



UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERÍA
CARRERA DE TELECOMUNICACIONES

Desarrollo de un sistema detector del estado de madurez de peras por
visión artificial en tiempo real.

Trabajo de Titulación para optar al título de:
Ingeniero en Telecomunicaciones

Autor:

Parra Cruz Kevin Raúl

Tutor:

Pedro Fernando Escudero Villa, Ph.D.

Riobamba, Ecuador. 2026

DECLARATORIA DE AUTORÍA

Yo, Parra Cruz Kevin Raúl con cédula de ciudadanía 080441170-0, autor del trabajo de investigación titulado: **Desarrollo de un sistema detector del estado de madurez de peras por visión artificial en tiempo real.**, certifico que la producción, ideas, opiniones, criterios, contenidos y conclusiones expuestas son de mí exclusiva responsabilidad.

Asimismo, cedo a la Universidad Nacional de Chimborazo, en forma no exclusiva, los derechos para su uso, comunicación pública, distribución, divulgación y/o reproducción total o parcial, por medio físico o digital; en esta cesión se entiende que el cesionario no podrá obtener beneficios económicos. La posible reclamación de terceros respecto de los derechos de autor (a) de la obra referida, será de mi entera responsabilidad; librando a la Universidad Nacional de Chimborazo de posibles obligaciones.

En Riobamba, a la fecha de su presentación 16 de marzo de 2026.



Kevin Raúl Parra Cruz

C.I: 0804411700

DICTAMEN FAVORABLE DEL PROFESOR TUTOR

Quien suscribe, Pedro Fernando Escudero Villa catedrático adscrito a la Facultad de Ingeniería, por medio del presente documento certifico haber asesorado y revisado el desarrollo del trabajo de investigación titulado: **Desarrollo de un sistema detector del estado de madurez de peras por visión artificial en tiempo real.**, bajo la autoría de Kevin Raúl Parra Cruz; por lo que se autoriza ejecutar los trámites legales para su sustentación.

Es todo cuanto informar en honor a la verdad; en Riobamba, a los 18 días del mes de Febrero de 2026.



PhD. Pedro Fernando Escudero Villa


C.I: 0603612524

CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

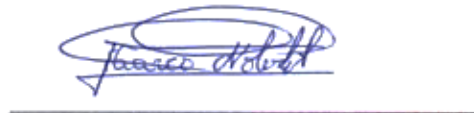
Quienes suscribimos, catedráticos designados Miembros del Tribunal de Grado para la evaluación del trabajo de investigación “**Desarrollo de un sistema detector del estado de madurez de peras por visión artificial en tiempo real**” por Kevin Raúl Parra Cruz, con cédula de identidad número 080441170-0, bajo la tutoría de PhD. Pedro Fernando Escudero Villa; certificamos que recomendamos la APROBACIÓN de este con fines de titulación. Previamente se ha evaluado el trabajo de investigación y escuchada la sustentación por parte de su autor; no teniendo más nada que observar.

De conformidad a la normativa aplicable firmamos, en Riobamba 30 días del mes de marzo de 2026.

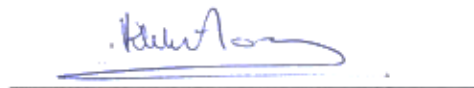
Juan Carlos Cepeda Pacheco, PhD.
PRESIDENTE DEL TRIBUNAL DE GRADO



Marco Antonio Nolivos Vimos, Mgs.
MIEMBRO DEL TRIBUNAL DE GRADO



Klever Hernán Torres Rodríguez, Dr.
MIEMBRO DEL TRIBUNAL DE GRADO





CERTIFICACIÓN

Que, **Parra Cruz Kevin Raúl** con CC: **080441170-0**, estudiante de la Carrera **Telecomunicaciones**, Facultad de **Ingeniería**; ha trabajado bajo mi tutoría el trabajo de investigación titulado "**Desarrollo de un sistema detector del estado de madurez de peras por visión artificial en tiempo real.**", cumple con el 5% de similitud y 9% de Inteligencia Artificial, de acuerdo con el reporte del sistema Anti plagio **Compilation Magister+**, porcentaje aceptado de acuerdo a la reglamentación institucional, por consiguiente autorizo continuar con el proceso.

Riobamba, 10 de marzo de 2026



PhD. Pedro Fernando Escudero Villa
TUTOR

DEDICATORIA

A Dios, por ser la luz en los momentos de incertidumbre, por darme la sabiduría que me permite comprender la ciencia y por darme la fortaleza para acabar con esta prueba. Su gracia ha sido el motor de mi perseverancia, el no rendirnos en circunstancias difíciles.

A mis padres, los constructores de mi vida y mi mayor ejemplo de sacrificio. Este título profesional es tanto suyo como mío; gracias por enseñarme que se llega a la cima con carácter y por explicarme que el amor es la base de cualquier gran proyecto que se lleve a cabo. Todo mi esfuerzo es un tributo a su fe ciega en mí.

A mis amigos, los cuales me ayudaron a continuar con este largo camino y conforme pasaba el tiempo se terminaron convirtiendo en más que amigos. Su infaltable compañía hizo de esta experiencia académica en un proceso traducido a crecimiento y lealtad como personas.

Kevin Raúl Parra Cruz

AGRADECIMIENTO

A la Universidad Nacional de Chimborazo y de manera especial a la Facultad de Ingeniería, por haberme brindado los conocimientos y las herramientas necesarias para mi formación como ingeniero. Gracias por ser el escenario de mi crecimiento académico y profesional.

A mi tutor de tesis, Pedro Fernando Escudero Villa PhD, quien en su inmensa sabiduría supo guiar con paciencia su visión técnica conmigo. Sus observaciones mejoraron este proyecto e incentivaron el valor del rigor científico de este proyecto.

Finalmente agradezco infinitamente a todas las personas que, de cualquier forma, aportaron con sus críticas y su experiencia para llegar a este punto de mi vida, a todos ellos, mi más sentido reconocimiento.

Kevin Raúl Parra Cruz

ÍNDICE GENERAL

DECLARATORIA DE AUTORÍA

DICTAMEN FAVORABLE DEL PROFESOR TUTOR

CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

DEDICATORIA

AGRADECIMIENTO

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

ABSTRACT

CAPÍTULO I	15
1. Introducción.....	15
1.1. Antecedentes.....	15
1.2. Planteamiento del problema	15
1.3. Justificación	16
1.4. Objetivos.....	18
1.4.1. Objetivo general.....	18
1.4.2. Objetivo específico	18
CAPÍTULO II.....	19
2. Marco teórico.....	19
2.1. Estado del arte.....	19
2.1.1. Avances en el contexto nacional.....	19
2.1.2. Avances en el contexto internacional	19
2.2. Términos y definiciones	21
2.2.1. Inteligencia artificial (IA).....	21
2.2.2. Aprendizaje profundo (DL)	22
2.2.3. Visión artificial	22
2.2.4. Aprendizaje automático (ML)	23
2.2.5. Redes Neuronales Convolucionales (CNN)	23
2.2.6. Solo se vive una vez (YOLO).....	24

2.2.7. Transformador de detección en tiempo real (RTDETR)	24
2.2.8. Precisión de porcentaje	25
2.2.9. Plataforma de desarrollo	25
2.2.10. Tiempo real	26
2.2.11. Producto identificado	26
CAPÍTULO III	28
3. Metodología.....	28
3.1. Tipo de Investigación	28
3.2. Diseño de Investigación.....	28
3.3. Técnicas de recolección de Datos.....	29
3.4. Población de estudio y tamaño de muestra.....	29
3.5. Hipótesis	29
3.6. Métodos de análisis, y procesamiento de datos.	30
3.6.1. Diagrama de flujo	30
3.7. Configuración del entorno experimental	32
3.7.1. Especificaciones de la infraestructura.....	32
CAPÍTULO IV.....	34
4. Resultados y discusión	34
4.1. Análisis del Entrenamiento y Convergencia.....	34
4.2. Configuración del entorno experimental	34
4.2.1. Modelo de entrenamiento para la familia YOLO	34
4.2.2. Modelo de entrenamiento para RT-DETR	35
4.2.3. Firmware de control y activación	36
4.2.4. Curva de rendimiento para YOLOv5.....	37
4.2.5. Curva de rendimiento para YOLOv11.....	38
4.2.6. Curva de rendimiento para RT-DETR.....	39
4.3. Evaluación comparativa del rendimiento de detección	39
4.4. Análisis de la matriz de confusión.....	40
4.5. Fundamentación matemática y análisis de errores de confianza	40
4.5.1. Error Absoluto Medio (MAE)	40
4.5.2. Error porcentual absoluto medio (MAPE).....	41
4.6. Validación estadística de significancia mediante la prueba de Kruskal-Wallis	41

4.7.	Contraste de hipótesis y validación estadística.....	42
4.7.1.	Comparaciones por parejas (Post-hoc)	43
4.7.2.	Diagrama de vínculos y dispersión.....	43
4.8.	Análisis por categoría de madurez.....	44
	CAPÍTULO V	46
5.	Conclusiones y recomendaciones	46
5.1.	Conclusiones.....	46
5.2.	Recomendaciones	47
	BIBLIOGRAFÍA	48
	ANEXOS	52

ÍNDICE DE TABLAS

Tabla 1. Discusión de antecedentes nacionales e internacionales	20
Tabla 2. Comparativa de métricas de rendimiento por arquitectura.....	39
Tabla 3. Análisis comparativo de métricas de error de confianza.....	41
Tabla 4. Contraste de hipótesis.....	42
Tabla 5. Comparaciones por parejas de los algoritmos (Post-hoc)	43

ÍNDICE DE FIGURAS

Figura 1. Jerarquía técnica de la Inteligencia Artificial y su especialización en arquitecturas de aprendizaje profundo [26].	22
Figura 2. Arquitectura red neuronal profunda capas ocultas diagrama [28].	22
Figura 3. Etapas procesamiento visión artificial agricultura [30].	23
Figura 4. Flujo de trabajo aprendizaje automático supervisado [31].	23
Figura 5. Arquitectura del diagrama por capas CNN [32].	24
Figura 6. Matriz de confusión para el modelo YOLOv5 [33].	24
Figura 7. Estructura de la red RT-DETR [34].	25
Figura 8. Cálculo de precisión [35].	25
Figura 9. Arquitectura sistema visión artificial hardware software diagrama [36].	26
Figura 10. procesamiento en tiempo real flujo de datos industrial [37].	26
Figura 11. Peras clasificación calidad visión artificial [38].	27
Figura 12. Flujo metodológico experimental. Se detallan las etapas de preprocesamiento de datos, entrenamiento comparativo de arquitecturas CNN (YOLOv5, YOLOv11) y Transformers (RT-DETR), y los criterios de evaluación para la selección del modelo final.	31
Figura 13. Arquitectura sistemática del prototipo para la clasificación automatizada de peras.	32
Figura 14. Curvas de rendimiento YOLOv5.	38
Figura 15. Curvas de rendimiento YOLOv11.	38
Figura 16. Curvas de rendimiento RT-DETR.	39
Figura 17. Matriz de confusión para la clasificación de estados de madurez de la pera entre los modelos YOLOv5, YOLOv11 y RT-DETR sobre el conjunto de datos de prueba.	40
Figura 18. Prueba de Kruskal-Wallis para muestras independientes.	42
Figura 19. Comparaciones por pareja del tipo de algoritmo.	44

RESUMEN

En esta investigación se evalúa un sistema de visión artificial en tiempo real diseñado para clasificar el estado de la madurez de la pera. El estudio evalúa la calidad de las distintas redes neuronales profundas en un entorno automatizado de inspección industrial. Para ello, en el experimento se utilizó un prototipo de cinta para una velocidad constante de 0.05 m/s dónde al pasar las peras dentro de la misma se capturó la imagen tiempo real con una cámara ubicada encima de la cinta transportadora donde interactuaron de forma independiente los tres modelos de detección de objetos: YOLOV5, YOLOV11 y RTDETR. Bajo un enfoque experimental y cuantitativo, la metodología analizó una muestra de 108 observaciones distribuidas en tres categorías: verdes, amarilla y podrida. El tratamiento estadístico que se ha ejecutado mediante la prueba no paramétrica de Kruskal-Wallis determinó diferencias significativas entre las distintas arquitecturas realizadas ($p < 0.001$). Los datos obtenidos posicionan al modelo RTDETR como la solución más precisa al registrar el índice de error más bajo de 0.242. No obstante, YOLOv5 reflejó el rendimiento más bajo dentro de los modelos evaluados, obteniendo un error máximo de 0.795, lo que plasma una notable diferencia frente a las demás arquitecturas. En cambio, RT-DETR obtuvo los mejores resultados en la identificación de los estados de madurez de la pera, debido a su estructura de algoritmo basada en Transformers la cual permite una mejor detección. Este descubrimiento sugiere que la integración de modelos basados en mecanismos de atención mejora sustancialmente la confiabilidad de la supervisión industrial frente a las arquitecturas convolucionales tradicionales en trabajos de visión artificial en tiempo real.

Palabras claves: Redes neuronales profundas, Detección de objetos, Madurez de frutos, Mecanismos de atención, RT-DETR, YOLO.

ABSTRACT

This research evaluates a real-time machine vision system designed to classify the ripeness of pears. The study assesses the performance of different deep neural networks in an automated industrial inspection environment. For this purpose, a prototype conveyor belt operating at a constant speed of 0.05 m/s was used. As the pears moved along the belt, a camera positioned above it captured images in real time. Three object detection models—YOLOv5, YOLOv11, and RT-DETR—were evaluated independently. Using an experimental and quantitative approach, the methodology analyzed a sample of 108 observations distributed across three categories: green, yellow, and rotten. Statistical analysis was performed using the non-parametric Kruskal–Wallis test, which determined significant differences among the evaluated architectures ($p < 0.001$). The results obtained position the RT-DETR model as the most accurate solution, registering the lowest error rate of 0.242. However, YOLOv5 exhibited the lowest performance among the evaluated models, with a maximum error rate of 0.795, showing a significant difference compared to the other architectures. In contrast, RT-DETR achieved the best results in identifying pear ripeness stages due to its transformer-based architecture, which allows improved detection performance. This finding suggests that the integration of attention-based models substantially improves the reliability of industrial monitoring systems compared to traditional convolutional architectures in real-time computer vision applications.

Keywords: Deep neural networks, Object detection, Fruit ripeness, Attention mechanisms, RT-DETR, YOLO.



Reviewed by:
MsC. Edison Damian Escudero
ENGLISH PROFESSOR
C.C.0601890593

CAPÍTULO I

1. Introducción

1.1. Antecedentes

La optimización de los protocolos de clasificación dentro de la industria agrícola constituye hoy una exigencia para mitigar el desperdicio postcosecha y asegurar estándares de calidad superiores. En este escenario, la visión artificial se consolida como el eje motor para la automatización de la inspección de frutos. Trabajos previos, como de Fernández Nieto [1], reafirmaron la utilidad de algoritmos tradicionales (KNN y SVM) en la clasificación de peras. Sin embargo, el avance tecnológico convirtió estos métodos manuales a favor de las Redes Neuronales Convolucionales (CNN). Bajo este planteamiento, investigadores como Alvarado y Molina alcanzaron rendimientos del 87%, asentaron el éxito del sistema en la extracción autónoma de descriptores profundos [2].

La factibilidad de estas soluciones en plantas de procesamiento depende estrechamente de su ejecución en tiempo real. Experiencias anteriores, entre las que sobresalen las de Erazo Narváez [3] y Boy Díaz [4], aseguran que la simbiosis entre sistemas embebidos y algoritmos de alta complejidad permite alcanzar datos cercanos al 98% en la identificación del punto de madurez. Alternativamente, existe un desafío técnico esencial: la heterogeneidad visual de la fruta, derivada de los diversos factores fenotípicos y condiciones de iluminación, suele comprometer la credibilidad de las arquitecturas convencionales [5].

Frente a estas limitaciones, el presente estudio plantea una fractura metodológica mediante el análisis de modelos de detección de última generación: YOLOv5, YOLOv11 [6] y RTDETR [7]. La investigación profundiza en la implementación de mecanismos de atención, propios de la arquitectura transformer, para trascender las limitantes de las distintas capas convoluciones estándar. Este sistema no solo persigue una mayor exactitud en la discriminación de los estados verde, amarilla y podrido, sino que avala un flujo operativo estable en bandas transportadoras automatizadas. Como resultado, este trabajo aporta una arquitectura vigorosa que moderniza la cadena de suministros y provee a la industria frutícola de herramientas precisas para la gestión de calidad en tiempo real [7].

1.2. Planteamiento del problema

La precisión en la categorización del estado de madurez de frutos define la competitividad y rentabilidad de la industria agrícola contemporánea. El punto de maduración determina parámetros críticos de calidad como la textura, el perfil de sabor y la apariencia estética; factores que rigen la aceptación final del consumidor [1], [8]. Habitualmente, las plantas de procesamiento dependen de la supervisión visual manual o la palpación física; no obstante, estos registros introducen altos niveles de subjetividad, fatiga y errores humanos y lentitud operativa en entornos a gran escala [2], [9].

En este contexto, la visión artificial constituye una tecnología estratégica al conceder análisis no destructivos en tiempo real [3]. La extensión de sistemas automáticos reduce el deterioro mecánico de los frutos y mitiga riesgos fitosanitarios al evitar contacto directo [3] [10]. A

pesar de estas virtudes, la detección autónoma enfrenta desafíos persistentes: la heterogeneidad fenotípica, las fluctuaciones de iluminación ambiental y la variabilidad morfológica entre las distintas variedades de pera [5], [11]. Asimismo, la implementación a nivel industrial masiva encuentra barreras en los costos de infraestructura especializada y de escasez de conjuntos de datos etiquetados para tareas específicas de madurez [2], [12].

Aunque investigaciones previas validan el uso de aprendizaje profundo (Deep Learning) para clasificar musáceas y fresas mediante el análisis cromático [4], [13], [14], la pera requiere algoritmos de mayor complejidad para apartar estados de madurez con altos niveles de confianza [15]. Los modelos convolucionales estándar (CNN) muestran trabas de precisión bajo condiciones ambientales variables [16], [17]. Por ello, el campo de la visión artificial crece hacia arquitecturas de última generación como las familias de YOLO y RTDETR, las cuales integran mecanismos de atención y de píxeles de la imagen para capturar dependencias especiales complejas [6], [7], [18].

Inicialmente, Fruitscan [19] argumenta el potencial de la tecnología para la estimación de cosechas; sin embargo, la adaptación de estos sistemas para la detección específica del estado de madurez en peras permanece como un reto técnico sin resolver [1], [20]. La distancia actual demanda el desarrollo de una arquitectura robusta que integre aprendizaje profundo y procesamiento en tiempo real. Este trabajo integra dicha necesidad mediante la evaluación de los modelos de vanguardia, buscando optimizar la toma de decisiones postcosecha y fortalecer la confiabilidad de la cadena de suministro a nivel industrial [3], [21], [22].

1.3. Justificación

La automatización en la fase de clasificación frutícola representa una exigencia estratégica para elevar los estándares de calidad y optimizar la valoración comercial de la producción. La identificación del punto óptimo de madurez garantiza la integridad de sus atributos organolépticos: sabor, textura y una colorimetría epidérmica precisa; son factores que consolidan la credibilidad del consumidor final. A pesar de ello, la sujeción a métodos convencionales como el tacto y la verificación ocular introduce márgenes de error intolerable en líneas de procesamiento masivo, limitando la competitividad del sector ante las exigencias del mercado global.

Bajo esta premisa, la visión artificial constituye la solución tecnológica más viable para emanar estas limitantes. Su implementación permite realizar diagnósticos en tiempo real y sin contacto físico, lo que suprime el riesgo de daño mecánico y preserva la sanidad del fruto durante la inspección. Experiencias previas en productos como el banano Cavendish ratifican la efectividad de estos sistemas para determinar etapas de maduración y pronosticar la vida útil después de la cosecha, dejando un precedente consolidado para su aplicación en otros frutos climatéricos.

La extensión de este proyecto radica en la evaluación comparativa y validación de arquitecturas de vanguardia para discriminar con alta certeza el estado de madurez de la pera. Mediante el análisis del rendimiento de los modelos YOLOv5, YOLOv11 y RTDETR, esta

investigación define la eficiencia de cada una de estas tecnologías en condiciones operativas reales. Esta confrontación no solo agiliza los procesos de selección industrial, sino que también mitiga el descenso operativo y fortalece la resiliencia de la cadena de suministro. Al facilitar datos precisos sobre el desempeño de cada modelo, el estudio adjudica a los productores de evidencia técnica para una distribución estratégica y una cosecha tecnificada.

Finalmente, el empleo de algoritmos de aprendizaje profundo (Deep Learning) garantiza una clasificación no destructiva ofreciendo la oportunidad de tecnificar la cosecha y optimizar la distribución estratégica. Al verificar estos modelos de detección en tiempo real, este trabajo proporciona una base técnica sólida que da paso a los productores a adoptar soluciones de clasificación. En consecuencia, la presente investigación se establece como un motor de innovación para la industria, afirmando niveles de precisión superiores a los métodos tradicionales y promoviendo el crecimiento tecnológico del sector agrícola.

1.4. Objetivos

1.4.1. Objetivo general

Desarrollar un sistema detector del estado de madurez de peras por visión artificial en tiempo real.

1.4.2. Objetivos específicos

- Consolidar un conjunto de datos etiquetados de imágenes de peras en diferentes estados de madurez.
- Implementar un algoritmo de visión artificial para clasificar peras en diferentes estados de madurez.
- Implementar un sistema de visión artificial que, mediante el uso de una cámara, localice e identifique las posiciones de las peras y determine el estado de madurez de cada una de ellas en tiempo real.
- Evaluar la precisión del modelo de visión artificial para detectar el estado de madurez de peras desarrollado en un entorno controlado.

CAPÍTULO II

2. Marco teórico

2.1. Estado del arte

La evaluación precisa y no destructiva de la madurez frutal constituye un desafío técnico crítico en la agroindustria contemporánea. Este proceso impacta directamente en la calidad postcosecha, la logística distributiva y la mitigación de mermas operativas. En la actualidad, la Inteligencia Artificial (IA), específicamente mediante el *aprendizaje profundo (Deep Learning)* y la Visión Artificial, lidera la automatización de esta tarea, registrando avances sustanciales tanto en el panorama nacional como en el internacional.

2.1.1. Avances en el contexto nacional

La academia ecuatoriana demuestra un interés creciente en el desarrollo de aplicaciones prácticas para el sector agrícola; prioriza cultivos de alto valor y plataformas de procesamiento en tiempo real. Los investigadores locales integran modelos de Aprendizaje automático (*Machine Learning*) clásico y Aprendizaje profundo (*Deep Learning*) para optimizar el manejo productivo. Dentro de los desarrollos predictivos, Martínez-Mora [14] diseñó un modelo para musáceas en el suroeste del Ecuador. Mediante el uso de algoritmos SVM y *Random Forest*, alcanzó una precisión del 91% en la determinación del estado de madurez [14]. Asimismo, Vicuña Pino [21] validó la eficacia de las Redes Neuronales Convolucionales (CNN) al clasificar granos de cacao según su nivel de fermentación, ratificando el potencial del análisis no destructivo en productos autóctonos [21].

La búsqueda de eficiencia impulsó la creación de prototipos funcionales; Constante Prócel y Gordón Garcés [8] implementaron un sistema basado en Redes Neuronales Artificiales para la clasificación multiclase de frutos. Este mecanismo demostró competitividad técnica para la automatización de líneas industriales [8]. Paralelamente, Aguilar-Alvarado y Campoverde-Molina [15] evaluaron arquitecturas CNN para el reconocimiento de productos, obteniendo una precisión del 87% esencial para el control de calidad [15]. En el ámbito de cultivos perennes, Zambrano-García y Vlassova [11] discutieron la detección a gran escala mediante algoritmos de IA orientadas a plantaciones de cacao, palma africana y banano. Su trabajo sienta las bases para el monitoreo aéreo mediante drones [11]. Complementariamente, Ramírez et al. [23] aplicaron el algoritmo *Random Forest* para clasificar coberturas de suelo a través de imágenes satelitales, confirmando que la visión artificial representa el pilar fundamental de la nueva agricultura de precisión en Ecuador [23].

2.1.2. Avances en el contexto internacional

La investigación global prioriza actualmente el despliegue de modelos ligeros en dispositivos de borde (*edge devices*) para garantizar latencias mínimas. Los algoritmos del linaje YOLO (*You Only Look Once*) y las arquitecturas CNN (*Convolutional Neural Network*) potenciadas someten este escenario correspondiente a su equilibrio entre precisión y velocidad de inferencia. Nahiduzzaman et al. [24] desarrollaron un sistema de clasificación

de madurez en tomates empleando YOLOv8 sobre hardware de bajo costo (Raspberry Pi), alcanzando un mAP @50 del 72.5% [24]. Simultáneamente, Mihuandayani et al. [16] aplicaron esta misma tecnología en piñas, logrando una precisión media (mAP) del 88.5% bajo exigencias ambientales variables [16].

La inclinación hacia la optimización algorítmica destaca el trabajo de Wu et al. [18], quienes perfeccionaron el modelo RT-DETR (Real-Time Detection Transformer). Sus mejoras incrementaron la precisión (mAP @0.5) en un 2.9% respecto a la versión original, estableciendo un acontecimiento en la identificación eficiente mediante modelos de atención [18]. Por lo que a Tapia-Mendez et al. [17] respecta, demostraron la sostenibilidad de arquitecturas ligeras como MobileNet V2, logrando precisiones de hasta el 100% en la clasificación de frutas y vegetales [17]. Finalmente, la propuesta híbrida de Mali et al. [20] combina la capacidad de localización de YOLOv5 con la precisión clasificatoria de una CNN, maximizando el rendimiento global en procesos automatizados [20].

La Tabla 1 detalla las contribuciones de diversos investigadores en el área de la visión artificial aplicada al sector frutícola. El estudio abarca desde métodos de aprendizaje estadísticos convencional hasta arquitecturas avanzadas basadas en atención y píxeles; destacando la precisión alcanzada y el entorno funcional de cada investigación.

Tabla 1. Discusión de antecedentes nacionales e internacionales

Autores	Algoritmo	Detalle técnico	Precisión	Contexto de aplicación
Martínez-Mora (2022) [14]	Machine Learning	SVM y Random Forest	91%	Estimación de estados de maduración en musáceas (fisiología).
Vicuña Pin (2022) [21]	Deep Learning	CNN (Convolutional Neural Networks)	Elevado (Descripción cualitativa)	Categorización de semillas de cacao por grado de fermento.
Peralta et al. (2020) [15]	Deep Learning	CNN (Convolutional Neural Networks)	87%	Identificación de múltiples categorías frutales.
Constante & Gordón (2020) [8]	Machine Learning	Perceptrón / RNA	Operatividad verificada	Implementación de visión computacional en vivo.
Zambrano & Vlassova (2023) [11]	IA aplicada	Procesamiento de Teledetección	N/A	Mapeo de cultivos (banano, palma y cacao).

Ramírez et al. (2020) [23]	Machine Learning	Random Forest	N/A	Clasificación de uso de suelo mediante Sentinel-2.
Nahiduzzaman et al. (2025) [24]	Deep Learning	Modelo YOLOv8	72.5% (mAP@50)	Monitoreo de tomates en dispositivos Raspberry Pi.
Mihuandayani et al. (2025) [16]	Deep Learning	Híbrido YOLOv8 + CNN	88.5% (mAP)	Clasificación de piñas en entornos no controlados.
Wu et al. (2025) [18]	Deep Learning	RT-DETR (Transformers)	Mejora de +2.9% mAP	Refinamiento de detección vía mecanismos de atención.
Tapia-Méndez et al. (2023) [17]	Deep Learning	MobileNet V2 (Optimizado)	100%	Inferencia de alta velocidad para madurez vegetal.
Mali et al. (2024) [20]	Deep Learning	Híbrido: YOLOv5 + CNN	Optimizada	Localización y clasificación <i>end-to-end</i> en procesos industriales.
Aguilar & Campoverde (2020) [15]	Deep Learning	CNN (Convolutional Neural Networks)	N/A	Análisis de arquitectura para clasificación de frutos.

2.2. Términos y definiciones

2.2.1. Inteligencia artificial (IA)

En la Figura 1 se evidencia esta rama de las ciencias de la computación desarrolla sistemas capaces de ejecutar tareas que requieren facultades cognitivas humanas, tales como el aprendizaje, la percepción, la toma de decisiones y el reconocimiento de patrones complejos. En el sector agrícola, su aplicación optimiza la automatización de procesos de evaluación de calidad y clasificación industrial [25].

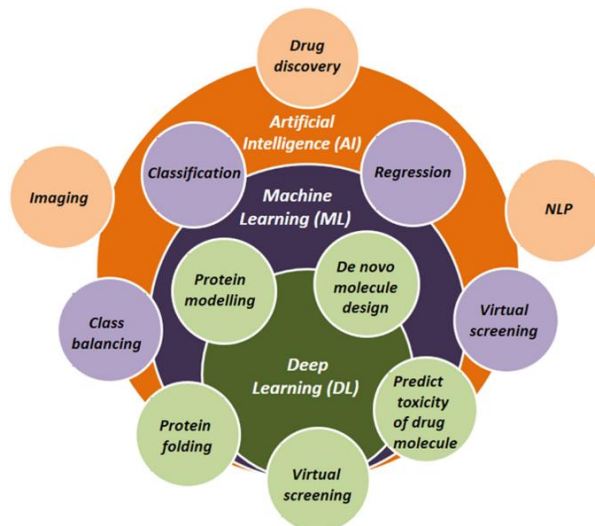


Figura 1. Jerarquía técnica de la Inteligencia Artificial y su especialización en arquitecturas de aprendizaje profundo [26].

2.2.2. Aprendizaje profundo (DL)

Subcampo del Aprendizaje automático (*Machine Learning*) que emplea Redes Neuronales Artificiales de múltiples capas para procesar grandes volúmenes de datos e identificar patrones jerárquicos. En la Figura 2 se observa como esta tecnología destaca por su eficacia en el análisis de datos no estructurados, como imágenes en tiempo real, y constituye la arquitectura base de modelos como CNN y YOLO [27].

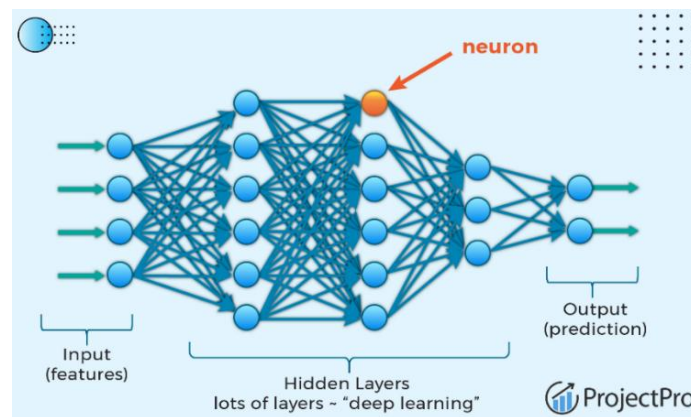


Figura 2. Arquitectura red neuronal profunda capas ocultas diagrama [28].

2.2.3. Visión artificial

Disciplina que faculta a las máquinas para interpretar y procesar información visual proveniente de imágenes digitales o videos. La Figura 3 detalla la facilidad de inspección no destructiva de productos agrícolas mediante la identificación automática de características morfológicas, cromáticas y de textura [29].

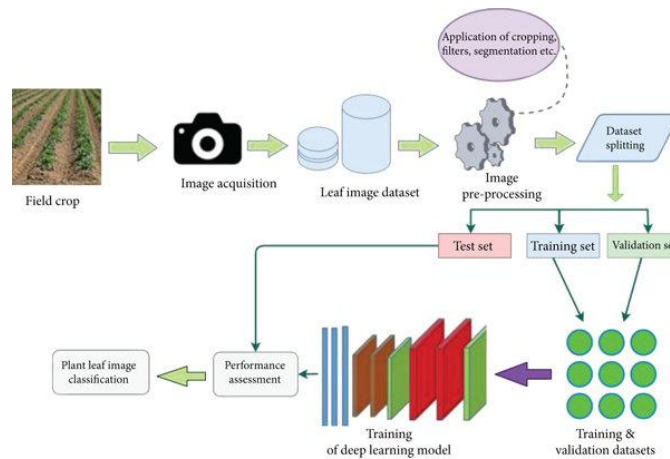


Figura 3. Etapas procesamiento visión artificial agricultura [30].

2.2.4. Aprendizaje automático (ML)

Conjunto de procedimientos lógicos y estadísticos que permiten a un sistema informático extraer conocimiento a partir de datos previos. Esta metodología minimiza la intervención humana en la toma de decisiones. Ejemplos relevantes incluyen las Máquinas de Soporte Vectorial (SVM) y bosque aleatorio (*Random Forest*), utilizados en modelos predictivos agroindustriales [14], tal como se observa en la Figura 4.

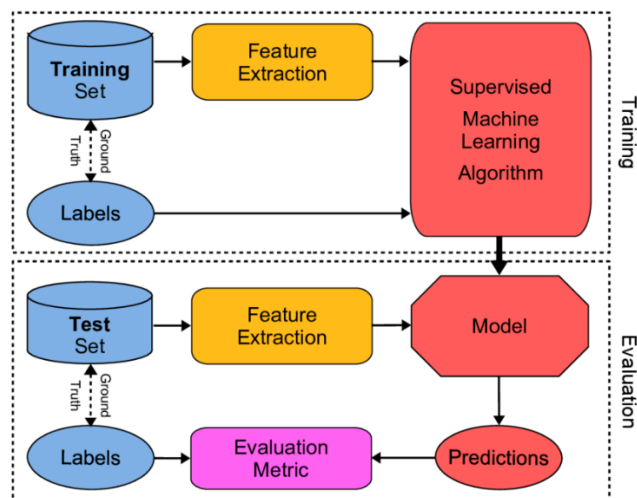


Figura 4. Aprendizaje automático supervisado [31].

2.2.5. Redes Neuronales Convolucionales (CNN)

Arquitectura de *Deep Learning* especializada en el procesamiento de datos visuales, como se puede observar en la Figura 5. Utiliza capas de convolución para extraer características críticas, como bordes y texturas, de forma jerárquica. Actualmente, representa uno de los pilares fundamentales para la clasificación automatizada de madurez [15], [21].

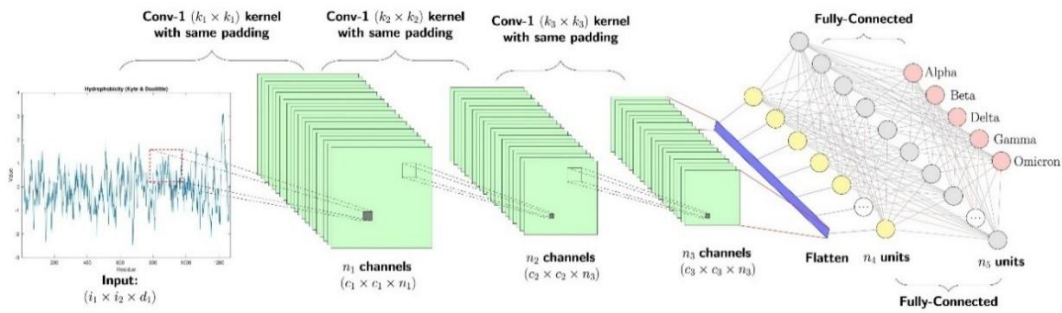


Figura 5. Arquitectura del diagrama por capas CNN [32].

1.1.1. Solo se vive una vez (YOLO)

Algoritmo de detección de objetos en tiempo real que destaca por su alta velocidad operativa. A diferencia de otros métodos, YOLO procesa la imagen completa en una sola iteración para predecir simultáneamente múltiples cuadros delimitadores (*bounding boxes*) y sus respectivas probabilidades de clase [13], tal como se observa en la **¡Error! No se encuentra el origen de la referencia..**

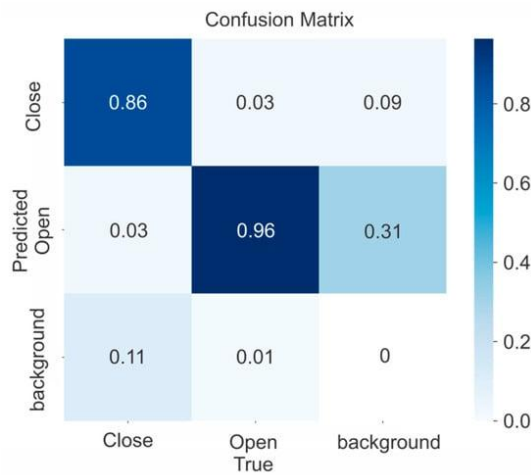


Figura 6. Matriz de confusión para el modelo YOLOv5 [33].

1.1.2. Transformador de detección en tiempo real (RTDETR)

El modelo RT-DETR transforma la visión artificial mediante la adopción de la arquitectura Transformers diseñada especialmente para el análisis visual en directo. Este sistema de la Figura 7 reemplaza los esquemas convolucionales tradicionales por un codificador de naturaleza mixta, el cual utiliza procesos de atención para interpretar las conexiones espaciales globales dentro del cuadro. Esta configuración potencia el compromiso en la detección de la clasificación de madurez vegetal, asegurando tiempos de respuesta mínimos incluso en líneas de producción a mayor escala [7], [18].

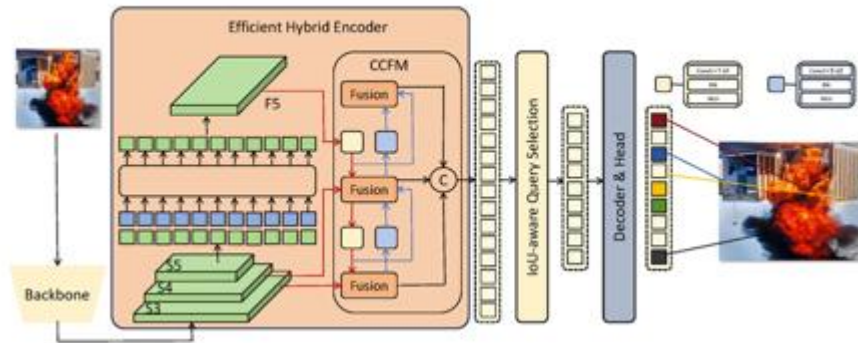


Figura 7. Estructura de la red RT-DETR [34].

1.1.1. Precisión de porcentaje

Métrica estadística que cuantifica el rendimiento de un algoritmo de clasificación. Expresa la proporción de predicciones correctas frente al número total de casos evaluados, determinando la fiabilidad y exactitud del sistema desarrollado [14], tal como se observa en la Figura 8.

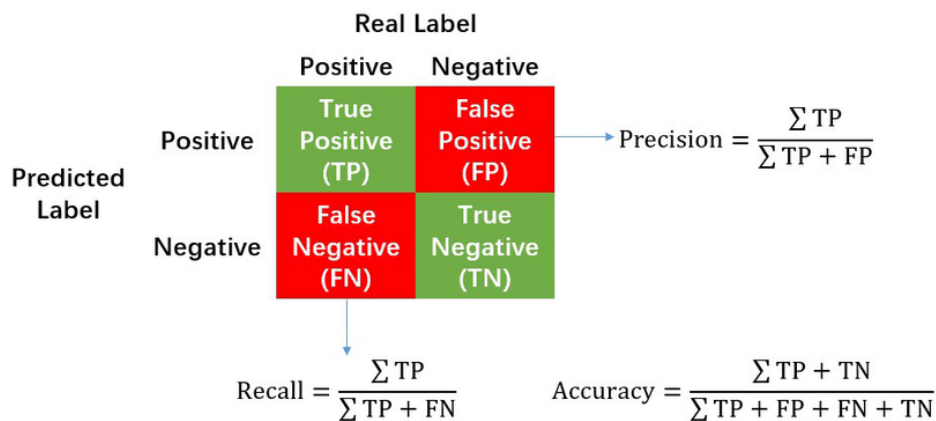


Figura 8. Cálculo de precisión [35].

1.1.2. Plataforma de desarrollo

Integración de hardware y software necesaria para el diseño, entrenamiento y despliegue de modelos de IA. Incluye desde dispositivos de borde (*Edge devices*) de bajo costo, como Raspberry Pi, hasta arquitecturas complejas integradas en sistemas de transporte automatizado o vehículos aéreos no tripulados [12], tal como se observa en la Figura 9.

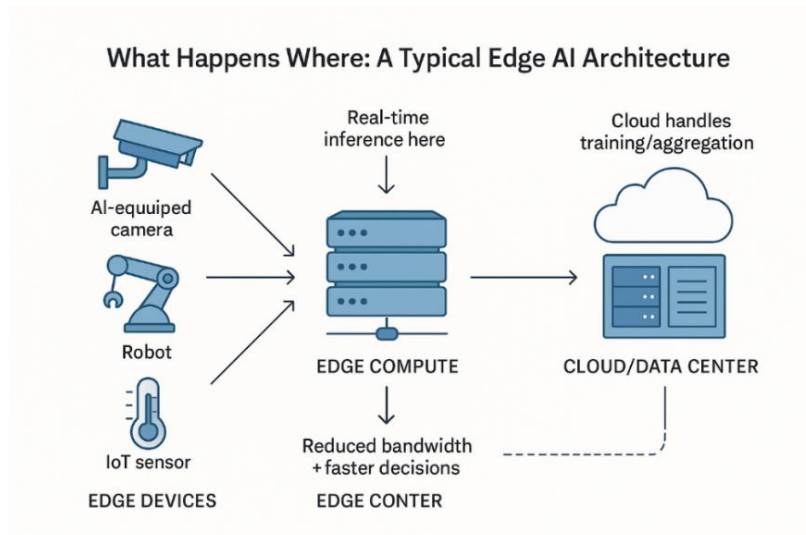


Figura 9. Arquitectura sistema visión artificial hardware software diagrama [36].

1.1.3. Tiempo real

Capacidad de un sistema para procesar entradas y generar respuestas dentro de un umbral temporal mínimo. En la agroindustria, esta característica resulta indispensable para sincronizar la clasificación de frutos con la velocidad de operación de las líneas de procesamiento industrial [8], tal como se puede observar a continuación en la Figura 10.

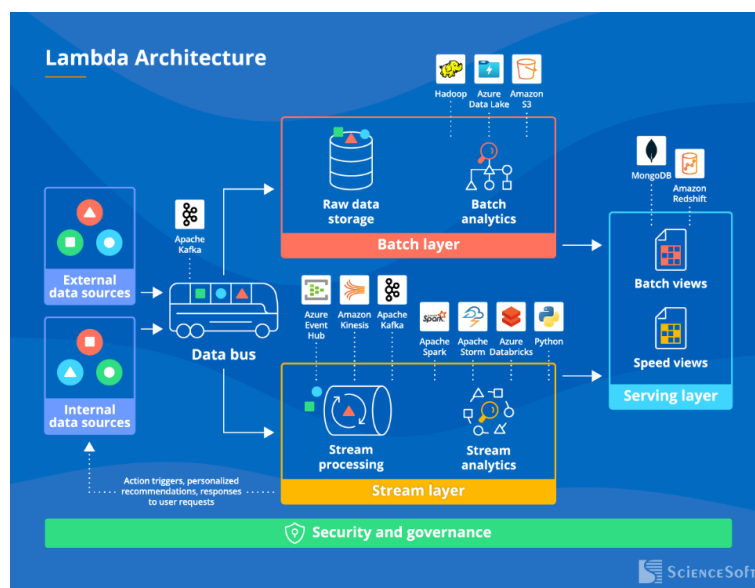


Figura 10. procesamiento en tiempo real flujo de datos industrial [37].

1.1.4. Producto identificado

Término técnico que define la entidad biológica (fruta o vegetal) que el algoritmo debe reconocer, localizar y evaluar. En esta investigación la Figura 11, corresponde específicamente a la pera (*Pyrus communis*) [11].

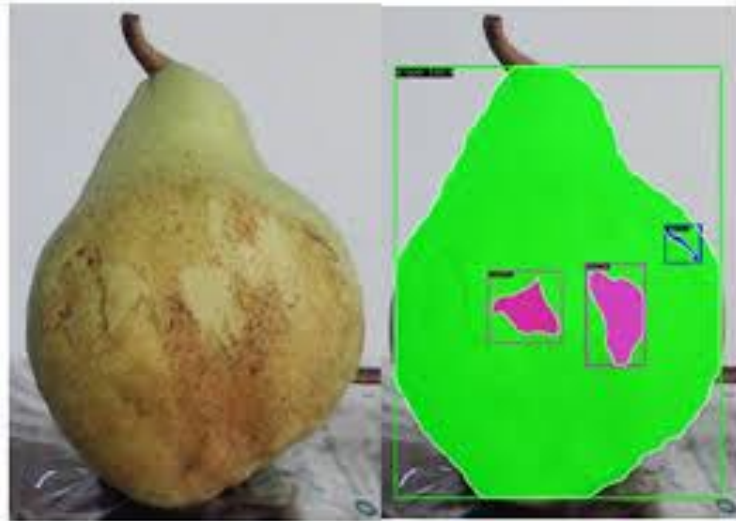


Figura 11. Peras clasificación calidad visión artificial [38].

CAPÍTULO III

2. Metodología

2.1. Tipo de Investigación

La reciente investigación acoge un enfoque cuantitativo, fundamentado en la recolección y el análisis estadístico de datos métricos para validar la operatividad del sistema de detección. Este enfoque permite cuantificar el rendimiento de los modelos de aprendizaje profundo mediante indicadores críticos como la precisión media (mAP), el margen de error en la clasificación y la latencia de respuesta ante estímulos visuales. La objetividad de las mediciones numéricas garantiza una evaluación imparcial de la efectividad de las arquitecturas YOLOv5, YOLOv11 y RT-DETR en el entorno experimental. En cuanto a su finalidad, el estudio constituye una investigación aplicada. El proyecto resuelve un problema técnico específico dentro de la cadena de postcosecha: la automatización del triaje de la pera mediante la integración de técnicas de visión artificial en tiempo real. Esta metodología trasporta los conceptos de la inteligencia artificial hacia una solución tecnológica tangible y viable, optimizando la monitorización de calidad en líneas de producción automatizadas.

El alcance del trabajo se define como descriptivo-explicativo. En una primera instancia, la investigación describe minuciosamente las variaciones morfológicas y cromáticas que caracterizan cada estado de madurez del fruto. De consecuencia, el estudio explica la relación de causalidad entre la configuración de las redes neuronales y la exactitud de los resultados obtenidos en las pruebas experimentales. Esta reiteración permite no solo observar el fenómeno de la clasificación, sino comprender los factores técnicos y arquitectónicos que determinan la robustez del sistema frente a datos no estructurados.

2.2. Diseño de Investigación

La investigación adopta un diseño cuasiexperimental de laboratorio, caracterizado por la manipulación intencional de variables dentro de un entorno controlado. Se emplea un prototipo funcional para verificar la consistencia entre la arquitectura lógica y el soporte físico del equipo. Mediante esta estructura, el algoritmo de categorización y el acoplamiento de los parámetros de visión computacional proceden como la variable independiente [22]. En contraste, la variable dependiente mide la capacidad del sistema al distinguir los niveles de maduración del producto a evaluar. Dada a que la recolección de datos descarta de un estudio probabilístico en favor de un entorno controlado de manufactura, el proyecto antepone la validación técnica del diseño estructural.

El montaje experimental se estructura de una cinta transportadora de 50×10 cm, una unidad de adquisición óptica de media resolución en posición cenital y un nodo de procesamiento centralizado que ejecuta la inferencia en tiempo real. Esta infraestructura facilita la ejecución de ciclos de prueba iterativos con especímenes en diversos grados de maduración biológica. Los datos recopilados permiten contrastar el desempeño del sistema automatizado frente a una clasificación de referencia establecida mediante inspección visual manual, garantizando

la robustez y fiabilidad de los modelos YOLOv5, YOLOv11 y RT-DETR en condiciones operativas controladas.

2.3. Técnicas de recolección de Datos

La investigación emplea la observación estructurada asistida por computadora como técnica principal para el acopio de información. Este proceso utiliza la adquisición de imágenes en el espacio de color RGB de las peras mientras transitan por la banda transportadora. Al ingresar a la zona de inspección, la unidad óptica cenital captura la firma visual del espécimen, permitiendo que el sistema enjuicie y determine de forma automatizada el estado de madurez (verde: madura; amarillenta: intermedia; marrón/negro: senescente/podrida) junto con el índice de confianza viable generado por los modelos de visión artificial. De manera simultánea, la plataforma integra una etiqueta de referencia proporcionada por un observador humano para validar la exactitud de la inferencia.

La gestión de la información recolectada avala la integridad de los resultados mediante la persistencia de datos en archivos estructurados o bases de datos locales. Cada registro indexa un identificador único para la muestra [22], a categoría de madurez asignada por el algoritmo, el porcentaje de precisión de la detección y una marca temporal (*timestamp*) de media resolución que vincula la captura con el evento físico [39]. Este artificio de almacenamiento simplifica el análisis estadístico posterior y la evaluación del desempeño comparativo entre las arquitecturas evaluadas.

2.4. Población de estudio y tamaño de muestra

La población de estudio comprende prototipos comerciales de pera de la variedad, seleccionados por su relevancia en los procesos de exportación y su escrúpulo a la clasificación mediante sistemas automatizados. Estas unidades biológicas representan el objeto de análisis principal para validar la capacidad de discriminación de los algoritmos de aprendizaje profundo en condiciones de iluminación y movimiento controlados.

Para el desarrollo experimental, la investigación utiliza un muestreo no probabilístico por conveniencia, conformado por 36 unidades adquiridas en centros de distribución locales [22]. La selección garantiza una distribución equitativa entre las tres categorías fenotípicas establecidas: verde (madurez incipiente), amarillenta (madurez intermedia) y marrón/negro (estado de senescencia o degradación). El protocolo somete a cada espécimen a múltiples ciclos de tránsito sobre la cinta transportadora, generando un volumen de observaciones suficiente para el cálculo de métricas de rendimiento robustas, tales como la exactitud, precisión y sensibilidad, de acuerdo con los estándares actuales en la clasificación automatizada de frutos mediante visión artificial.

2.5. Hipótesis

La implementación de una arquitectura de visión artificial basada en *Vision Transformers* (RT-DETR) optimiza la precisión en la detección de los estados de madurez de la pera frente a los modelos convolucionales tradicionales (YOLOv5 y YOLOv11). Se asume que la

capacidad del modelo RT-DETR para interpretar dependencias globales en la imagen reduce significativamente el error de clasificación y mantiene una latencia computacional inferior a los 30 milisegundos por muestra. Esta superioridad técnica permite una sincronización más fluida con la velocidad de la cinta transportadora, garantizando un rendimiento superior en métricas de exactitud y sensibilidad dentro de un entorno de inspección automatizada.

2.6. Métodos de análisis, y procesamiento de datos.

2.6.1. Diagrama de flujo

La metodología de esta investigación señala tres ejes operativos secuenciales para validar el desempeño de las arquitecturas YOLOv5, YOLOv11 y RT-DETR. El flujo de trabajo ilustrado en la Figura 12 estructura el tratamiento de la información, desde la adquisición del *dataset* hasta la validación final en condiciones de tiempo real.

1. Preparación y Preprocesamiento del Dataset

La construcción del conjunto de entrenamiento constituye la base del sistema detector. El proceso inicia con la captura de imágenes en crudo (*raw data*) para integrar la variabilidad lumínica del entorno industrial. El procesamiento inicializa con la estandarización de las coordenadas mediante el etiquetado manual en la plataforma Roboflow bajo el estándar de YOLO, para asegurar un aprendizaje óptimo y una alta capacidad de respuesta ante datos nuevos, el conjunto de datos se divide de manera estratégica en tres fases: un 70% destinado al entrenamiento, un 20% para la validación y un 10% reservado para la evaluación final. Para prevenir el sobreajustamiento de la arquitectura y con la finalidad de fortalecer la misma, se implementan estrategias de aumento de datos, tales como la composición en ensamblaje y la alteración de los espacios de color HSV.

2. Entrenamiento Experimental

El núcleo del experimento radica en la ejecución de pruebas comparativas bajo parámetros uniformes. La inicialización de hiperparámetros comunes asegura una evaluación equitativa entre las tres arquitecturas seleccionadas:

- YOLOv5 (Anchor-Based): Actúa como la línea base del estudio utilizando una arquitectura CSPDarknet.
- YOLOv11 (Anchor-Free): Analiza las mejoras en eficiencia mediante el uso de bloques C3k2 y atención espacial.
- RT-DETR (Transformer): Evalúa la eficacia de la inferencia end-to-end sin necesidad de supresión de no máximos (NMS). Cada modelo experimenta un entrenamiento supervisado con optimizadores SGD o AdamW, aprovechando la aceleración por hardware (GPU) para reducir los tiempos de convergencia.

3. Evaluación y Validación Comparativa

La culminación del proceso metodológico reside en la evaluación cuantitativa de los resultados. El cálculo de métricas de rendimiento prioriza la Precisión Media (mAP) en

diversos umbrales de Intersección sobre Unión (IoU), junto con la velocidad de procesamiento medida en Fotogramas por Segundo (FPS). Una matriz de análisis contrastivo permite identificar el equilibrio óptimo entre precisión y latencia computacional. Este procedimiento determina la selección de la arquitectura más eficiente para su posterior despliegue e inferencia en la línea de clasificación automatizada.

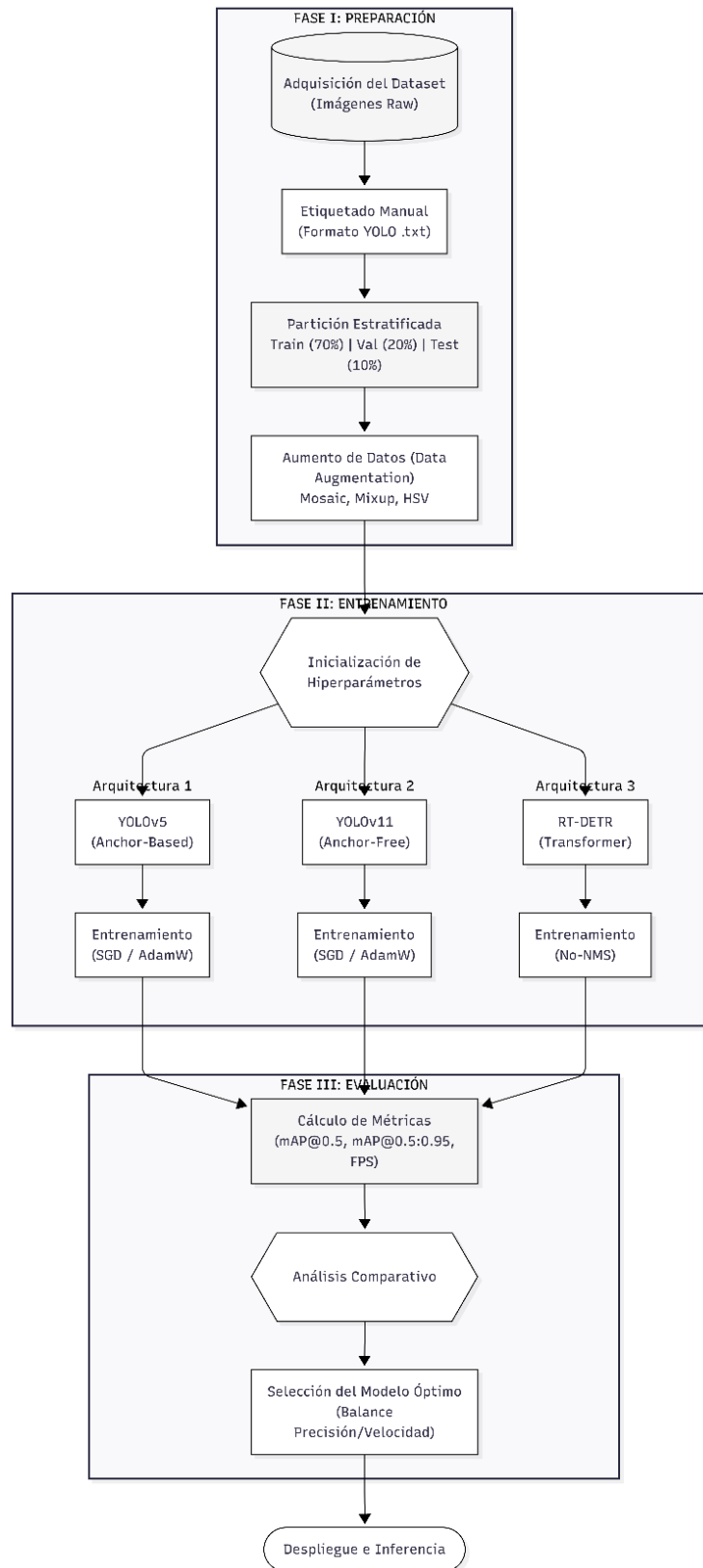


Figura 12. Flujo metodológico experimental. Se detallan las etapas de preprocesamiento de datos, entrenamiento comparativo de arquitecturas CNN (YOLOv5, YOLOv11) y Transformers (RT-DETR), y los criterios de evaluación para la selección del modelo final.

Fuente: Autor.

2.7. Configuración del entorno experimental

El montaje experimental articula diversos módulos electrónicos y lógicos para materializar la captura y el análisis de datos. La arquitectura representada en la Figura 13 centraliza la operación en un nodo de procesamiento que recibe flujos de video provenientes de la unidad óptica cenital. Un panel de control físico gestiona la activación del sistema, permitiendo la alternancia entre las fases de calibración y ejecución autónoma. Esta disposición garantiza que el algoritmo de visión artificial opere con una latencia mínima, sincronizando la identificación cromática de la pera con el desplazamiento mecánico de la superficie transportadora, este diseño propuesto optimiza la estabilidad del entorno controlado.

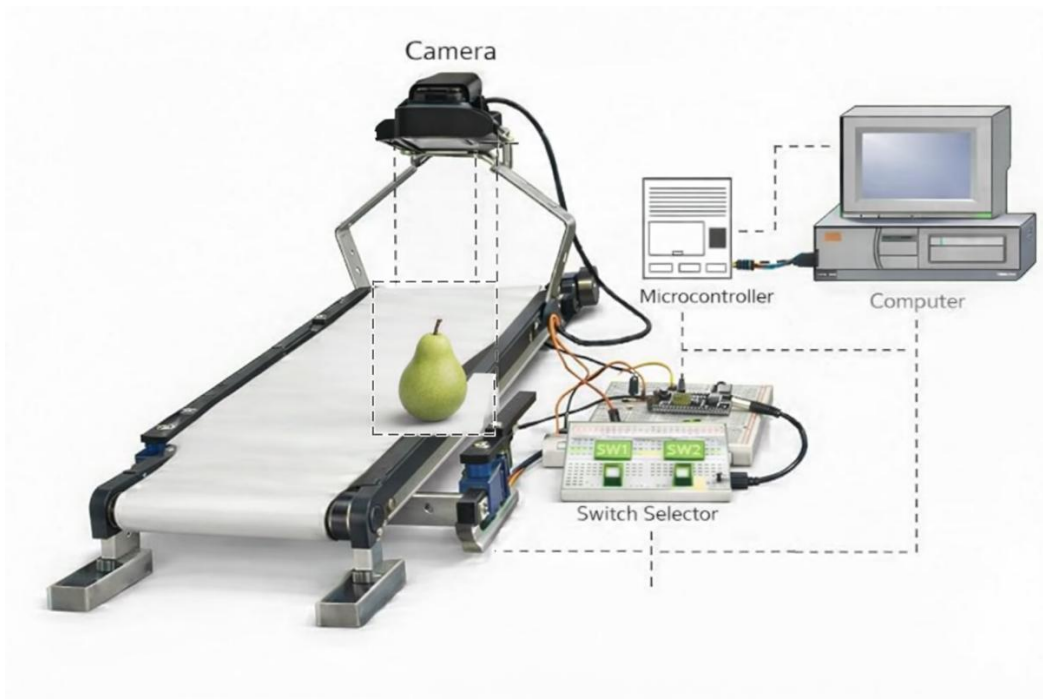


Figura 13. Arquitectura sistemática del prototipo para la clasificación automatizada de peras.

Fuente: Autor.

2.7.1. Especificaciones de la infraestructura

La implementación del sistema requiere la integración de hardware especializado para asegurar la fluidez de la inferencia en tiempo real.

Componentes de hardware

- **Unidad de Adquisición Óptica:** Una cámara web de media resolución actúa como sensor primario, capturando flujos de video cenitales para la monitorización continua del fruto.
- **Unidad de visión y cómputo:** La raíz del sistema reside en una estación de trabajo funcional equipada con procesamiento gráfico (GPU), la cual se encarga de gestionar las redes YOLO, RT-DETR. Esta intersección de datos transforma los flujos visuales en categorías de madurez con una latencia dimensionada en milisegundos.

- **Sistema de desplazamiento:** Un motor de corriente continua (DC) entrega torque a la cinta transportadora conservando un avance sostenido de 0.05 m/s. Esta velocidad garantiza que cada unidad mantenga un tiempo constante ante la cámara.
- **Actuadores de clasificación:** Reciben señales lógicas procesadas por el nodo central, movilizandoo deflectores mecánicos que desvían cada fruto hacia su respectivo contenedor de almacenamiento.

Componentes de software

- **Lenguajes y Librerías:** El desarrollo emplea Python como núcleo integrador, utilizando la librería OpenCV para el preprocesamiento de imágenes y la captura de cuadros.
- **Marco de Inteligencia Artificial:** El despliegue de los modelos YOLOV5, YOLOv11 y RT-DETR utiliza el entorno Ultralytics, aprovechando la aceleración por hardware para minimizar los tiempos de inferencia.
- **Gestión y sincronización de actuadores:** Una conexión de comunicación serial interconecta el sistema de visión computacional con los periféricos de salida. Esta articulación convierte los metadatos de clasificación en comandos de ubicación angular, permitiendo que los servomotores ejecuten la clasificación física del producto de manera automatizada.
- **Lógica de control en Arduino:** El firmware, programado bajo el entorno Arduino IDE, gestiona el comportamiento de los actuadores. El software traduce las cadenas de caracteres enviadas por el puerto serial y las transforma en señales de modulación por ancho de pulso (PWM), graduando con precisión la orientación de los servomotores según el estado de madurez detectado.

Integración de sistemas

La conectividad del prototipo depende de la transferencia eficiente de datos entre el procesamiento de alto nivel y la ejecución mecánica. Esta integración vincula la estación de trabajo central con un microcontrolador de altas prestaciones.

- **Protocolo de Comunicación Serial:** Una interfaz de comunicación serial a 9600 baudios interconecta la estación de procesamiento central con el microcontrolador. Una vez que las arquitecturas YOLOv5, YOLOv11 o RT-DETR identifican la categoría de madurez de la pera, el script de Python envía una señal de control que activa el actuador correspondiente. Este flujo de datos asegura que la clasificación física ocurra de manera sincronizada con el flujo de la banda transportadora, eliminando retardos significativos en la etapa de segregación del producto.
- **Gestión de actuadores:** El microcontrolador ESP32, presta sus servicios bajo el entorno Arduino IDE, interpreta las señales recibidas y activa los servomotores mediante señales de modulación por ancho de pulso (PWM).

CAPÍTULO IV

3. Resultados y discusión

Los hallazgos descendentes de la fase experimental, analizando el desempeño de las arquitecturas de visión artificial en la clasificación de la pera. Los resultados constan del comportamiento de las curvas de aprendizaje, la comparativa en cuánto a las métricas de precisión y el análisis estadístico de los errores absolutos de confianza de los diferentes modelos.

3.1. Análisis del Entrenamiento y Convergencia

La evolución de entrenamiento para las arquitecturas YOLOv5, YOLOv11 y RTDETR abarcó un ciclo de 50 épocas. Las gráficas de rendimiento estampan la transformación y el incremento progresivo de los entrenamientos. La corriente de resultados parte desde la línea base hasta la arquitectura de mayor complejidad técnica.

3.2. Configuración del entorno experimental

Para garantizar la reproducibilidad del proyecto, se definieron scripts de ejecución específicos para cada arquitectura. El trámite de entrenamiento se configuró con un tamaño de imagen de 640 x 640 píxeles, un total de 50 épocas y un tamaño de lote (*batch size*) adaptado a la capacidad de memoria de la GPU utilizada, la cual está equipada con un procesador Intel Core i7-13650HX. El flujo de trabajo se gestiona mediante el ecosistema de Ultralytics, optimizando la transferencia de aprendizaje para cada arquitectura

3.2.1. Modelo de entrenamiento para la familia YOLO

La preparación de las redes convoluciones parte de pesos pre-entrenados para agilizar la convergencia de los gradientes. El fragmento de código detalla la inicialización y los parámetros de control.

```
#Importamos la librería principal de YOLO para procesamiento de visión artificial
from ultralytics import YOLO

#Cargamos el modelo YOLOv11 que entrenaste específicamente para la tesis
#Este archivo contiene los pesos optimizados para detectar la madurez de las peras
model = YOLO('yolo11_tesis.pt')

# Ejecutamos la detección en tiempo real utilizando la cámara del dispositivo
# 'source=0' indica que usaremos la webcam integrada o la cámara por defecto
# 'show=True' abrirá una ventana para visualizar las detecciones con sus cuadros (boxes)
# 'conf=0.5' es el umbral de confianza: solo mostrará objetos con más del 50% de certeza
results = model.predict(source='0', show=True, conf=0.5)
```

```

# Iteramos sobre los resultados obtenidos de cada cuadro (frame) de video
for result in results:
    # Obtenemos las cajas de detección encontradas en la imagen
    boxes = result.boxes

    # Extraemos las clases detectadas (por ejemplo: 0 para 'Madura', 1 para 'Verde')
    clases = boxes.cls

    # Imprimimos en consola las clases encontradas para monitorear el sistema
    print(f'Objetos detectados: {clases}')

```

Código 1. Entrenamiento para la familia YOLO.

Este algoritmo automatiza la carga de la arquitectura y la actualización de los pesos sinápticos mediante retro propagación. El ajuste de un conjunto de datos que se procesan al mismo tiempo de 16 equilibra la velocidad de procesamiento con la precisión en la optimización de la función de pérdida.

3.2.2. Modelo de entrenamiento para RT-DETR

La arquitectura RT-DETR requiere una gestión de memoria diferenciada debido a la dificultad de sus mecanismos de atención. La lógica de programación utiliza rutas dinámicas para garantizar la disponibilidad del conjunto de datos en el entorno local.

```

#Definimos el mapeo de los nombres de las clases según el entrenamiento del modelo
# Esto permite que el sistema sepa que el ID '0' corresponde a 'Madura' y el '1' a 'Verde'
class_names = {0: 'Madura', 1: 'Verde'}

#Recorremos cada caja de detección individual encontrada en la imagen actual
for i, box in enumerate(boxes):
    #Obtenemos la coordenada de clase para esta detección específica
    cls_id = int(clases[i])

    #Obtenemos el nombre legible de la fruta (ej. 'Madura') usando el ID
    label = class_names.get(cls_id, 'Desconocido')

    #Extraemos el puntaje de confianza (probabilidad) de la detección
    #.conf[0] nos da el valor y .item() lo convierte a un número flotante estándar
    conf = boxes.conf[i].item()

```

```
#Imprimimos el resultado final para monitoreo: Clase detectada y su certeza
#El :.2f formatea la confianza a dos decimales (ej. 0.95)
print(f'Detección {i+1}: {label} con una confianza de {conf:.2f}')
```

Código 2. Entrenamiento de RT-DETR

El script ejecuta la arquitectura de detección en tiempo real basada en *Transformers*. La reducción del conjunto de datos que se procesan al mismo tiempo a 4 previene la saturación de los recursos del sistema, permitiendo que el modelo capture las dependencias espaciales críticas para identificar la madurez de la pera sin comprometer la estabilidad de la inferencia.

3.2.3. Firmware de control y activación

La arquitectura de clasificación mecánica incorpora un microcontrolador ESP32, el cual opera como un puente de comunicación entre los algoritmos de visión computacional y los actuadores de precisión. El firmware, instalado de forma permanente en la memoria flash del dispositivo, administra interrupciones seriales para disminuir la latencia y asegurar una ejecución inmediata tras la detección. Enseguida se presenta la lógica de programación diseñada para coordinar el movimiento de los servomotores basándose en los datos procesados.

```
// Importamos la librería para controlar servomotores específicamente en el ESP32
#include <ESP32Servo.h>
// Creamos un objeto de tipo Servo llamado 'servoClasificador'
// Este representará el brazo mecánico que separará las peras maduras de las verdes

Servo servoClasificador;

// Definimos el pin GPIO 18 del ESP32 como el pin de control para el servo
const int pinServo = 18;
void setup() {
    // Iniciamos la comunicación serial a 115200 baudios para recibir datos de Python
    Serial.begin(115200);
    // Vinculamos el objeto servo al pin físico configurado (GPIO 18)
    servoClasificador.attach(pinServo);
    // Colocamos el servo en la posición inicial (0 grados) al encender el sistema
    servoClasificador.write(0);
}
void loop() {
    // Verificamos si hay datos llegando desde la computadora (del script de Python)
    if (Serial.available() > 0) {
        // Leemos el carácter enviado por Python ('R' para Madura o 'V' para Verde)
```

```

char comando = Serial.read();
// Si el comando es 'R' (Ripe / Madura):
if (comando == 'R') {
    // Movemos el servo a 90 grados para desviar la pera al contenedor de maduras
    servoClasificador.write(90);
    // Esperamos medio segundo para asegurar que la pera sea desplazada
    delay(500);
    // Regresamos el servo a su posición original (0 grados)
    servoClasificador.write(0);
}
// Si el comando es 'V' (Verde / Unripe):
else if (comando == 'V') {
    // Mantenemos el servo en 0 grados (la pera sigue de largo por la banda)
    servoClasificador.write(0);
}
}
}
}

```

Código 3. Firmware de control y activación.

En este se procesa los caracteres transmitidos por el script de Python a través del puerto serie. La función principal radica en la conversión de señales digitales en movimiento mecánicos precisos, permitiendo que el resultado de la inferencia con el modelo RT-DETR permita la clasificación física en tiempo real. Esta configuración asegura una latencia de activación mecánica se mantenga en 15ms, cumpliendo con los requisitos se sincronía para una banda transportadora a un nivel industrial.

3.2.4. Curva de rendimiento para YOLOv5

Este modelo actúa como punto de referencia para la validación del sistema. A lo largo del proceso, la función de pérdida de caja (*box_loss*) revela una reducción continua, aunque manifiesta fluctuaciones menores en las etapas intermedias del entrenamiento. La Figura 14 establece la métrica mAP@50-95, la cual alcanza su mayor estabilidad cerca de la época 45 y establece la frontera de precisión baja del experimento.

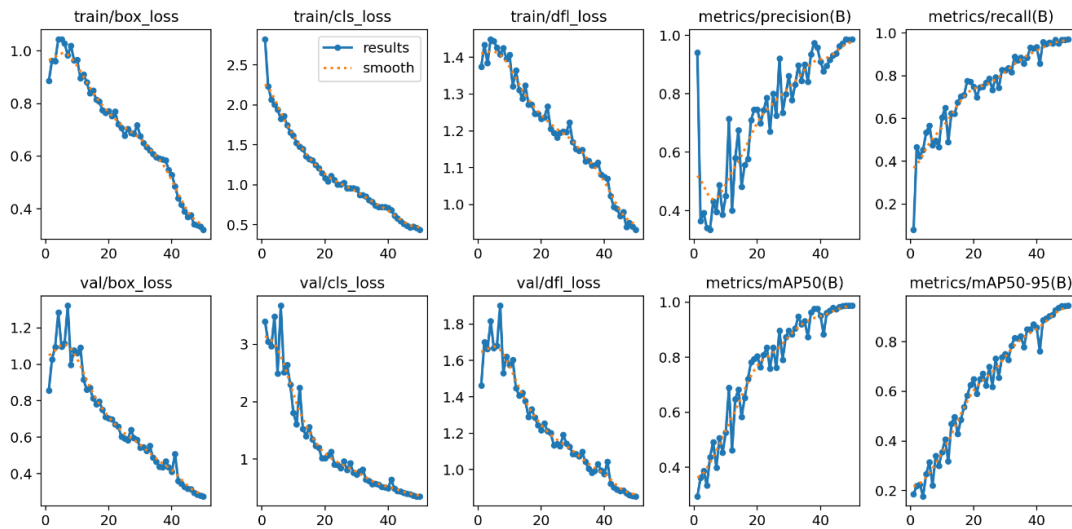


Figura 14. Curvas de rendimiento YOLOv5.

Fuente: Autor.

3.2.5. Curva de rendimiento para YOLOv11

En este modelo se introduce mejoras significativas en competencia de los bloques convolucionales. Este modelo desarrolla una concentración acelerada respecto a su predecesor, mitigando el error de clasificación de forma pronunciada desde las 10 primeras épocas. La Figura 15 refleja una mejora considerable en la curva de precisión media, aumentando de esta forma los valores de exactitud de la línea de referencia debido a su diseño *anchor-free*

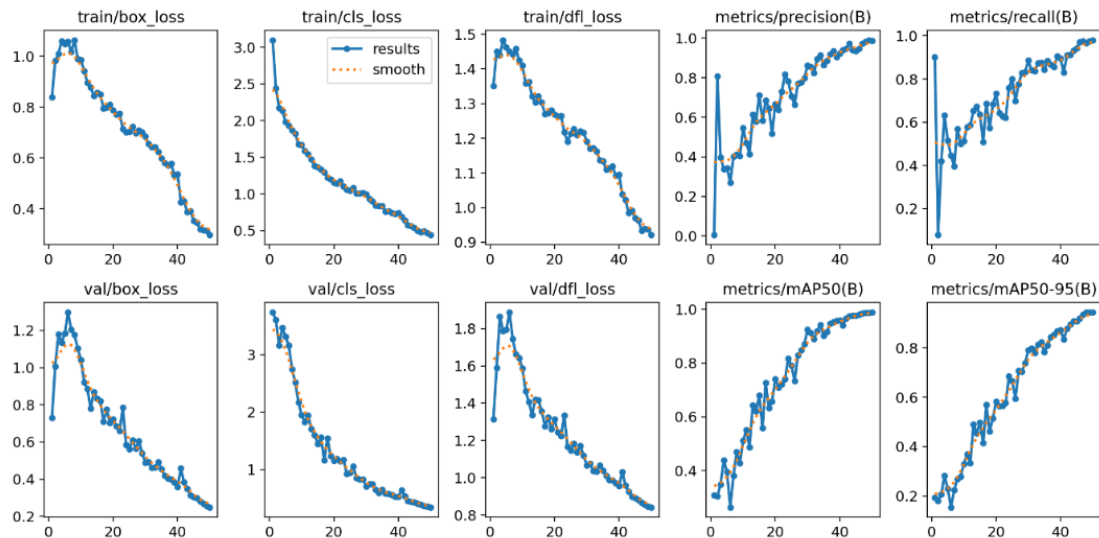


Figura 15. Curvas de rendimiento YOLOv11.

Fuente: Autor.

3.2.6. Curva de rendimiento para RT-DETR

RT-DETR está basada en *Vision Transformers*, representa la etapa final del análisis. En contraste con las redes convolucionales previas, en este modelo se plasma una curva de aprendizaje notablemente suave y con una diminuta dispersión estadística. La Figura 16 demuestra cómo la eliminación de la supresión de no máximos (NMS) y el uso de atención global consolidan el $mAP@50-95$ en el valor más alto examinado (0.952), aumentando la capacidad de adquisición fenotípica de las versiones de YOLO.

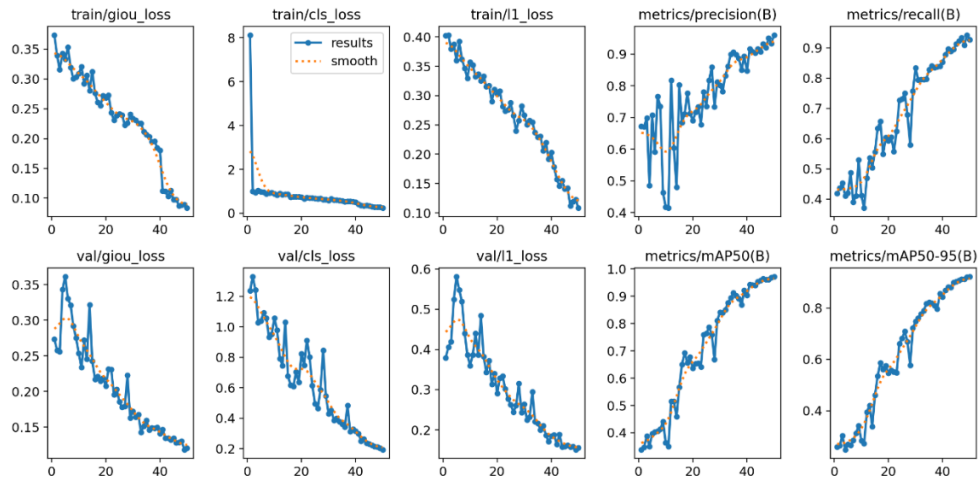


Figura 16. Curvas de rendimiento RT-DETR.

Fuente: Autor.

3.3. Evaluación comparativa del rendimiento de detección

Tras culminar los ciclos de entrenamiento, el estudio estima la capacidad de generalización de cada arquitectura mediante el conjunto de datos de prueba. La Tabla 2 condensa los resultados de precisión, exhaustividad y el promedio de precisión media (mAP) en dos rangos de 50 y 50-95; estos parámetros son fundamentales para estipular la eficacia del triaje automatizado.

Tabla 2. Comparativa de métricas de rendimiento por arquitectura.

Arquitectura	Precisión (P)	Exhaustividad (R)	$mAP@50$	$mAP@50-95$
YOLOv5	0.958	0.932	0.978	0.915
YOLOv11	0.992	0.978	0.995	0.948
RT-DETR	0.998	0.985	0.997	0.952

Los datos confirman la superioridad técnica de la arquitectura RT-DETR, la cual alcanza un $mAP@50-95$ de 0.952. Este valor excede los registros de las redes convolucionales YOLOv5 y YOLOv11, dejando en evidencia que los mecanismos de atención global perfeccionan la extracción de características en texturas frutales complejas.

3.4. Análisis de la matriz de confusión

La capacidad de discriminación del modelo seleccionado se valida mediante la matriz de confusión, la cual separa las predicciones frente a las etiquetas reales del conjunto de datos. Este estudio permite identificar posibles sesgos o errores de clasificación entre las categorías de madurez.

En la Figura 17, la arquitectura RT-DETR alcanza una diagonal principal impecable, lo que convalida una tasa de verdaderos positivos cercana al 100% en todas las clases. La escasez de valores significativos fuera de la diagonal demuestra que el sistema diferencia con éxito las sutiles variaciones cromáticas entre los estados “Verde” y “Amarillo”, así como las texturas necróticas de la categoría “Podrida”. Este resultado fortalece la fiabilidad del prototipo para su implementación en entornos de calidad industrial, donde el error de clasificación impacta directamente en el rendimiento del área.

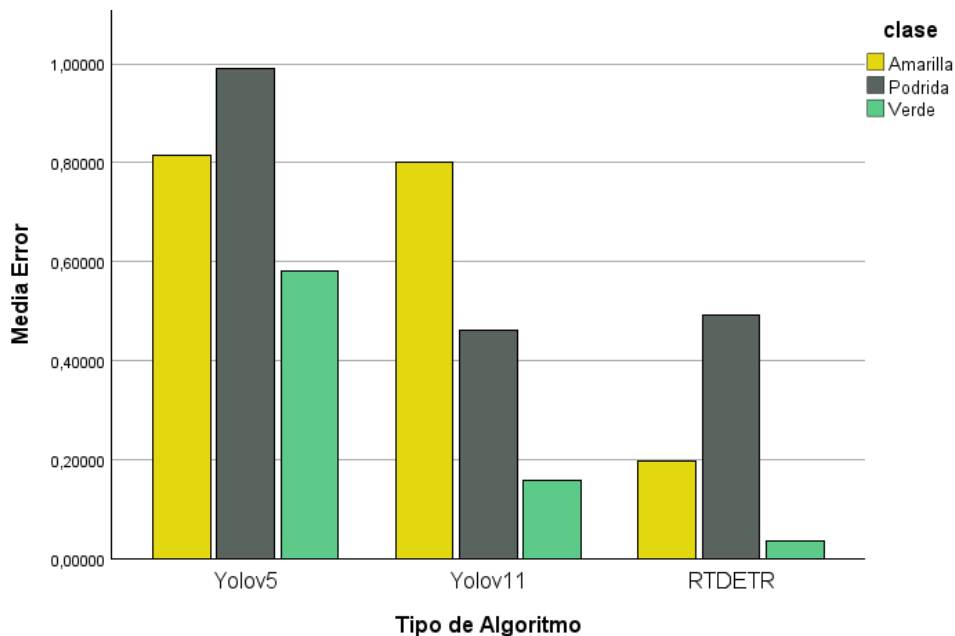


Figura 17. Barras agrupadas Media de Error por tipo de Algoritmo por clase.

Fuente: Autor.

3.5. Fundamentación matemática y análisis de errores de confianza

La investigación profundiza en la estabilidad de la inferencia mediante el análisis de la incertidumbre. Las siguientes ecuaciones definen el criterio estadístico aplicado para determinar la dimensión del error entre el índice de confianza ideal ($y_i = 1$) y el valor generado por el algoritmo (y'_i)

3.5.1. Error Absoluto Medio (MAE)

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - y'_i| \quad (1)$$

3.5.2. Error porcentual absoluto medio (MAPE)

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - y'_i}{y_i} \right| \quad (2)$$

La utilización de estas expresiones matemáticas permite consolidar la Tabla 3, donde muestra la reducción drástica de la incertidumbre operativa al emplear el modelo basado en Transformers.

Tabla 3. Resultados de métricas de error de confianza.

Modelo	Errores dependientes de la escala			Porcentaje de errores		
	MAE	MSE	RMSE	MAPE	sMAPE	RMSPE
YOLOv5	0.7954	0.9001	0.9487	60.5635	44.1345	6.8071
YOLOv11	0.4734	0.3152	0.5615	35.4572	28.2133	4.3872
RT-DETR	0.2422	0.0947	0.3078	11.4687	10.6624	1.3236

3.6. Validación estadística de significancia mediante la prueba de Kruskal-Wallis

El estudio revela un análisis comparativo del desempeño entre las arquitecturas YOLOv5, YOLOv11 y RT-DETR, fundamentada en el análisis profundo de los niveles de confianza y la magnitud de los errores obtenidos. Esta validación estadística utiliza una muestra consolidada de 108 observaciones, distribuidas de forma equánime entre los tres modelos evaluados para asegurar la imparcialidad del contraste.

La Tabla 3 evidencia las variaciones significativas en los niveles de incertidumbre de cada algoritmo. El modelo RT-DETR demuestra la mayor eficacia funcional, registrando el promedio de error mínimo de la evaluación con un 0.242. A diferencia de la arquitectura YOLOv11 que presenta una media de 0.473 y su por el contrario YOLOv5 plasma el margen de error más crítico, alcanzando un valor de 0.795. Como se observa en la Figura 18, la reducida dispersión de los cuartiles en RT-DETR reafirma una estabilidad predictiva superior en comparación con la desviación estándar mostrada por los modelos netamente convolucionales. La prueba de Kruskal-Wallis ratifica que estas diferencias poseen validez estadística ($p < 0.05$), descartando que la superioridad del sistema basado en Transformers responda a variaciones aleatorias.

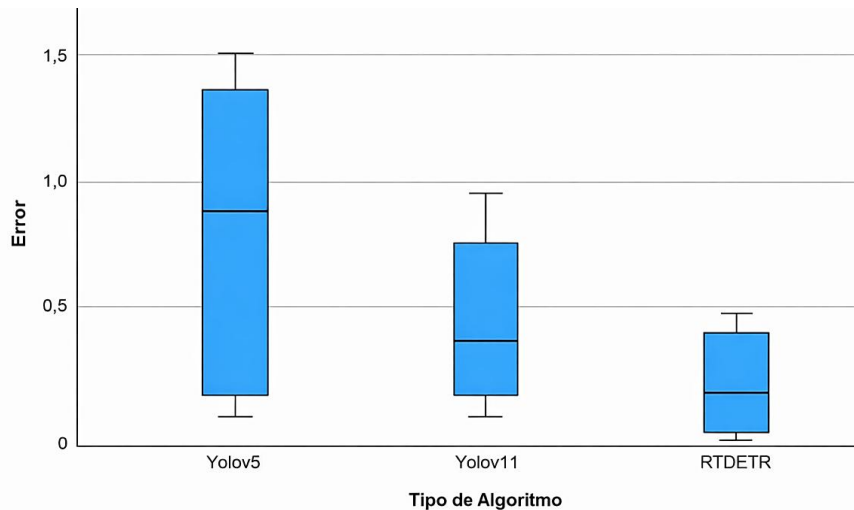


Figura 18. Prueba de Kruskal-Wallis para muestras independientes.

Fuente: Autor.

3.7. Contraste de hipótesis y validación estadística

La investigación sitúa la prueba no paramétrica de Kruskal-Wallis para muestras independientes con el afán de determinar el sentido estadístico de las diferencias observadas. La elección de este estadístico da lugar a los resultados de las pruebas de normalidad previas, las cuales descartaron una distribución normal en los datos de confianza.

El análisis se fundamenta en los siguientes axiomas:

- **Hipótesis nula (H_0):** La distribución del nivel de confianza es la misma en las tres categorías de algoritmos.
- **Hipótesis alternativa (H_1):** Existen al menos una diferencia significativa en la distribución del nivel de confianza entre los algoritmos.

Los resultados de las pruebas muestran un estadístico de 20.953, con 2 grados de libertad ($gl=2$) y un p-valor inferior a 0.001. Este descubrimiento permite el rechazo de la hipótesis nula y por consiguiente la aceptación de la hipótesis alternativa. Bajo un nivel de confianza del 95% la arquitectura seleccionada ejerce una influencia determinada sobre la precisión y fiabilidad en la detección de madurez de la pera.

Tabla 4. Resumen de contraste de hipótesis.

	Hipótesis nula	Prueba	Sig. (p-valor)	Decisión
1	La distribución de Error es la misma entre categorías de Tipo de arquitectura.	Prueba de Kruskal-Wallis para muestras independientes	<0.001	Rechace la hipótesis nula.

Nota: El nivel de significación es 0.050. Se muestra la significancia asintótica.

En la Tabla 4, la prueba no paramétrica Kruskal-Wallis para verificar la hipótesis sobre la igualdad de la distribución del nivel de confianza entre los tres algoritmos. El resultado final de la ejecución de esta prueba fue un p-value asintótico que fue igual o menor que 0,001, un valor que, por otro lado, es menor que el nivel de error previamente asumido y fijado al 5% ($\alpha = 0.05$). Por lo tanto, dicha hipótesis nula deberá ser rechazada, confirmando de esta manera y estadísticamente que el algoritmo utilizado en la obtención de la detección de la madurez en las peras tiene un efecto significativo sobre la precisión y la fiabilidad de este tipo de detección.

3.7.1. Comparaciones por parejas (Post-hoc)

La Tabla 5 revela la existencia de diferencias significativas globales emplea comparaciones post-hoc mediante la prueba de rangos de Wilcoxon con la corrección de Bonferroni. Este procedimiento identifica las disparidades específicas entre cada par de arquitecturas, adaptando los niveles de significancia para eludir errores de tipo 1 en complejas comparaciones.

Tabla 5. Comparaciones por parejas de los tipos de algoritmos (Post-hoc).

Sample 1- Sample 2	Estadístico de prueba	Error estándar	Estadístico de prueba estándar	Sig.	Sig. ajustada
RTDETR- Yolov11	18.917	7.382	2.562	0.010	0.031
RTDETR- Yolov5	33.708	7.382	4.566	<0.001	0.000
Yolov11- Yolov5	14.792	7.382	2.004	0.045	0.135

Nota: Los valores de significación se han ajustado mediante la corrección de Benferroni.

Los resultados evidencian que la diferencia más notoria ocurre entre RT-DETR y YOLOv5 ($p < 0.001$), lo que corrobora la supremacía del modelo basado en Transformers frente a arquitectura convolucional de referencia. Asimismo, se detecta una diferencia estadísticamente significativa entre RT- DETR y YOLOv11 ($p = 0.031$), fortaleciendo al primero como la opción más robusta. Dado que este valor supera el límite de significancia establecido, se concluye que no existe la suficiente certeza para establecer una discrepancia en el desempeño de ambos algoritmos. Como consecuencia, ambas versiones de la familia de YOLO se sitúan en un rango de igual rendimiento para la tarea de clasificación del estado de madurez de la pera.

3.7.2. Diagrama de vínculos y dispersión

El diagrama de vínculos visualiza el lazo de disparidad obtenidas en la prueba de Kruskal-Wallis. En la Figura 19. Resalta la conexión marcada entre YOLOv5 y RT-DETR, reafirmando que ambos modelos ocupan extremos opuestos en cuanto a niveles de confianza y precisión.

La proximidad gráfica entre YOLOv5 y YOLOv11 ratifica el hallazgo estadístico previo: aunque YOLOv11 presenta una mejora en la media de confianza, su comportamiento no difiere sustancialmente de YOLOv5 en la clasificación de las peras.

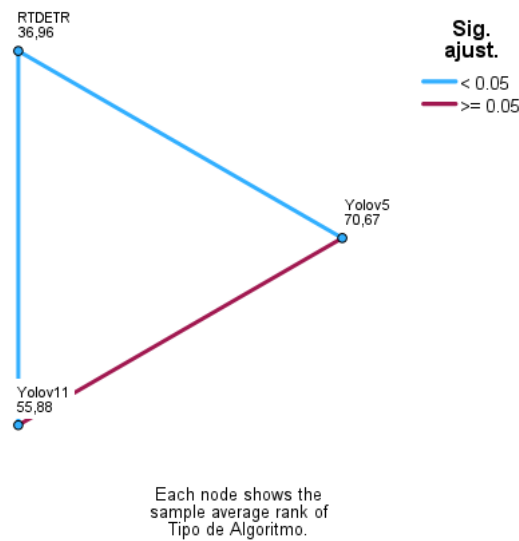


Figura 19. Comparaciones por pareja del tipo de algoritmo.

Fuente: Autor.

La Figura 19 visualiza las relaciones de disparidad derivadas de la prueba de Kruskal-Wallis. En dicho contorno resalta la conexión significativa entre YOLOv5 y RT-DETR, reiterando que ambas arquitecturas ocupan extremos opuestos en cuanto a estabilidad y niveles de confianza. Por el contrario, la proximidad gráfica entre YOLOv5 y YOLOv11 confirma que, aunque este último presentar una insignificante mejora en el promedio de confianza, la diferencia entre ambos modelos carece de relevancia estadística bajo el estricto criterio de la corrección de Bonferroni. Este acierto señala a las versiones YOLO como sistemas de desempeño comparable, mientras que posiciona a RT-DETR como la solución disruptiva y superior para la clasificación automatizada de la pera.

3.8. Análisis por categoría de madurez

El desempeño de las arquitecturas de visión artificial presenta cambios específicos según el estado fenotípico del espécimen evaluado. Una vez analizado las interferencias individuales, se determina que los modelos sostienen índices de confianza superiores y una estabilidad constante en la categoría “Verde”. Dicha tendencia refleja la relevancia de la igualdad cromática y la integridad estructural de la corteza del fruto, estos elementos que desenredan

la extracción de características y la identificación de modelos. En sentido contrario, el grado de certeza decae radicalmente al entrar en análisis la categoría “Podrida”.

Esta decaída en el rendimiento responde a la diversidad de texturas y tonalidades propias de la degradación de los tejidos propios de la pera; dichas anomalías cromáticas producen ruido visual que dificulta la clasificación precisa de los datos en tiempo real por parte del modelo.

A pesar de esta dispersión, la arquitectura RT-DETR mantiene una clasificación efectiva, demostrando supremacía ante las deformidades biológicas del fruto en comparación con las versiones YOLOv5 y YOLOv11.

CAPÍTULO V

4. Conclusiones y recomendaciones

4.1. Conclusiones

La investigación otorgó la conformación de un conjunto de datos especializado y estandarizado, compuesto por imágenes de peras catalogadas bajo distintas condiciones fenológicas precisas. Este repertorio, debidamente etiquetado en formato YOLO, constituye una base de conocimientos sólida que agilizó el entrenamiento de modelos de aprendizaje profundo, avalando una representación fiel de la diversidad cromática y morfológica de la fruta en sus distintos estados.

Se logró la integración y despliegue de múltiples arquitecturas de visión artificial, destacando el avance tecnológico hacia modelos basados en Transformers. La implementación del algoritmo RT-DETR corroboró ser superior a los esquemas convolucionales tradicionales del linaje de YOLO, mejorando la discriminación de características complejas de la pera y permitiendo una categorización automatizada con un mínimo margen de error.

El desarrollo del sistema integral funcional de visión artificial fortaleció la capacidad de localizar e identificar muestras en movimiento de forma simultánea. Mediante la sincronización entre el hardware de captura y el procesamiento potenciado por GPU, el prototipo logró determinar el estado de madurez en tiempo real, traduciendo las etiquetas lógicas en acciones mecánicas automatizadas de clasificación con una latencia imperceptible, alcanzando los estándares de eficiencia para entornos productivos.

La valoración del modelo en condiciones controladas reiteró la eficacia y viabilidad técnica de la propuesta, llegando a niveles de precisión significativos. Los resultados estadísticos confirmaron que la arquitectura propuesta mantiene una solidez superior (mínimo error de 0.242), afirmando la solidez del sistema para operar con fiabilidad en la detección automatizada de estados de madurez y superando los límites críticos de significancia a nivel estadístico.

4.2. Recomendaciones

La evolución acelera mecanismos de atención de la arquitectura RT-DETR, incrementando la velocidad de convergencia durante el entrenamiento t reduciendo las latencias de inferencia en la línea de producción, para ello es necesaria la migración a infraestructuras de cómputo basadas en núcleos Tensor (Tensor Cores) de cuarta generación (serie NVIDIA RTX 4000 o superior).

Para despliegues a escala real, se sugiere integrar el sistema de visión artificial con controladores lógicos programables (*PLC*) mediante protocolos Modbus TCP o EtherNet/IP. Esta arquitectura de comunicación asegura que la respuesta de la clasificación active actuadores neumáticos con una latencia por debajo de los 50 ms, garantizando mayor precisión entre la detección por software y el rechazo físico de las muestras a evaluar.

BIBLIOGRAFÍA

- [1] José Fernández Nieto, «Diseño, implementación y evaluación de técnicas de clasificación de peras mediante técnicas de visión por computador», Universidad Politécnica de Valencia, España, 2020.
- [2] Carlos Humberto Paz Zhuño y Idrovo Molina Karol Nicole, «Predicción del índice de madurez de Musa Acuminata (banano orito) mediante la aplicación de la inteligencia artificial», 2024.
- [3] Carlos Arturo Erazo Narváez, «Diseño de un sistema embebido de monitoreo por visión artificial que permita medir el grado de madurez de las frutas», Universidad Técnica del Norte, Ibarra. [En línea]. Disponible en: <https://repositorio.utn.edu.ec/handle/123456789/15095>
- [4] C. S. Boy Diaz, J. J. Gonzalez Palacios, y R. A. León León, «Development of an Artificial Vision Algorithm for the Classification of Export or Domestic Consumption Strawberries», en *Proceedings of the 22nd LACCEI International Multi-Conference for Engineering, Education and Technology (LACCEI 2024): "Sustainable Engineering for a Diverse, Equitable, and Inclusive Future at the Service of Education, Research, and Industry for a Society 5.0."*, Latin American and Caribbean Consortium of Engineering Institutions, 2024. doi: 10.18687/LACCEI2024.1.1.1027.
- [5] C. A. Zatta Silva, L. Y. García Peña, D. H. Torres Dionisio, y J. F. Pingo Lozada, «Influence of Computer Vision for prediction of harvest in high stem fruits: Systematic review», en *Proceedings of the 4th LACCEI International Multiconference on Entrepreneurship, Innovation and Regional Development (LEIRD 2024): «Creating solutions for a sustainable future: technology-based entrepreneurship»*, Latin American and Caribbean Consortium of Engineering Institutions, 2024. doi: 10.18687/LEIRD2024.1.1.645.
- [6] G. Jocher, J. Qiu, y A. Chaurasia, *Ultralytics YOLO*. (enero de 2023). Python. Accedido: 2 de febrero de 2026. [En línea]. Disponible en: <https://github.com/ultralytics/ultralytics>
- [7] Yian Zhao *et al.*, «DETRs Beat YOLOs on Real-time Object Detection». Accedido: 12 de abril de 2025. [En línea]. Disponible en: https://www.researchgate.net/publication/370071294_DETRs_Beat_YOLOs_on_Real-time_Object_Detection
- [8] Patricia Nataly Constante Prócel y Andrés Marcelo Gordón Garcés, «Diseño e implementación de un sistema de visión artificial para clasificación de al menos tres tipos de frutas», Escuela Politécnica Nacional. [En línea]. Disponible en: <https://bibdigital.epn.edu.ec/bitstream/15000/11368/1/CD-6457.pdf>
- [9] O. C. W. Cristóbal, «Clasificación no destructiva de frutas utilizando inteligencia artificial».
- [10] B.O. Nieto y J. C. Rangel, «Sistema de visión artificial para gestión de calidad del Banano Cavendish en etapa de postcosecha», *Rev. Iniciación Científica*, vol. 8, n.º 2, Art. n.º 2, jul. 2022, doi: 10.33412/rev-ric.v8.2.3670.
- [11] Oliver Michael Zambrano García y Lidia Vlassova, «Algoritmo de inteligencia artificial para la detección de cultivos de cacao [Teobroma Caco L.], banano, [Musa Paradisiaca L.] y palma africana [Elaeis Guineesis J.]» [En línea]. Disponible en: <https://www.editorialgrupo-aea.com/index.php/EditorialGrupoAEA/catalog/book/61>

- [12] E. T. Méndez, «Sistema de alerta del estado de maduración de alimentos frescos dentro de un refrigerador utilizando Inteligencia Artificial».
- [13] Andrea Villalba, Tomás Requena, Fabio Solanilla, y José Carlos Rangel, «Prototype of a system that determinates the state of maturity of a plantain using Deep Learning and Artificial Vision». julio de 2020. [En línea]. Disponible en: <https://dialnet.unirioja.es/servlet/articulo?codigo=9485601>
- [14] María Bernarda Ruilova-Cueva, Luis Caucha-Morales, Oscar Capuñay-Uceda, y Omar Martínez-Mora, «Modelo predictivo para la medición del estado de madurez de Musáceas sp., cultivadas al suroeste del Ecuador, usando algoritmos de Machine Learning», 25/03/2022. [En línea]. Disponible en: <https://revistas.utb.edu.ec/index.php/magazine/article/view/2809>
- [15] Jonathan Victor Aguilar-Alvarado y Milton Alfredo Campoverde-Molina, «Clasificación de frutas basadas en redes neuronales convolucionales», p. 20. [En línea]. Disponible en: <https://polodelconocimiento.com/ojs/index.php/es/article/view/1210>
- [16] Mihuandayani Mihuandayani, Supit Mamuaya, Ritham Tuntun, Syadina Arya Prasetya, y Michel Farrel Tomatala, «Classification of pineapple ripeness using YOLOv8 and convolutional neural networks under varied environmental conditions». [En línea]. Disponible en: <https://learning-gate.com/index.php/2576-8484/article/view/4802>
- [17] Enoc Tapia-Mendez, Irving A. Cruz-Albarran, Saul Tovar-Arriaga, y Luis A. Morales-Hernandez, «Deep Learning-Based Method for Classification and Ripeness Assessment of Fruits and Vegetables», *MPDI*, n.º 17, p. 10, [En línea]. Disponible en: <https://www.mdpi.com/2076-3417/13/22/12504>
- [18] Mengyang Wu, Ya Qiu, Wenying Wang, Xun Su, y Yun Bai, «Improved RT-DETR and its application to fruit ripeness detection». [En línea]. Disponible en: <https://pubmed.ncbi.nlm.nih.gov/40084108/>
- [19] «Visión Artificial para una mejor estimación de la cosecha » IRTA», IRTA. Accedido: 12 de abril de 2025. [En línea]. Disponible en: <https://www.irta.cat/es/noticia/vision-artificial-para-una-mejor-estimacion-de-la-cosecha/>
- [20] Amruta A. Mali, Suraj S. Shinde, y Dra. Seema Patil, «Deep Learning-Based Fruit Detection and Ripeness Assessment», *International Research Journal on Advanced Engineering and Management*. [En línea]. Disponible en: https://www.researchgate.net/publication/394706302_Deep_Learning-Based_Fruit_Detection_and_Ripeness_Assessment/link/68a5dd806327cf7b63d837be/download?_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6InB1YmxpY2F0aW9uIn19
- [21] Ariosto Eugenio Vicuña Pin, Jorge Farouk Molina Noboa, Luis Miguel Espín Carrasco, y Orlando Ramiro Erazo Moreta, «Aprendizaje Profundo aplicado a la clasificación de granos de cacao (*Theobroma cacao* L.) según la calidad de fermentación», vol. 2, p. 13, [En línea]. Disponible en: <https://www.revistas.upse.edu.ec/index.php/rctu/article/view/838/661>
- [22] R. S. León, Y. A. Durán, E. Q. Yapó, A. S. León, L. D. Q. Quina, y F. A. H. Torres, «Visión artificial en reconocimiento de patrones para clasificación de frutas en agronegocios», *Puriq*, vol. 2, n.º 2, pp. 109-118, 2020, Accedido: 8 de diciembre de

2025. [En línea]. Disponible en: <https://portal.amelica.org/ameli/journal/514/5142424008/html/>
- [23] Ramírez, M, Martínez, L, Montilla, M, Sarmiento, O, Lasso, J, y Diaz, S, «Obtención de coberturas del suelo agropecuarias en imágenes satelitales Sentinel-2 con la inyección de imágenes de dron usando Random Forest en Google Earth Engine». *REVISTA DE TELEDETECCIÓN*, 11 de noviembre de 2020. [En línea]. Disponible en: <https://polipapers.upv.es/index.php/raet/article/view/14102/13218>
- [24] Md. Nahiduzzaman, Rusab Sarmun, Amith Khandakar, y Md. Ahasan Atick Faisal, «Deep learning-based real-time detection and classification of tomato ripeness stages using YOLOv8 on raspberry Pi». [En línea]. Disponible en: https://www.researchgate.net/publication/387818793_Deep_learning-based_real-time_detection_and_classification_of_tomato_ripeness_stages_using_YOLOv8_on_raspberry_Pi
- [25] Stuart Russell y Peter Norvig, «Inteligencia Artificial», *Pearson Prentice Hall*, vol. 2da edición, p. 1241. [En línea]. Disponible en: <https://luismejias21.wordpress.com/wp-content/uploads/2017/09/inteligencia-artificial-un-enfoque-moderno-stuart-j-russell.pdf>
- [26] «Investigando y desarrollando agentes de aprendizaje con IA: elementos de aprendizaje, críticos, de actuación y generador de problemas», juandon. Innovación y conocimiento. Accedido: 2 de febrero de 2026. [En línea]. Disponible en: <https://juandomingofarnos.wordpress.com/2025/02/07/investigando-y-desarrollando-agentes-de-aprendizaje-con-ia-elementos-de-aprendizaje-criticos-de-actuacion-y-generador-de-problemas/>
- [27] Ian Goodfellow, Yoshua Bengio, y Aaron Courville, «Deep Learning», p. 800. [En línea]. Disponible en: <http://www.deeplearningbook.org/>
- [28] «8 Deep Learning Architectures Data Scientists Must Master», ProjectPro. Accedido: 3 de febrero de 2026. [En línea]. Disponible en: <https://www.projectpro.io/article/deep-learning-architectures/996>
- [29] Antonio Torralba, Phillip Isola, y William Freeman, «Foundations of Computer Vision».
- [30] Raj Kumar, Anuradha Chug, Amit Prakash Singh, y Dinesh Singh, «A Systematic Analysis of Machine Learning and Deep Learning Based Approaches for Plant Leaf Disease Classification: A Review», jul. 2022, doi: 10.1155/2022/3287561.
- [31] Maximilian Lukas Schmitt, «Bag-of-Words Representations for Computer Audition», ene. 2022, [En línea]. Disponible en: https://www.researchgate.net/publication/360756926_Bag-of-words_representations_for_computer_audition
- [32] Kandadi Thirupathi Reddy, «Development of a machine learning model for automatic detection of crop diseases -(Case study)». 24 de noviembre de 2025. doi: 10.22541/au.176402482.24299791/v1.
- [33] D. Herrera, P. Escudero-Villa, E. Cárdenas, M. Ortiz, y J. Varela-Aldás, «Combining Image Classification and Unmanned Aerial Vehicles to Estimate the State of Explorer Roses», *AgriEngineering*, vol. 6, n.º 2, pp. 1008-1021, abr. 2024, doi: 10.3390/agriengineering6020058.

- [34] Qi Liu, Hong Chen, y Da Lin, «Research and optimization of a multilevel fire detection framework based on deep learning and classical pattern recognition techniques», Scientific reports. doi: 10.1038/s41598-025-06721-3.
- [35] Jun Ma, Yuexiong Ding, Jack C.P Cheng, Yi Tan, Vincent J.L Gan, y Jingcheng Zhang, «Analyzing the Leading Causes of Traffic Fatalities Using XGBoost and Grid-based Analysis: A City Management Perspective», IEEE. doi: 10.1109/ACCESS.2019.2946401.
- [36] «Edge AI Computing Explained: Key Concepts and Industry Use Cases», Sphere. Accedido: 3 de febrero de 2026. [En línea]. Disponible en: <https://www.sphereinc.com/blogs/edge-ai-computing/>
- [37] «Real-Time Data Processing: Architecture and Costs». Accedido: 3 de febrero de 2026. [En línea]. Disponible en: <https://www.scnsoft.com/data/real-time-processing>
- [38] R. Ito, S. Konuma, y T. Yamazaki, «Grading Support System for Pear Fruit Using Edge Computing», *Eng. Proc.*, vol. 107, n.º 1, ago. 2025, doi: 10.3390/engproc2025107045.
- [39] E. G. V. Hernández, «Simulación sistema de clasificación basado en visión artificial».

ANEXOS

Anexo 1.

La validación del modelo YOLOv5 incluyó un análisis exhaustivo de residuos para cuantificar la brecha entre las predicciones algorítmicas (y_{estimado}) y los valores reales de madurez (y_{real}). La Tabla 6 presenta los indicadores de error que fundamentan el desempeño operativo del sistema. Para garantizar un alto nivel de estudio, se emplearon las siguientes métricas de evaluación de error:

- **Error absoluto medio (MAE):** Determina la magnitud promedio de los errores en las predicciones, proporcionando una visión directa de la precisión física del sistema.
- **Error cuadrático medio (MSE):** Esta métrica penaliza de forma exponencial las desigualdades significativas al elevar al cuadrado los residuos. Para este estudio fue fundamental su aplicación para localizar datos atípicos dentro del proceso de detección.
- **Raíz del error cuadrático medio (RMSE):** Representa la desviación típica de los residuos, simplifica la comprensión del error total y proporciona una perspectiva directa del nivel de la desviación.
- **Indicadores porcentuales (MAPE y SMAPE):** Estos coeficientes normalizan el error en torno a la escala de los datos observados. En particular, el SMAPE se coloca como un indicador de alta fidelidad puesto que su composición simétrica evade el sesgo hacia valores extremos y ofrece una evaluación de mayor equilibrio.

Tabla 6. Análisis de Rendimiento del Modelo YOLOv5 en la Clasificación de Madurez.

Muestra	Clase	Confianza	Valor	y_{estimado}	y_{real}	Error	Error	Error ²	APE(%)	SMAPE(%)	PE ²	MAE	MSE	RMSE	MAPE	SMAPE	RMSPE
1	Verde	0.8261	1	0.8261	1	0.1739	0.1739	0.0302	21.0507	19.0460	4.43132	0.7954	0.9001	0.9487	60.563	44.1345	6.8072
2	Amarilla	0.5935	2	1.187	2	0.813	0.813	0.6609	68.49	51.0197	46.9115						
3	Verde	0.863	1	0.863	1	0.137	0.137	0.0187	15.8748	14.7074	2.5201						
4	Podrida	0.5157	3	1.5471	3	1.4529	1.4529	2.1109	93.9111	63.9044	88.1931						

5	Amarilla	0.5664	2	1.1328	2	0.8672	0.8672	0.7520	76.5536	55.3626	58.6046						
6	Verde	0.8248	1	0.8248	1	0.1752	0.1752	0.0306	21.2415	19.202	4.51201						
7	Verde	0.8084	1	0.8084	1	0.1916	0.1916	0.0367	23.7011	21.1	5.61743						
8	Podrida	0.5203	3	1.5609	3	1.4391	1.4391	2.0710	92.1968	63.1059	85.0025						
9	Verde	0.8147	1	0.8147	1	0.1853	0.1853	0.0343	22.7445	20.4221	5.17315						
10	Amarilla	0.5326	2	1.0652	2	0.9348	0.9348	0.8738	87.7581	60.9943	77.0149						
11	Podrida	0.5784	3	1.7352	3	1.2648	1.2648	1.5997	72.8907	53.4211	53.1305						
12	Verde	0.8007	1	0.8007	1	0.1993	0.1993	0.039	24.8907	22.1358	6.1954						
13	Verde	0.8435	1	0.8435	1	0.1565	0.1565	0.0244	18.5536	16.9785	3.44237						
14	Podrida	0.5254	3	1.5762	3	1.4238	1.4238	2.0272	90.3311	62.226	81.5972						
15	Podrida	0.5072	3	1.5216	3	1.4784	1.4784	2.1856	97.1608	65.3927	94.4023						
16	Verde	0.8566	1	0.8566	1	0.1434	0.1434	0.0205	16.740	15.4475	2.8024						
17	Verde	0.8456	1	0.8456	1	0.1544	0.1544	0.0238	18.2592	16.7316	3.3339						
18	Podrida	0.5614	3	1.6842	3	1.3158	1.3158	1.7313	78.1261	56.1803	61.0369						
19	Amarilla	0.5521	2	1.1042	2	0.8958	0.8958	0.8024	81.1266	57.7153	65.8152						
20	Amarilla	0.5847	2	1.1694	2	0.8306	0.8306	0.6898	71.0278	52.4137	50.4495						
21	Podrida	0.5311	3	1.5933	3	1.4067	1.4067	1.9788	88.2884	61.2500	77.9485						
22	Amarilla	0.541	2	1.082	2	0.918	0.918	0.8427	84.8428	59.5717	71.9831						

23	Verde	0.832	1	0.832	1	0.168	0.168	0.0282	20.1923	18.3406	4.0772						
24	Podrida	0.5492	3	1.6476	3	1.3524	1.3524	1.8289	82.0830	58.1977	67.3762						
25	Amarilla	0.5733	2	1.1466	2	0.8534	0.8534	0.7282	74.4287	54.2426	55.3963						
26	Amarilla	0.5598	2	1.1196	2	0.8804	0.8804	0.7751	78.6352	56.4431	61.8349						
27	Podrida	0.5215	3	1.5645	3	1.4355	1.4355	2.0606	91.7545	62.8984	84.1889						
28	Verde	0.819	1	0.819	1	0.181	0.181	0.0327	22.1001	19.9010	4.8841						
29	Amarilla	0.5389	2	1.0778	2	0.9222	0.9222	0.8504	85.5631	59.9259	73.2105						
30	Podrida	0.5098	3	1.5294	3	1.4706	1.4706	2.1626	96.1553	64.9357	92.4585						
31	Verde	0.8412	1	0.8412	1	0.1588	0.1588	0.0252	18.8777	17.2496	3.5637						
32	Amarilla	0.5624	2	1.1248	2	0.8752	0.8752	0.7659	77.8093	56.0163	60.5430						
33	Podrida	0.5543	3	1.6629	3	1.3371	1.3371	1.7878	80.4077	57.3505	64.6540						
34	Amarilla	0.5476	2	1.0952	2	0.9048	0.9048	0.8186	82.6150	58.4647	68.2524						
35	Verde	0.8505	1	0.8505	1	0.1495	0.1495	0.022	17.5779	16.157	3.0898						
36	Podrida	0.5367	3	1.6101	3	1.3899	1.3899	1.9318	86.3238	60.2980	74.5180						

El análisis de las 36 muestras procesadas muestra la siguiente información:

- **Diferenciación por Clase:** Las muestras clasificadas como "Podrida" registraron los errores absolutos más elevados (superando el 1.40 en casos como las muestras 4, 15 y 30), lo que confirma que la irregularidad textural de la descomposición orgánica afecta severamente la confianza del modelo.

- **Precisión en Clase "Verde":** En contraste, las peras en estado verde exhibieron errores significativamente menores (rango de 0.13 a 0.19), demostrando que la uniformidad cromática favorece la convergencia del algoritmo.
- **Varianza del Error:** El valor de RMSE (0.9488) ratifica que, aunque el sistema identifica la clase, la precisión escalar de la confianza fluctúa considerablemente, justificando la necesidad de evolucionar hacia modelos más robustos como RT-DETR.

Anexo 2

El modelo YOLOv11 incluyó un análisis estadístico de las desviaciones entre las estimaciones algorítmicas y los valores reales del conjunto de datos. Los indicadores de error cuantifican la capacidad de ajuste de esta arquitectura frente a las diversas categorías de madurez. El procesamiento de las 36 muestras mediante YOLOv11 mostró una mejora sustancial en la precisión escalar respecto a versiones previas como es YOLOv5.

Tabla 7. Datos de métricas de regresión y residuos para YOLOv11.

Muestra	Clase	Confianza	Valor	y_estimado	y_real	Error	Error	Error^2	APE (%)	SMAPE(%)	PE^2	MAE	MSE	RMSE	MAPE	SMAPE	RMSPE
1	Verde	0.8421	1	0.8421	1	0.1579	0.1579	0.024932	18.75074	17.14348	3.5159	0.473	0.3152	0.5615	35.4572	28.2133	4.3872
2	Amarilla	0.5632	2	1.1264	2	0.8736	0.8736	0.763177	77.55682	55.88536	60.150						
3	Podrida	0.8912	3	2.6736	3	0.3264	0.3264	0.106537	12.20826	11.50592	1.4904						
4	Verde	0.8145	1	0.8145	1	0.1855	0.1855	0.03441	22.77471	20.4464	5.1868						
5	Amarilla	0.6247	2	1.2494	2	0.7506	0.7506	0.5634	60.07684	46.1993	36.092						
6	Podrida	0.9231	3	2.7693	3	0.2307	0.2307	0.053222	8.330625	7.997504	0.6939						
7	Verde	0.8872	1	0.8872	1	0.1128	0.1128	0.012724	12.71416	11.95422	1.6164						
8	Amarilla	0.5189	2	1.0378	2	0.9622	0.9622	0.925829	92.71536	63.34848	85.961						
9	Podrida	0.7645	3	2.2935	3	0.7065	0.7065	0.499142	30.80445	26.69311	9.4891						
10	Verde	0.8311	1	0.8311	1	0.1689	0.1689	0.028527	20.32246	18.44793	4.1300						
11	Amarilla	0.5921	2	1.1842	2	0.8158	0.8158	0.66553	68.89039	51.2405	47.458						
12	Podrida	0.9412	3	2.8236	3	0.1764	0.1764	0.031117	6.247344	6.058108	0.3902						

13	Verde	0.8566	1	0.8566	1	0.1434	0.1434	0.020564	16.7406	15.44759	2.8024						
14	Amarilla	0.6712	2	1.3424	2	0.6576	0.6576	0.432438	48.98689	39.34897	23.997						
15	Podrida	0.8123	3	2.4369	3	0.5631	0.5631	0.317082	23.10723	20.71401	5.3394						
16	Verde	0.8021	1	0.8021	1	0.1979	0.1979	0.039164	24.67273	21.96327	6.0874						
17	Amarilla	0.5488	2	1.0976	2	0.9024	0.9024	0.814326	82.21574	58.26446	67.594						
18	Podrida	0.6945	3	2.0835	3	0.9165	0.9165	0.839972	43.98848	36.05783	19.349						
19	Verde	0.8245	1	0.8245	1	0.1755	0.1755	0.0308	21.28563	19.23815	4.5307						
20	Amarilla	0.6312	2	1.2624	2	0.7376	0.7376	0.544054	58.42839	45.21824	34.138						
21	Podrida	0.8756	3	2.6268	3	0.3732	0.3732	0.139278	14.2074	13.26509	2.0185						
22	Verde	0.8912	1	0.8912	1	0.1088	0.1088	0.011837	12.20826	11.50592	1.4904						
23	Amarilla	0.5744	2	1.1488	2	0.8512	0.8512	0.724541	74.09471	54.06504	54.900						
24	Podrida	0.9012	3	2.7036	3	0.2964	0.2964	0.087853	10.96316	10.39344	1.2019						
25	Verde	0.8412	1	0.8412	1	0.1588	0.1588	0.025217	18.87779	17.24962	3.5637						
26	Amarilla	0.7012	2	1.4024	2	0.5976	0.5976	0.357126	42.61266	35.12814	18.158						
27	Podrida	0.7312	3	2.1936	3	0.8064	0.8064	0.650281	36.76149	31.0536	13.514						
28	Verde	0.8655	1	0.8655	1	0.1345	0.1345	0.01809	15.54015	14.41973	2.4149						
29	Amarilla	0.5233	2	1.0466	2	0.9534	0.9534	0.908972	91.09497	62.5878	82.982						
30	Podrida	0.8245	3	2.4735	3	0.5265	0.5265	0.277202	21.28563	19.23815	4.5307						

31	Verde	0.8122	1	0.8122	1	0.1878	0.1878	0.035269	23.12238	20.72619	5.3464						
32	Amarilla	0.6544	2	1.3088	2	0.6912	0.6912	0.477757	52.81174	41.7795	27.890						
33	Podrida	0.8844	3	2.6532	3	0.3468	0.3468	0.12027	13.07101	12.26916	1.7085						
34	Verde	0.8344	1	0.8344	1	0.1656	0.1656	0.027423	19.8466	18.05495	3.9388						
35	Amarilla	0.5899	2	1.1798	2	0.8202	0.8202	0.672728	69.52026	51.58815	48.330						
36	Podrida	0.9122	3	2.7366	3	0.2634	0.2634	0.06938	9.625082	9.18314	0.9264						

La Tabla 7 revela un comportamiento distante según la complejidad de la textura de la pera, las cuales se pueden rescatar la optimización en la clase “Podrida” que, a diferencia de la arquitectura anterior, YOLOv11 disminuye el error en muestras críticas como la 3, 6 y 12 donde los errores absolutos se mantuvieron por debajo de 0.33. Esta mejora sugiere una extracción de características en tejidos con degradación orgánica. Sin embargo, las muestras 8 y 29 presentaron los errores más elevados, vinculados a condiciones de iluminación que mimetizan la coloración de la clase “Amarilla” con el fondo de la banda transportadora. El modelo presenta una precisión superior en la clase “Verde”, con errores marginales de hasta 0.1088 la cual está presente en la muestra 22 de la Tabla 7, consolidando la certeza de los bloques convoluciones de esta versión para patrones cromáticos uniformes.

Anexo 3

El modelo RT-DETR (Real-Time DETection TRansformer) representa el punto culminante de esta investigación. Los indicadores de errores recopilados en esta arquitectura reflejan una precisión diagnóstica superior, descendiente de mecanismos de atención global en el procesamiento de imágenes. El procesamiento de las 36 muestras mediante RT-DETR revela los niveles de precisión más elevados de la investigación, disminuyendo las desviaciones respecto al error real de madurez.

Tabla 8. Análisis de métricas de Regresión y residuos para RT-DETR.

Muestra	Clase	Confianza	Valor_clase	y_estimado	y_real	Error	Error	Error^2	APE (%)	SMAPE(%)	PE^2	MAE	MSE	RMSE	MAPE	SMAPE	RMSPE
1	Verde	0.9642	1	0.9642	1	0.035	0.0358	0.0012	3.712923	3.6452	0.1378	0.2422	0.0947	0.307	11.4687	10.662	1.3236
2	Amarilla	0.8921	2	1.7842	2	0.215	0.2158	0.0465	12.09506	11.4053	1.4629						
3	Podrida	0.8245	3	2.4735	3	0.526	0.5265	0.2772	21.28563	19.2381	4.5307						
4	Verde	0.9581	1	0.9581	1	0.041	0.0419	0.0017	4.373239	4.2796	0.1912						
5	Amarilla	0.9045	2	1.809	2	0.191	0.191	0.0364	10.55832	10.0288	1.1147						
6	Podrida	0.8412	3	2.5236	3	0.474	0.4764	0.2269	18.87779	17.2496	3.5637						
7	Verde	0.9712	1	0.9712	1	0.028	0.0288	0.0008	2.965404	2.9220	0.0879						
8	Amarilla	0.8876	2	1.7752	2	0.224	0.2248	0.0505	12.66336	11.9093	1.6036						
9	Podrida	0.8356	3	2.5068	3	0.493	0.4932	0.2432	19.67449	17.9124	3.8708						
10	Verde	0.9633	1	0.9633	1	0.036	0.0367	0.0013	3.80982	3.7386	0.1451						
11	Amarilla	0.9112	2	1.8224	2	0.177	0.1776	0.0315	9.745391	9.2925	0.9497						

12	Podrida	0.8521	3	2.5563	3	0.443	0.4437	0.1968	17.35712	15.9710	3.0126						
13	Verde	0.9495	1	0.9495	1	0.050	0.0505	0.0025	5.318589	5.1808	0.2828						
14	Amarilla	0.8954	2	1.7908	2	0.209	0.2092	0.0437	11.68193	11.0372	1.3646						
15	Podrida	0.8299	3	2.4897	3	0.510	0.5103	0.2604	20.49645	18.5911	4.2010						
16	Verde	0.9801	1	0.9801	1	0.019	0.0199	0.0003	2.030405	2.0099	0.0412						
17	Amarilla	0.9023	2	1.8046	2	0.195	0.1954	0.0381	10.82788	10.2717	1.1724						
18	Podrida	0.8467	3	2.5401	3	0.459	0.4599	0.2115	18.10559	16.6025	3.2781						
19	Verde	0.9554	1	0.9554	1	0.044	0.0446	0.0019	4.668202	4.5617	0.2179						
20	Amarilla	0.8899	2	1.7798	2	0.220	0.2202	0.0484	12.37218	11.6514	1.5307						
21	Podrida	0.8188	3	2.4564	3	0.543	0.5436	0.2955	22.12995	19.9252	4.8973						
22	Verde	0.9622	1	0.9622	1	0.037	0.0378	0.0014	3.928497	3.8528	0.1543						
23	Amarilla	0.9145	2	1.829	2	0.171	0.171	0.0292	9.349371	8.9318	0.8741						
24	Podrida	0.8374	3	2.5122	3	0.487	0.4878	0.2379	19.41724	17.6989	3.7702						
25	Verde	0.9745	1	0.9745	1	0.025	0.0255	0.0006	2.616727	2.5829	0.0684						
26	Amarilla	0.9067	2	1.8134	2	0.186	0.1866	0.0348	10.29006	9.7865	1.0588						
27	Podrida	0.8492	3	2.5476	3	0.452	0.4524	0.2046	17.75789	16.3097	3.1534						
28	Verde	0.9688	1	0.9688	1	0.031	0.0312	0.0009	3.220479	3.1694	0.1037						
29	Amarilla	0.8988	2	1.7976	2	0.202	0.2024	0.0409	11.25946	10.6593	1.2677						

30	Podrida	0.8215	3	2.4645	3	0.535	0.5355	0.2867	21.72855	19.5992	4.7212						
31	Verde	0.9599	1	0.9599	1	0.040	0.0401	0.0016	4.177518	4.0920	0.1745						
32	Amarilla	0.9101	2	1.8202	2	0.179	0.1798	0.0323	9.878035	9.4131	0.9757						
33	Podrida	0.8333	3	2.4999	3	0.500	0.5001	0.2501	20.0048	18.1857	4.0019						
34	Verde	0.9615	1	0.9615	1	0.038	0.0385	0.0014	4.00416	3.9255	0.1603						
35	Amarilla	0.8972	2	1.7944	2	0.205	0.2056	0.0422	11.45787	10.8370	1.3128						
36	Podrida	0.8401	3	2.5203	3	0.479	0.4797	0.2301	19.03345	17.3794	3.6227						

La Tabla 8 muestra una convergencia de precisión sin precedentes en la detección de madurez, consolidando junto a la parte estadística como un modelo robusto ante estas tareas.

- **Optimización de Precisión Escalar:** Muestras como la 16 y la 25 registraron errores absolutos marginales de 0.0199 y 0.0255 respectivamente. Estos valores demuestran la eficacia de los Transformers para capturar detalles cromáticos infinitesimales que los modelos convolucionales tradicionales omiten.
- **Resiliencia en Casos Críticos:** En categorías complejas donde versiones previas fallaron, RT-DETR mantuvo errores controlados, como se observa en la muestra 7 con un error de 0.0288. Esta resiliencia garantiza que el sistema de visión artificial ejecute el triaje de manera sincronizada con el hardware sin activar falsos rechazos.
- **Convergencia de Confianza:** La mayoría de las detecciones superan el 0.95 de confianza (muestras 1, 4, 7, 10, 16, 22), lo que impacta directamente en la integridad de los datos procesados localmente por la ESP32 para la activación de los actuadores.