EfficientNetV2 [8]. Su eficacia en aplicaciones agrícolas ha sido validada en múltiples trabajos, incluyendo la detección de enfermedades en hojas de tomate y la clasificación de plantas en condiciones de campo [3].

El presente estudio se enfoca en la evaluación y comparación del rendimiento de inferencia de MobileNetV2 en tres arquitecturas de hardware representativas del cómputo tradicional y del edge computing: una CPU de escritorio (AMD Ryzen 5 5600G), una GPU de consumo (NVIDIA GTX 1070) y un sistema embebido con coprocesador TPU (Google Coral Dev Board). El objetivo central es determinar cuál de estas plataformas resulta más adecuada para clasificar insectos agrícolas con énfasis en la mosquita blanca (B. tabaci) bajo dos modalidades operativas: inferencia en línea, que simula el monitoreo en tiempo real imagen por imagen, e inferencia por lotes, orientada al análisis masivo de colecciones de imágenes. Las métricas evaluadas son el tiempo promedio de inferencia, los cuadros por segundo (FPS) y la precisión del modelo. Los resultados buscan proveer criterios objetivos que orienten la selección de hardware para sistemas de visión embebida en agricultura de precisión, especialmente en contextos con restricciones de infraestructura tecnológica y energética, y sentar bases

experimentales para el desarrollo de soluciones de Agri-IoT de bajo costo desplegables en zonas rurales.

2. Trabajos Relacionados

La investigación presentada en este artículo se sitúa en la intersección de tres áreas de conocimiento activas, el primero es la aplicación de inteligencia artificial e IoT en la agricultura de precisión, el segundo es uso de modelos ligeros de visión por computadora para la detección de plagas y enfermedades en cultivos y, el tercero la comparación experimental de plataformas de hardware para inferencia en el borde. A continuación, se revisan los trabajos más relevantes en cada uno de estos ejes temáticos, identificando los vacíos que el presente estudio busca llenar.

2.1. Inteligencia artificial e IoT en la agricultura de precisión

La integración de tecnologías como el Internet de las Cosas (IoT), la inteligencia artificial (IA) y los sistemas embebidos ha transformado profundamente la gestión de las operaciones agrícolas en lo que se conoce como Agri-IoT. Esta convergencia tiene un potencial significativo para optimizar la toma de decisiones, mejorar la gestión de recursos hídricos y fitosanitarios, y aumentar la productividad de forma sostenible. Lykas y Vagelas resaltan que la adopción de estas tecnologías en sistemas agroalimentarios no solo mejora la eficiencia productiva, sino que también contribuye a la sostenibilidad ambiental a largo plazo, subrayando la relevancia de innovar en sistemas embebidos para el monitoreo en tiempo real [2].

En cuanto al monitoreo ambiental y de recursos, Morchid et al. propusieron un sistema de riego inteligente basado en IoT para la gestión del agua agrícola que integra sistemas embebidos, telemetría y cómputo en la nube, demostrando la viabilidad de soluciones de bajo costo para decisiones de campo en tiempo real [5]. Por su parte, Dos Santos et al. desarrollaron un sistema embebido que combina redes de sensores inalámbricos con un motor de predicción basado en el modelo ARIMA para anticipar condiciones adversas en cultivos, mostrando que la arquitectura de procesamiento local reduce la latencia de respuesta frente a enfoques que dependen exclusivamente de la nube [9].

La aplicación de algoritmos de aprendizaje automático (ML) y técnicas de IA en la agricultura ha sido ampliamente documentada como vía para optimizar la producción, predecir tendencias, identificar patrones y automatizar tareas críticas [10]. Saez Rojas et al. presentaron un prototipo IoT para la optimización del riego agrícola en la Región del Biobío (Chile), donde la combinación de sensores en campo con modelos predictivos permitió reducir el consumo de agua sin afectar el rendimiento de los cultivos [11]. El denominador común en todos estos estudios es la necesidad de soluciones que sean no solo precisas, sino también desplegables en condiciones de infraestructura limitada, lo que motiva directamente el enfoque de edge computing adoptado en el presente trabajo.

Pintus et al. documentaron la tendencia hacia sistemas de Edge AIoT en tiempo real para la clasificación de imágenes agrícolas, argumentando que los avances en hardware acelerador de IA y en modelos de aprendizaje profundo han hecho factible el procesamiento directo en el dispositivo de campo, eliminando la dependencia de la nube para tareas de detección urgente [6][4]. Esta transición es particularmente relevante para regiones con conectividad a Internet intermitente o inexistente, como las zonas agrícolas rurales del noroeste de México.

2.2 Modelos ligeros de visión por computadora para la detección de plagas y enfermedades

El desarrollo de modelos de visión por computadora aplicados a la detección de plagas y enfermedades agrícolas ha avanzado de forma significativa en los últimos años, impulsado en gran medida por la disponibilidad de arquitecturas CNN ligeras diseñadas para dispositivos con restricciones de cómputo. En este contexto, MobileNetV2 ha emergido como una de las arquitecturas de referencia más utilizadas, gracias a su eficiente balance entre precisión y requisitos computacionales [7].

Sharma et al. demostraron la eficacia de MobileNetV2 en la clasificación de enfermedades en hojas de tomate mediante un modelo ensemble que combina esta arquitectura con ResNet50, alcanzando una exactitud del 99.91% sobre un conjunto de datos de 11,000 imágenes en 10 categorías [3]. Este resultado refuerza la capacidad de MobileNetV2 para extraer características relevantes en imágenes agrícolas complejas incluso con tamaños de conjunto de datos moderados, lo cual es particularmente valioso en escenarios de monitoreo regional donde la recopilación masiva de imágenes resulta costosa.

Para el caso específico de detección de insectos, la revisión sistemática de Teixeira et al. sobre la detección automática de insectos mediante aprendizaje profundo documenta múltiples estudios que emplean trampas amarillas adhesivas e imágenes de campo para la identificación y conteo de plagas como mosquita blanca, áfidos y trips [12]. Los autores señalan que la variabilidad visual de los insectos en condiciones de campo determinada por factores como la distancia de captura, la densidad de organismos y las condiciones de iluminación constituye el principal reto técnico para la generalización de estos modelos, lo que justifica el uso de conjuntos de datos diversificados como el adoptado en el presente trabajo.

Srinivasa [13] y otros trabajos orientados al uso de drones para la detección temprana de plagas y malezas en

campo destacan la ventaja del monitoreo aéreo automatizado, pero señalan que el procesamiento en tiempo real de las imágenes sigue siendo un cuello de botella cuando se depende de transmisión a la nube, reforzando la necesidad de soluciones de inferencia embebida. Dong et al. propusieron un sistema IoT de monitoreo ambiental agrícola que combina LoRaWAN para transmisión de largo alcance con un módulo de reconocimiento de plagas basado en TensorFlow ejecutado en nodos de borde; reportaron una precisión de reconocimiento del 89% con tiempos de procesamiento compatibles con la operación en campo, y concluyeron que el desplazamiento del cómputo hacia el borde reduce en más del 60% la carga de transmisión y procesamiento en la nube [14].

La literatura revisada muestra consistentemente que MobileNetV2 es una de las arquitecturas con mayor adopción en sistemas agrícolas embebidos, en parte porque fue diseñada explícitamente para operar en dispositivos móviles y de baja potencia [7], y en parte porque su compatibilidad nativa con TensorFlow Lite facilita su conversión y despliegue en plataformas de hardware especializado como la Edge TPU de Google Coral [6]. Sin embargo, hasta donde se ha identificado, pocos estudios han realizado una comparación experimental directa y sistemática del rendimiento de inferencia de un mismo modelo en CPU de escritorio, GPU y TPU bajo modalidades de procesamiento en línea y por lotes, en el dominio específico de insectos agrícolas, lo que representa el aporte central del presente artículo.

2.3 Comparación de hardware para inferencia en el borde

La selección del hardware adecuado para la inferencia de modelos de aprendizaje profundo en sistemas embebidos es una decisión de diseño crítica con implicaciones directas sobre la latencia, el consumo energético y el costo de implementación. Diversos estudios han abordado esta pregunta comparando las plataformas disponibles en el mercado de edge computing, con resultados que orientan el diseño de los experimentos del presente artículo.

Tobiasz et al. realizaron un estudio comparativo de inferencia en plataformas de edge computing, Google Coral USB, Google Coral PCIe, Intel Neural Compute Stick 2 y NVIDIA Jetson Nano. Evaluando las familias MobileNet y EfficientNet con diferentes tamaños de entrada [6]. Sus resultados confirman que, para MobileNetV2 con resolución de entrada 224×224, los dispositivos Coral logran el mayor rendimiento en FPS, superando a Jetson Nano en un factor de 4.42× y al Neural Compute Stick 2 en 9.08×. Adicionalmente, documentan que MobileNetV3 no puede ejecutarse íntegramente en la Edge TPU por incompatibilidad de la función de activación hard-swish, lo que explica la selección de MobileNetV2 en el presente estudio.

Pérez-García et al. ampliaron este análisis comparando cinco dispositivos de edge computing los cuales eran la Raspberry Pi 4, Google Coral Dev Board, Google Coral Mini, NVIDIA Jetson Nano y HummingBoard Pro. Utilizando múltiples modelos de clasificación y detección de objetos [15]. Midieron tiempos de inferencia, consumo de RAM, uso de CPU y energía, encontrando que la Edge TPU de Coral logra la mejor eficiencia energética durante la inferencia activa, mientras que el Jetson Nano consume significativamente más RAM. Los autores concluyen que la selección del dispositivo óptimo depende del perfil de uso: Coral es superior para inferencia continua de alta frecuencia, mientras que Jetson ofrece mayor flexibilidad para modelos que no pueden cuantizarse completamente a INT8.

En el ámbito específico de la agricultura, la revisión de Pintus et al. sobre edge AIoT para clasificación de imágenes agrícolas en tiempo real analiza el estado del arte en hardware acelerador de IA incluyendo GPU, TPU, NPU y FPGA y concluye que el flujo de trabajo óptimo para sistemas Agri-IoT consiste en entrenar y optimizar el modelo de aprendizaje profundo en hardware de alto rendimiento (GPU), y, convertir y desplegar el modelo en dispositivos de edge para la inferencia en campo [6]. Este flujo de dos etapas es precisamente el adoptado en el presente estudio. Adicionalmente, los autores identifican como uno de los principales desafíos vigentes la falta de benchmarks estandarizados que permitan comparar plataformas de hardware bajo condiciones de aplicación agrícola específicas, señalando así el vacío que esta investigación contribuye a llenar.

En conjunto, los trabajos revisados en esta sección muestran que, si bien existen comparaciones de hardware para inferencia de modelos ligeros, estas se realizan generalmente con datasets de propósito general (ImageNet, CIFAR) y no con colecciones de imágenes de plagas agrícolas capturadas en condiciones de campo real. El presente estudio se distingue por evaluar el rendimiento de inferencia sobre un dataset propio de insectos agrícolas de importancia fitosanitaria para el noroeste de México, añadiendo la perspectiva de la modalidad de procesamiento (en línea vs. por lotes) como variable de análisis, lo que aporta directrices prácticas para el diseño de sistemas de monitoreo en este dominio.

3. Metodología

Para el análisis, se utilizó el modelo MobileNetV2, conocido por su eficiencia y diseño optimizado para dispositivos móviles y embebidos. El entrenamiento del modelo se realizó con un conjunto de datos que contenía 250 imágenes para entrenamiento y 51 para validación por cada clase, incluyendo la mosquita blanca, mosquitos, trips y otros insectos. La Tabla 1 resume la distribución del conjunto de datos utilizado para el entrenamiento del modelo.

Tabla 1. Fotos usadas en entrenamiento

Clase	Fotos entrenamiento	Fotos validación
Mosquita blanca	250	51
Mosquitos	250	51
Trip	250	51
Otros insectos	250	51

El resultado del entrenamiento de este conjunto de imágenes confirmó la convergencia del modelo y su alta precisión, obteniendo las siguientes métricas:

- Precisión: 0.9967
- Pérdida: 0.006
- Precisión de Validación: 0.9892
- Perdida de Validación: 0.0892

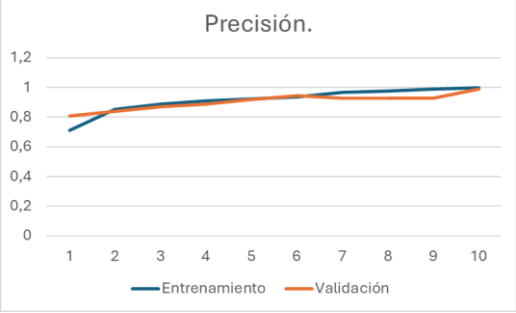

Fig. 1. Gráfica de precisión en el entrenamiento

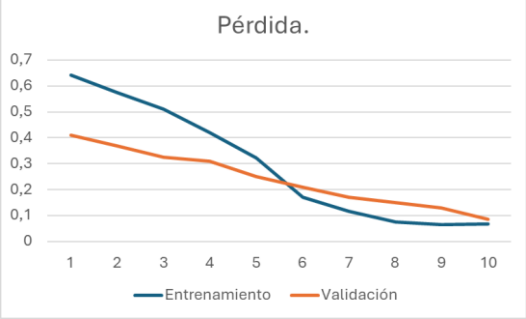

Fig. 2. Gráfica de pérdida en el entrenamiento

Para diversificar los casos y asegurar la robustez del modelo al enfrentarse a diferentes escenarios en el campo, se utilizó un conjunto de datos variado para la inferencia. Las imágenes capturan ejemplos de la mosquita blanca en distintos contextos, tales como sobre una hoja de planta (ver Fig. 3), en trampas amarillas a mayor distancia (ver Fig. 4) y acompañada de otros insectos en tomas de mayor acercamiento (ver Fig. 5).

Fig.3. Mosquita blanca en hoja

Fig. 4. Mosquita blanca en trampa amarilla

Fig. 5. Mosquita blanca en trampa amarilla

Una vez entrenado el modelo, el experimento se llevó a cabo en las siguientes plataformas de hardware para realizar las pruebas de inferencia:

PC de Escritorio: Equipada con una CPU AMD Ryzen 5 5600G y una GPU NVIDIA 1070.

Sistema Embebido: Una Coral Dev Board con su CPU NXP i.MX 8M SoC y el coprocesador Edge TPU ML accelerator.

Se realizaron pruebas de inferencia utilizando un conjunto de datos propio (ver Tabla 2).

Tabla 2. Fotos para inferencias.

Clase	Fotos para inferencias
Mosquita blanca	75
Mosquitos	52
Trip	38
Otros insectos	55

Las pruebas se ejecutaron bajo dos modalidades:

- Modo en línea (*Online*): Se procesó una sola imagen por cada iteración,

simulando una aplicación en tiempo real.

- Modo por lotes (Batch): Se procesaron múltiples imágenes en un solo lote, lo cual es relevante para tareas que requieren un alto rendimiento masivo, como el análisis de imágenes aéreas.

Las métricas evaluadas fueron el tiempo promedio de inferencia, los cuadros por segundo (FPS) y el porcentaje de aciertos del modelo. Es importante destacar que no fue posible realizar la inferencia por lotes en el núcleo TPU debido a su arquitectura, ya que su entrada solo permite inferir una imagen a la vez.

4. Resultados

A continuación, se presenta la comparativa de los resultados de inferencia obtenidos en las distintas plataformas y modalidades evaluadas. La Tabla 3 resume las métricas de tiempo promedio de inferencia, cuadros por segundo (FPS) y porcentaje de precisión del modelo MobileNetV2. Es importante destacar que no fue posible realizar la inferencia por lotes en el núcleo TPU debido a su arquitectura, ya que su entrada solo permite inferir una imagen a la vez.

Tabla 3. comparativa de inferencias en diferentes escenarios

Tipo de inferencia	Dispositivo	Tiempo prom. inferencia (s)	FPS	Precisión (%)
Lotes	ARM-Cortex A53	0.2129	4.69	96.50%
	Ryzen 5600	0.01286	72.13	97.93%
	NVIDIA 1070	0.00146	683.28	97.93%
Online	Ryzen 5600	0.0464	21.54	96.55%
	NVIDIA 1070	0.0415	23.93	96.5%
	ARM-Cortex A53	0.4217	2.37	91,13%
	TPU Coral	0.00287	348.96	91.76%

Para una visualización más clara de la tasa de procesamiento, se generaron gráficas de barras con desviación estándar para cada modalidad. Como se observa en la Fig. 6, en el modo por lotes, la GPU NVIDIA 1070 presenta la mayor cantidad de cuadros por segundo.

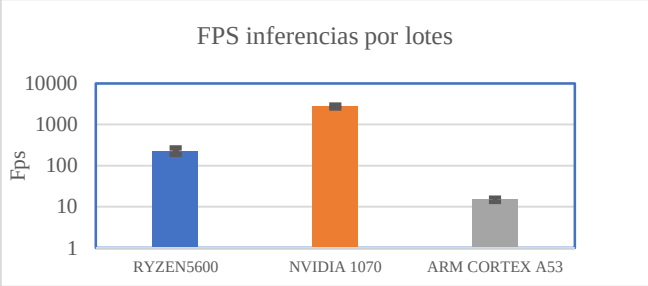

Fig. 6. Comparativa FPS inferencias por lotes

Por otro lado, la Fig. 7 muestra el rendimiento en FPS en la modalidad en línea, destacando las métricas obtenidas por la TPU Coral frente al resto de los dispositivos.

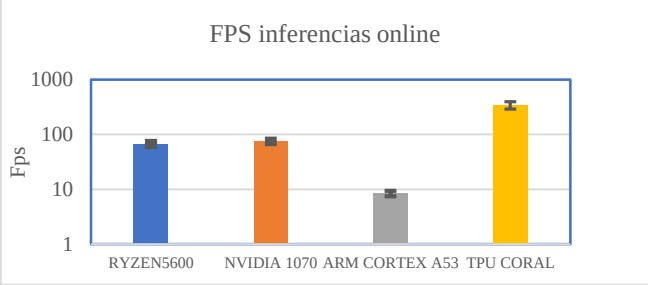

Fig. 7. Comparativa FPS inferencias online

Para registrar el comportamiento del hardware a través del tiempo, se graficó el tiempo total requerido para cada iteración. En la modalidad de inferencia por lotes, la Fig. 8 muestra el desempeño en la CPU del sistema embebido (ARM-Cortex A53). Por su parte, la Fig. 9 presenta los tiempos de la CPU Ryzen 5600, y la Fig. 10 ilustra la inferencia en la GPU NVIDIA 1070.

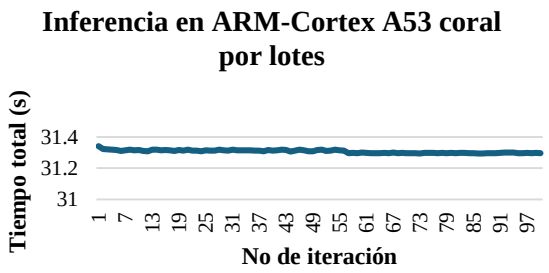

Fig. 8. Inferencia ARM-Cortex A53 coral por lotes

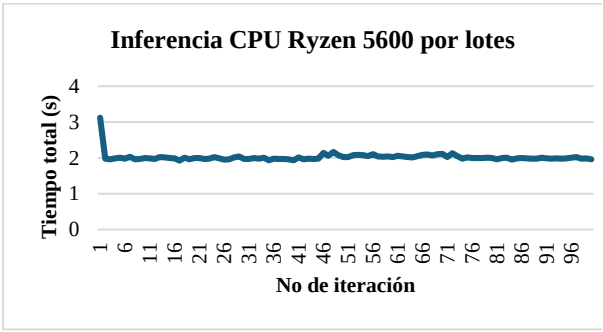

Fig. 9. Inferencia CPU Ryzen 5600 por lotes

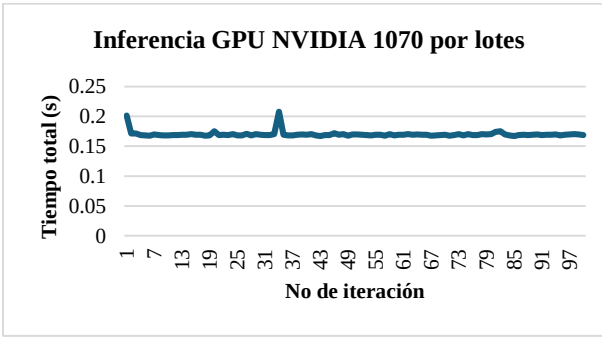

Fig. 10. Inferencia GPU NVIDIA 1070 por lotes

En cuanto a la modalidad en línea, la Fig. 11 detalla la curva de iteraciones de la CPU Ryzen 5600. La Fig. 12 expone el rendimiento de la GPU NVIDIA 1070. La Fig. 13 evidencia los tiempos de inferencia de la CPU del sistema embebido y, finalmente, la Fig. 14 documenta el comportamiento de la TPU Coral a lo largo de las cien iteraciones.

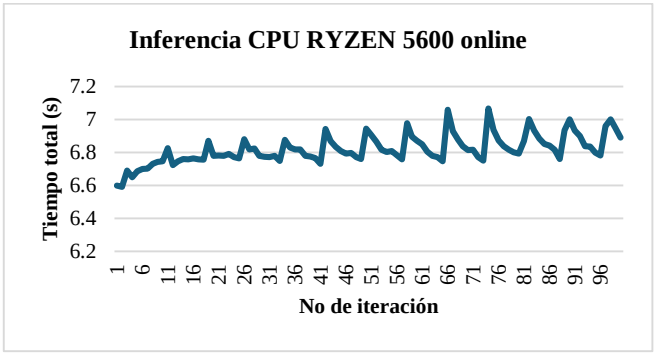

Fig. 11 Inferencia CPU Ryzen 5600 online

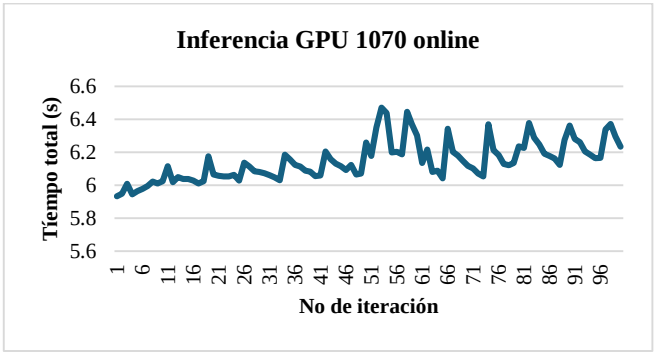

Fig. 12. Inferencia GPU 1070 online

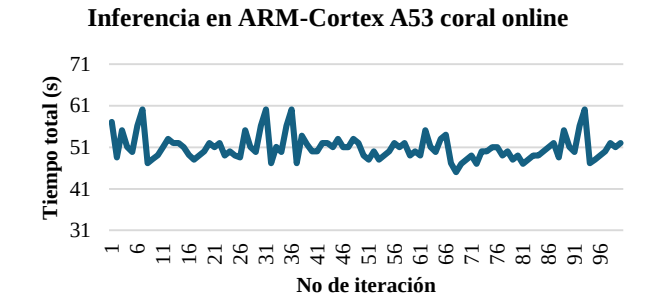

Fig. 13. Inferencia en ARM-Cortex A53 coral online

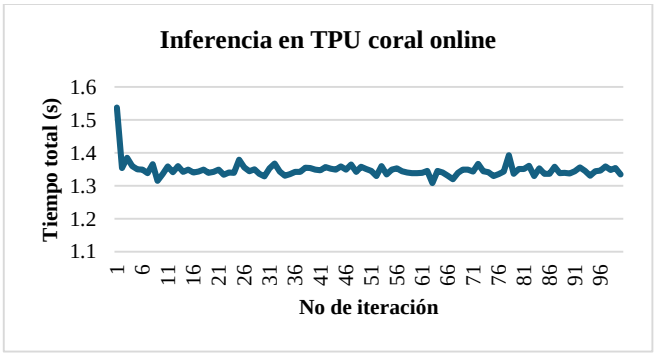

Fig. 14. Inferencia en TPU coral online

5. Análisis de Resultados

Los datos obtenidos confirman que el rendimiento de inferencia depende de manera directa de la arquitectura del hardware utilizado y de la modalidad de procesamiento exigida por la aplicación. La precisión del modelo se mantuvo consistentemente alta en todos los dispositivos, validando la robustez de MobileNetV2; sin embargo, los tiempos de respuesta variaron drásticamente.

En el procesamiento por lotes, la GPU (NVIDIA 1070) demostró una superioridad absoluta al alcanzar el mejor rendimiento con un tiempo de inferencia de 0.00146 segundos y 683.28 FPS. Este resultado se explica por la capacidad inherente de la GPU para paralelizar el procesamiento. Como se evidenció en la Fig. 10, la estabilidad de los tiempos a lo largo de las iteraciones confirma que esta arquitectura es la idónea para tareas que implican el análisis masivo de datos o bases de imágenes recopiladas previamente.

En contraste, para la inferencia en línea la cual es esencial para aplicaciones en tiempo real donde cada imagen debe ser procesada individualmente con la mínima latencia posible, la TPU Coral demostró ser la arquitectura más eficiente. Logró el tiempo de inferencia más bajo en esta modalidad (0.00287 segundos, equivalente a 348.96 FPS). Aunque la GPU también reportó tiempos bajos en la modalidad en línea (Fig. 12), mostró ligeras variaciones y no alcanzó el nivel de optimización de la TPU. La curva casi plana observada en la Fig. 14 confirma que la TPU está diseñada específicamente para ejecutar inferencias con una eficiencia extrema y mínima latencia, cumpliendo con los requisitos críticos del edge computing.

Respecto al rendimiento de las unidades centrales de procesamiento, la CPU de escritorio (Ryzen 5600) mantuvo un desempeño intermedio y estable, aunque con una latencia considerable para aplicaciones de respuesta instantánea. Por último, la CPU del sistema embebido (ARM-Cortex A53) presentó los tiempos de inferencia más altos (0.4217 segundos y 2.37 FPS en modo en línea). Este bajo desempeño subraya que las CPU tradicionales de bajo consumo tienen fuertes limitaciones para el procesamiento de modelos de aprendizaje profundo en tiempo real, lo que hace indispensable la integración de coprocesadores especializados, como la TPU, para viabilizar el desarrollo de sistemas agrícolas inteligentes en dispositivos embebidos.

6. Conclusiones

Este estudio concluye que, para la ejecución eficiente de modelos de aprendizaje profundo ligero como MobileNetV2 en aplicaciones de agricultura inteligente, la selección de la arquitectura de hardware es un factor determinante. Los resultados demuestran que la TPU (Google Coral Dev Board) es la solución óptima para la inferencia en tiempo real en sistemas embebidos, gracias a su excelente rendimiento de latencia y bajo consumo de energía, características indispensables para el edge computing en zonas de cultivo. Por otro lado, para tareas que no exigen inmediatez, como el análisis masivo fuera de línea de grandes bases de datos fotográficas, las GPUs (como la NVIDIA 1070) continúan siendo la opción más eficiente debido a su sobresaliente capacidad de procesamiento paralelo.

La importancia de este análisis radica en demostrar la viabilidad de utilizar dispositivos de bajo costo y bajo consumo energético para tareas críticas, como la detección oportuna de la mosquita blanca, superando las limitaciones tradicionales de las CPU en sistemas embebidos. Como trabajo futuro, se propone desarrollar un prototipo de sistema completo que integre la Coral Dev Board con una cámara para la clasificación automática de objetos agrícolas en un entorno real. Este prototipo permitirá la validación de la metodología directamente en el campo y facilitará su integración con otros componentes del Internet de las Cosas (IoT), sentando las bases para una gestión proactiva, autónoma y más eficiente en el ámbito de la agricultura de precisión.

7. Referencias

[1] Y. Li, G. N. Mbata, S. Punnuri, A. M. Simmons, y D. I. Shapiro-Ilan, “Bemisia tabaci on Vegetables in the southern United States: Incidence, impact, and management”, *Insects*, vol. 12, núm. 3, pp. 1–29, mar. 2021, doi: 10.3390/insects12030198.

[2] C. Lykas y I. Vagelas, “Innovations in Agriculture for Sustainable Agro-Systems”, el 1 de septiembre de 2023, Multidisciplinary Digital Publishing Institute (MDPI). doi: 10.3390/agronomy13092309.

[3] J. Sharma *et al.*, “Deep learning based ensemble model for accurate tomato leaf disease classification by leveraging ResNet50 and MobileNetV2 architectures”, *Sci. Rep.*, vol. 15, núm. 1, p. 13904, dic. 2025, doi: 10.1038/s41598-025-98015-x.

[4] M. Pintus, F. Colucci, y F. Maggio, “Emerging Developments in Real-Time Edge AIoT for Agricultural Image Classification”, el 1 de marzo de 2025, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/iot6010013.

[5] A. Morchid, R. Jebabra, H. M. Khalid, R. El Alami, H. Qjidaa, y M. Ouazzani Jamil, “IoT-based smart irrigation management system to enhance agricultural water security using embedded systems, telemetry data, and cloud computing”, Results in Engineering, vol. 23, p. 102829, sep. 2024, doi: 10.1016/j.rineng.2024.102829.

[6] R. Tobiasz, G. Wilczynski, P. Graszka, N. Czechowski, y S. Luczak, "Edge Devices Inference Performance Comparison", Journal of Computing Science and Engineering, vol. 17, núm. 2, pp. 51–59, 2023, doi: 10.5626/JCSE.2023.17.2.51.

[7] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, y L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks", en 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510–4520. doi: 10.1109/CVPR.2018.00474.

[8] T. Njoroge, R. Kibuku, y K. Mugoye, "Comparative and edge-hybrid modeling of EfficientNetV2 and MobileNetV2 for multi-classcrop disease classification with statistical validation", Journal of Edge Computing, vol. 4, pp. 234–262, nov. 2025, doi: 10.55056/jec.905.

[9] U. J. L. dos Santos, G. Pessin, C. A. da Costa, y R. da Rosa Righi, "AgriPrediction: A proactive internet of things model to anticipate problems and improve production in agricultural crops", Comput. Electron. Agric., vol. 161, pp. 202–213, jun. 2019, doi: 10.1016/j.compag.2018.10.010.

[10] S. Prof, B. Sharon, P. P, P. S, S. Prince, y R. R, "Smart Agriculture with IoT", International Journal of Innovative Research in Information Security, vol. 09, núm. 03, pp. 225–228, jun. 2023, doi: 10.26562/ijiris.2023.v0903.31.

[11] J. I. Saez Rojas, J. M. Pantoja, M. Matamala, I. C. Briceno, J. P. Vasconez, y A. R. Romero-Conrado, "An IoT-Based Prototype for Optimizing Agricultural Irrigation: A Case Study in the Biobio Region of Chile", Proce*dia Comput. Sci.*, vol. 238, pp. 1009–1014, 2024, doi: 10.1016/j.procs.2024.06.127.

[12] A. C. Teixeira, J. Ribeiro, R. Morais, J. J. Sousa, y A. Cunha, "A Systematic Review on Automatic Insect Detection Using Deep Learning", el 1 de marzo de 2023, *MDPI*. doi: 10.3390/agriculture13030713.

[13] B. Revanasiddappa, C. S. Arvind, y S. Swamy, "Real-time early detection of weed plants in pulse crop field using drone with IoT", *International Journal of Agricultural Technology*, vol. 16, núm. 5, pp. 1227–1242, 2020, Consultado: el 4 de mayo de 2026. [En línea]. Disponible en: https://li04.tci-thaijo.org/index.php/IJAT/article/view/7422

[14] M. Dong, H. Yu, Z. Sun, L. Zhang, Y. Sui, y R. Zhao, "Research on agricultural environmental monitoring Internet of Things based on edge computing and deep learning", *Journal of Intelligent Systems*, vol. 33, may 2024, doi: 10.1515/jisys-2023-0114.

[15] A. Garcia-Perez, R. Miñón, A. I. Torre-Bastida, y E. Zulueta-Guerrero, "Analysing Edge Computing Devices for the Deployment of Embedded AI", *Sensors*, vol. 23, núm. 23, p. 9495, dic. 2023, doi: 10.3390/s23239495.