Simposio CEA de Robótica, Bioingeniería, Visión Artificial y Automática Marina 2025

Sistema de visión artificial para el control de un robot manipulador agrícola

Fontádez-Parrón, P., Gómez-Espinosa, A.[a]*, Garcia-Chica, A.[a], Torres-Moreno, J. L.[a]

[a] *Department of Engineering, ceiA3, CIESOL, University of Almería, Ctra.Sacramento s/n, La Cañada de San Urbano, Almería, 04120, España, {agespinosa, agc989, jltmoreno}@ual.es*

Resumen

Este trabajo presenta el diseño e implementación de un sistema de visión artificial para la recolección autónoma de tomates en invernaderos, integrando algoritmos de aprendizaje automático y un robot manipulador móvil. Se emplea la cámara Intel RealSense D435 para obtener información tridimensional del entorno, combinada con redes neuronales convolucionales, específicamente el modelo YOLOv8, para detectar y clasificar tomates según su madurez. Se evaluaron las versiones YOLOv8n y YOLOv8s, donde YOLOv8n fue seleccionada por su mejor desempeño en tiempo real, con métricas de precisión aceptables y una inferencia significativamente más rápida. El sistema logró detectar con alta efectividad los frutos, aunque presentó dificultades en la clasificación de tomates semi-maduros y ciertos falsos positivos con fondos similares. La integración con el robot manipulador, controlado mediante MoveIt y ROS, permitió una recolección precisa y autónoma. Se concluye que la propuesta es técnicamente viable, aunque se identifican oportunidades de mejora en la robustez del sistema y la representatividad del conjunto de datos.

Palabras clave: Robótica, agricultura de precisión, automatización, YOLOv8, invernaderos inteligentes.

Machine vision system for controlling an agricultural manipulator robot

Abstract

This work presents the design and implementation of an artificial vision system for the autonomous harvesting of tomatoes in greenhouses, integrating machine learning algorithms and a mobile manipulator robot. The Intel RealSense D435 camera is used to obtain three-dimensional information of the environment, combined with convolutional neural networks, specifically the YOLOv8 model, to detect and classify tomatoes according to their maturity. The YOLOv8n and YOLOv8s versions were evaluated, where YOLOv8n was selected for its better real-time performance, with acceptable accuracy metrics and significantly faster inference. The system was able to detect the fruits with high effectiveness, although it presented difficulties in the classification of semi-ripe tomatoes and certain false positives with similar backgrounds. Integration with the manipulator robot, controlled by MoveIt and ROS, allowed for accurate and autonomous harvesting. It is concluded that the proposal is technically feasible, although opportunities for improvement in the robustness of the system and the representativeness of the dataset are identified.

Keywords: Robotics, Precision agriculture, Automation, YOLOv8, smart greenhouse.

1. Introducción

En los últimos años, la integración de robots en la horticultura de invernadero ha sido una de las principales iniciativas para aumentar la rentabilidad, la producción y la calidad de los cultivos, además de mejorar la sostenibilidad del sector (Sánchez-Molina *et al*., 2024). La implementación de la robótica en diversas labores puede contribuir a disminuir los costos del producto final, al mismo tiempo que optimiza los resultados generales de producción, incrementa la calidad de la fruta y elimina tareas peligrosas como la aplicación de pesticidas (Meshram *et al*., 2022; Brosque and Fischer, 2022) y trabajos monótonos como la recolección (Feng *et al*, 2015; Ji *et al*, 2023).

Desde la década de 1980, muchos investigadores han analizado el uso de robots en tareas de producción de invernaderos en general y han trabajado en su desarrollo. La implementación de dichos robots en este entorno es complicada ya que tienen que trabajar en un sistema más complejo que otros robots como los robots de campo industriales o agrícolas, debido a que se ven influenciados por la naturaleza de los cultivos (color, tamaño, forma variable, textura y ubicación) (Benavides *et al*., 2020) y su entorno

*Autor para correspondencia: agespinosa@ual.es

(iluminación, posición de los frutos, ramas y hojas, corte del suelo, etc.).

Uno de los principales desafíos en la recolección autónoma por visión artificial es garantizar la precisión y la confiabilidad de los sistemas de detección, especialmente en condiciones variables de iluminación, oclusión de cultivos y variabilidad en la apariencia de estos (Mavridou *et al*, 2019). Para abordar estos desafíos, se requiere un desarrollo continuo de algoritmos de visión artificial robustos y adaptativos (Padhiary *et al*., 2024; Tang *et al*., 2020; Suresh-Kumar and Mohan, 2023).

En el caso de este trabajo, se busca implementar un sistema de recolección autónoma por visión artificial, basada en el desarrollo y la implementación de algoritmos de detección y localización de cultivos utilizando técnicas de procesamiento de imágenes y aprendizaje automático. El presente trabajo tiene como objetivo diseñar y desarrollar un algoritmo capaz de realizar la clasificación, selección y localización de productos hortofrutícolas concretamente en el cultivo de tomate. Estos algoritmos permiten a los sistemas de visión artificial identificar la ubicación y la madurez de los cultivos en tiempo real, para posteriormente guiar a los robots o dispositivos de recolección para realizar la cosecha de manera eficiente y precisa.

Este trabajo tiene como finalidad implementar un sistema de recolección autónoma basado en visión artificial, fundamentado en el desarrollo de algoritmos avanzados de detección y localización de frutos en cultivos mediante técnicas de procesamiento de imágenes y aprendizaje automático. El objetivo principal es diseñar y desarrollar un algoritmo capaz de clasificar, seleccionar y localizar productos hortofrutícolas, centrándose específicamente en el cultivo del tomate. Estos algoritmos permiten que los sistemas de visión artificial identifiquen en tiempo real tanto la ubicación como el estado de madurez de los frutos, facilitando así la guía precisa de robots o dispositivos de recolección para llevar a cabo la cosecha de forma eficiente y automatizada.

2. **Materiales y métodos**

2.1. Sistemas de visión artificial

Este artículo se enfoca en el diseño, programación e implementación de un sistema de visión artificial y aprendizaje automático para clasificar, seleccionar y localizar productos hortofrutícolas con el fin de permitir la recolección autónoma en invernaderos. Este sistema se basa en la capacidad de la visión artificial para capturar, procesar y analizar imágenes del mundo real, permitiendo a una máquina interpretar la información visual. La meta es que el sistema pueda identificar la ubicación y madurez de los cultivos en tiempo real, guiando a un robot manipulador para la cosecha eficiente y precisa. La detección de objetos, que fusiona la localización y la clasificación, es el enfoque principal de este estudio.

Para la obtención de información del entorno tridimensional y la percepción de profundidad, un aspecto crucial para la interacción robótica, el sistema emplea la cámara Intel RealSense D435. Esta cámara específica utiliza la tecnología Active IR Stereo. Esta tecnología es un método activo que combina la proyección de un patrón de luz infrarroja estructurada con un sistema de doble cámara infrarroja estéreo. Funciona capturando dos puntos de vista con las cámaras infrarrojas y proyectando un patrón de luz. La profundidad se calcula mediante triangulación al estudiar la deformación del patrón proyectado y comparando las diferencias observadas por cada lente infrarroja. La cámara RealSense D435 puede generar flujos sincronizados de color, profundidad e infrarrojos, y la tecnología Active IR Stereo ofrece ventajas como menor vulnerabilidad a la luz ambiental, mejor rendimiento en baja luz u oscuridad, mayor resolución de profundidad y eficiencia con objetos en movimiento.

El proceso de obtención de las coordenadas tridimensionales del punto a partir de las coordenadas 2D de un píxel (u, v) y su profundidad correspondiente (Z) se realiza mediante principios geométricos de proyección de cámara obteniendo la posición en X e Y mediante la ecuación (1) y (2), respectivamente.

$$X = \left(\frac{u - c_x}{f_x}\right) \cdot Z \tag{1}$$

Donde el cx es el punto principal y fx es la distancia focal en píxeles.

$$Y = \left(\frac{v - c_y}{f_y}\right) \cdot Z \tag{2}$$

Donde el cy es el punto principal y fy es la distancia focal en píxeles.

Para este trabajo la obtención de coordenadas tridimensionales es una etapa considerada e implementada, calculada con las ecuaciones anteriores. La cámara RealSense D435, al ser una cámara calibrada y utilizada con su SDK (como Pyrealsense2), proporciona la profundidad Z y utiliza internamente (o expone para su uso) los parámetros intrínsecos necesarios para aplicar estas transformaciones matemáticas y determinar la ubicación tridimensional (X, Y, Z) de un punto detectado en la imagen.

El procesamiento de las imágenes y la realización de las tareas de detección de objetos se lleva a cabo mediante algoritmos de aprendizaje automático, específicamente utilizando Redes Neuronales Convolucionales (CNNs). Las CNNs son modelos adecuados para trabajar con datos de imagen, compuestas por capas convolucionales, de pooling y densas que procesan la imagen para extraer características y generar resultados (Lubinus-Badillo *et al.*, 2021). Para la tarea de detección de objetos, que implica identificar y localizar múltiples objetos en una imagen, se han desarrollado diversos métodos, divididos en enfoques de dos etapas y métodos de una etapa. El proyecto se basa en YOLO (You Only Look Once), un método de una etapa reconocido por ser rápido y preciso (Redmon *et al*., 2016). YOLO procesa la imagen completa en una sola pasada para predecir simultáneamente las cajas delimitadoras y las clases de los objetos. Se utiliza la versión más reciente, YOLOv8, lanzada en enero de 2023. YOLOv8 es elegido por su versatilidad, precisión, velocidad de procesamiento y facilidad de implementación, y puede realizar diversas tareas, incluida la detección de objetos. Para aplicaciones en tiempo real, se consideran modelos más ligeros de YOLOv8, como YOLOv8n (Ma *et al*., 2024).

2.2. Red neuronal y entrenamiento

En el presente trabajo se optó por la utilización de YOLOv8 como modelo base para la detección de objetos, en lugar de

alternativas más recientes como YOLO-NAS, debido a una combinación de criterios relacionados con la madurez del entorno de desarrollo, la versatilidad funcional, y la eficiencia computacional en entornos de recursos limitados.

YOLOv8, desarrollado por Ultralytics, presenta una arquitectura optimizada para múltiples tareas de visión por computador (detección, segmentación, clasificación y estimación de pose), lo cual aporta flexibilidad en el diseño de sistemas complejos sin necesidad de adoptar modelos independientes para cada subtarea. A diferencia de YOLO-NAS, cuya optimización mediante técnicas de búsqueda neural automatizada (Neural Architecture Search) ofrece ventajas principalmente en precisión sobre benchmarks estáticos, YOLOv8 proporciona una relación más equilibrada entre precisión, velocidad de inferencia y facilidad de implementación, especialmente en escenarios donde el rendimiento en tiempo real y la portabilidad son factores determinantes. Además, YOLOv8 cuenta con una documentación extensa, una comunidad activa de desarrolladores, y un ecosistema robusto que facilita la integración en flujos de trabajo de entrenamiento, validación y exportación a múltiples formatos (ONNX, TensorRT, CoreML, TFLite, entre otros). Estas características resultan particularmente relevantes cuando se busca reducir la complejidad operativa del pipeline de despliegue, como es el caso en aplicaciones de robótica embebida o dispositivos de borde.

Para entrenar los detectores de objetos, se compiló un conjunto de datos de 2363 imágenes de tomates provenientes de fuentes públicas. Se seleccionaron específicamente los datasets Laboro Tomato[1], Tomato Detection Dataset[2] y Hibikino Image Dataset[3]. Este dataset fue sometido a un proceso de preprocesamiento, que incluyó la normalización del tamaño a 640x640 píxeles, y se anotaron manualmente las imágenes con cajas delimitadoras. La clasificación se realizó basándose en tres clases de madurez: Fully-ripe (maduro), Semi-ripe (semi-maduro) y Unripe (verde). Tras la anotación, se cuantificó el número total de instancias etiquetadas, resultando en un total de 24906 anotaciones distribuidas de la siguiente manera: 7986 para Fully-ripe, 5505 para Semi-ripe y 11415 para Unripe. Como se ha señalado, la clase Semi-ripe es la menos representada, constituyendo aproximadamente el 22.10% del total de anotaciones. Esta distribución desigual entre clases, particularmente la menor proporción de tomates semi-maduros, es un factor relevante para la interpretación de las métricas de rendimiento del modelo, ya que puede influir en la capacidad de detección y precisión para esta clase específica. Para el entrenamiento se dividió el dataset en 70% para entrenamiento, 20% validación y 10% prueba, aplicando aumento de datos para robustez. El proceso de entrenamiento se realizó utilizando las arquitecturas ligeras YOLOv8n y YOLOv8s con la librería Ultralytics y Ultralytics Hub, partiendo de modelos preentrenados en COCO. Ambos modelos se entrenaron durante 80 épocas, monitoreando métricas de precisión y recall. La evaluación comparativa se centró en la precisión y, críticamente, la velocidad de inferencia. YOLOv8n fue seleccionado como el modelo final debido a su velocidad de procesamiento significativamente mayor (240ms/fotograma vs 628ms/fotograma para YOLOv8s), optimizando así el rendimiento para la aplicación en tiempo real a pesar de una precisión ligeramente inferior.

2.3. Integración con el robot manipulador

La integración del sistema de visión artificial con el robot manipulador se basa en una arquitectura conjunta. Una vez que el sistema de visión artificial ha identificado y localizado los tomates en el entorno, proporciona esta información posicional al sistema de control del robot. Esta arquitectura permite una coordinación adecuada entre la detección de los tomates y las acciones que el robot debe llevar a cabo para recolectarlos.

De este modo, utilizando la información de localización tridimensional obtenida por el sistema de visión, el robot manipulador puede planificar y ejecutar los movimientos necesarios para alcanzar y recolectar los tomates de manera eficiente.

En cuanto al hardware del sistema robótico, se propone la utilización de un robot manipulador móvil. Este sistema incluye varios componentes físicos esenciales: una base móvil, el manipulador (brazo robótico) en sí, una pinza para la interacción directa con el producto, y el sensor de visión Intel RealSense D435 para la percepción del entorno. Los controladores encargados de gestionar y coordinar los movimientos del robot se plantean como una combinación de RoboRIO y Raspberry Pi. Es crucial que el sistema de control pueda obtener los valores posicionales de cada uno de los motores/actuadores que componen el robot manipulador y mantener comunicación con ellos para el correcto funcionamiento autónomo.

Por el lado del software, la arquitectura de control propuesta consta de varios bloques funcionales clave. Incluye un bloque de Percepción y Localización, que procesa las imágenes del sensor de visión para detectar, clasificar y obtener las coordenadas espaciales de los tomates. Esta información es transmitida a un Coordinador de Tareas (específicamente para la recolección). El coordinador interactúa con el bloque del Robot Manipulador, que a su vez contiene los Algoritmos de Planificación y los Controladores de Articulaciones. Los algoritmos de planificación determinan la trayectoria y los movimientos que el brazo robótico debe seguir para alcanzar el objetivo (el tomate), mientras que los controladores de articulaciones gestionan la acción de los motores/actuadores para ejecutar dichos movimientos. En este artículo se incorpora el entorno de desarrollo MoveIt como componente del sistema, el cual, se trata de un framework de software ampliamente utilizado en la comunidad robótica para la planificación de movimiento, suele integrarse estrechamente con el middleware ROS. Esta elección sugiere que MoveIt desempeña un rol fundamental en la implementación de los algoritmos de planificación del sistema descrito. El lenguaje de programación Python es fundamental para la implementación de la configuración inicial de la cámara y el robot manipulador, el procesamiento de imágenes y el control del brazo robótico.

[1]https://www.kaggle.com/datasets/nexuswho/laboro-tomato

[2]https://www.kaggle.com/datasets/andrewmvd/tomato-detection

[3]https://github.com/Hibikino-Musashi-Home/rcj2016_object_image_dataset

2.4. Entorno de pruebas

El entorno de un invernadero presenta complejidades significativas para la implementación de sistemas robóticos. Estas complejidades son influenciadas por la naturaleza de los cultivos (incluyendo el color, tamaño, forma variable, textura y ubicación de los frutos) y el entorno en sí mismo. Un desafío importante para la visión artificial en este contexto son las condiciones variables de iluminación, así como la oclusión de cultivos y la variabilidad en su apariencia. La cámara de profundidad utilizada, la Intel RealSense D435, emplea tecnología Active IR Stereo que la hace menos vulnerable a la luz ambiental y le permite operar eficazmente en condiciones de baja luminosidad o completa oscuridad. Se menciona que la cámara funciona de manera óptima en un rango de profundidad entre 0,3 y 3 metros en interiores, donde se encuentran las condiciones más favorables.

Aunque el objetivo final es la recolección en un invernadero real, las fuentes sugieren que el entorno de prueba y validación del sistema de visión se llevó a cabo en condiciones que reproducen o simulan una plantación de tomates (León *et al.*, 2024), en un ambiente controlado (Figura 1). Se utilizan conjuntos de datos públicos que incluyen imágenes de plantaciones de tomates en el interior de invernaderos para el entrenamiento del modelo de detección. El sistema de visión artificial busca detectar y clasificar tomates según su madurez (Fully-ripe, Semi-ripe, Unripe), lo que implica la necesidad de reconocer las características visuales asociadas a cada etapa de maduración bajo las mencionadas condiciones ambientales.

Figura 1: Reproducción de plantación tomates.

3. **Resultados**

Para evaluar la fiabilidad del sistema de detección de objetos, se utilizaron diversas métricas estándar en aprendizaje automático, las cuales son necesarias para comprender el rendimiento del modelo entrenado. Se calcularon indicadores como True Positive (TP) (predicción correcta de la presencia de un objeto), False Positive (FP) (predicción incorrecta de un objeto), True Negative (TN) (reconocimiento correcto de la ausencia de un objeto), y False Negative (FN) (fallo en la identificación de un objeto). A partir de estos indicadores, se derivaron métricas clave como la precisión (proporción de verdaderos positivos respecto al total de detecciones) y Recall (capacidad del modelo para identificar correctamente todos los casos positivos, proporción de verdaderos positivos respecto a la suma de TP y FN). En la Figura 2 se representan las curvas de precisión de Recall, donde se puede observar que Unripe muestra el mejor resultado de precisión frente al resto de curvas.

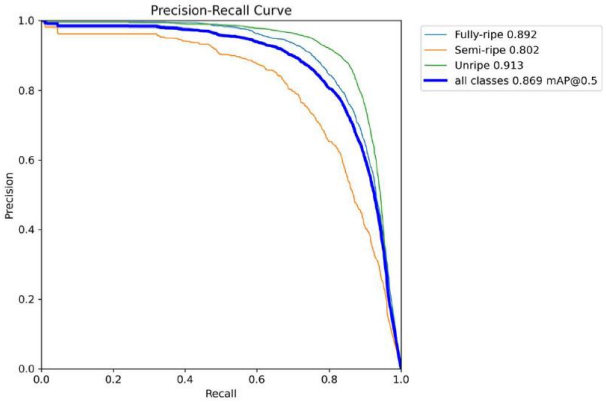

Figura 2: Gráfica de la curva Precision-Recall.

Además, se empleó Intersection over Union (IoU), una métrica que compara la similitud entre las cajas delimitadoras predichas y las reales, y Mean Average Precision (mAP), que promedia la precisión para cada clase, con versiones específicas como mAP50 (IoU > 0.50) y mAP50-95 (IoU entre 0.50 y 0.95). Otras métricas de entrenamiento visualizadas incluyeron Box loss (diferencia de IoU entre cajas predichas y reales), Class loss (diferencia entre clases predichas y reales), y Object loss (diferencia entre objetos predichos y existentes). En la fase de entrenamiento y validación, se compararon dos arquitecturas de YOLOv8. Estas fueron YOLOv8n (nano) y YOLOv8s (small), las cuales se muestran en la Tabla 1, entrenadas durante 80 épocas con el mismo conjunto de datos.

Tabla 1: Comparación de métricas para las arquitecturas entrenadas

Modelo	YOLOv8n	YOLOv8s
Precisión	0,817	0,825
Recall	0,771	0,796
mAP 50	0,843	0,850
mAP 50-95	0,693	0,710
Inferencia Media (ms)	240	628

Aunque YOLOv8s mostró métricas ligeramente superiores, los resultados no mostraron una mejora significativa en comparación con el modelo más ligero.

Por otro lado, el análisis de la matriz de confusión normalizada (Figura 3) proporcionó detalles sobre los casos de éxito y fallos específicos. Los casos de éxito se reflejan en la diagonal principal, que representa los verdaderos positivos, siendo mayor del 85% para cada clase. Esto indica una alta tasa de detección correcta para los tomates etiquetados. Los casos de fallo más notables se identificaron como falsos positivos (FP) en la detección de tomates verdes con el fondo de la imagen. Esto podría deberse a que el modelo detecta hojas con baja confianza como tomates inmaduros o a la existencia de tomates verdes sin etiquetar. La matriz de confusión mostró que la clase menos representada (semi-maduros) tuvo una tasa de verdaderos positivos ligeramente menor.

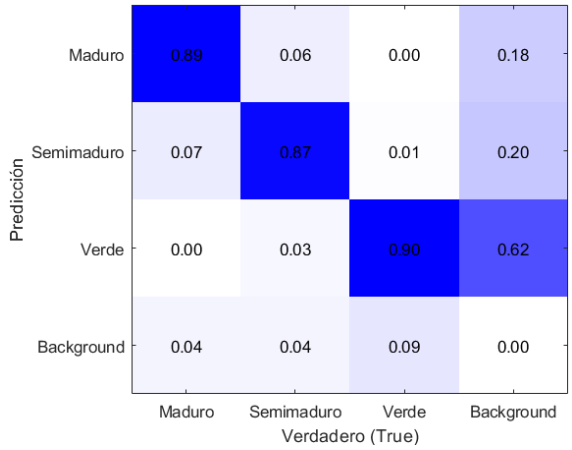

Figura 3: Matriz de confusión normalizada.

Respecto al tiempo de respuesta o velocidad de inferencia, se realizaron pruebas para determinar la rapidez de procesamiento de cada modelo, un parámetro crucial para aplicaciones en tiempo real. Utilizando un script en Python y OpenCV para procesar fotogramas de video, se obtuvieron los siguientes tiempos de inferencia promedio por fotograma en el mismo equipo de procesamiento, como se muestra en la Tabla 1, para YOLOv8n una inferencia de 240 ms y para YOLOv8s una inferencia de 628 ms. Se observó una diferencia de casi medio segundo por fotograma, siendo YOLOv8n significativamente más rápido. Dado que YOLOv8n presentó métricas de detección solo ligeramente inferiores, pero una inferencia mucho menor, fue seleccionado como el modelo a utilizar para la implementación de la aplicación en tiempo real. La optimización del modelo post-entrenamiento mediante OpenVINO y cuantificación (reducción de precisión de fp32 a int8) se plantea como una estrategia para mejorar aún más la velocidad de inferencia y reducir el uso de memoria, especialmente en hardware Intel.

4. Discusión

Los resultados obtenidos en este estudio demuestran que la implementación de un sistema de visión artificial basado en aprendizaje profundo es una alternativa viable para la automatización de la recolección de tomates en entornos de invernadero. El uso de la arquitectura YOLOv8, particularmente su versión ligera YOLOv8n, permitió alcanzar un equilibrio adecuado entre precisión y velocidad de inferencia, lo cual es esencial para aplicaciones en tiempo real. Aunque YOLOv8s ofreció un rendimiento ligeramente superior en métricas como mAP y precisión, su mayor tiempo de inferencia (más del doble que YOLOv8n) lo hace menos adecuado para sistemas robóticos en movimiento que requieren respuesta inmediata. Esto refuerza la elección de modelos optimizados y ligeros cuando se busca integración directa con hardware embebido y sistemas de control en robots manipuladores móviles.

En cuanto al desempeño del sistema de detección, los valores altos de precisión y recall (>0.77 en ambos modelos) indican una capacidad sólida para identificar y clasificar tomates en diferentes estados de madurez. Sin embargo, las dificultades observadas con los tomates verdes, particularmente los falsos positivos generados por hojas o fondos con colores similares sugieren que el sistema aún es susceptible a errores en escenarios con alta oclusión o bajo contraste. Este comportamiento era previsible dadas las condiciones variables de iluminación y complejidad visual en los entornos de invernadero. Estos resultados están en línea con lo reportado por otros autores que subrayan la sensibilidad de los sistemas de visión artificial a las condiciones ambientales y la necesidad de continuar entrenando los modelos con imágenes más representativas y variadas.

Estos resultados fueron contrastados con los obtenidos por otros autores. (Li *et al.*, 2023), quienes utilizaron una versión mejorada de YOLOv5s, denominada YOLOv5s-tomato, con técnicas como Mosaic augmentation, CSPNet y una función de pérdida basada en EIoU. Su modelo alcanzó una precisión de 95.58%, un mAP de 97.42% y un tiempo de inferencia de 9.2 ms por imagen en dispositivos agrícolas móviles. Por otro lado, (Zeng *et al.*, 2023) aplicaron una versión optimizada de YOLOv5s con MobileNetV3 y poda de canales, obteniendo una precisión del 93% y mAP de 96.9%, con una velocidad de 42.5 ms por imagen en CPU y 26.5 FPS en dispositivos Android.

Otros trabajos emplearon arquitecturas más ligeras. (Alvarez *et al*., 2023) usaron YOLOv3-tiny para detectar seis etapas de madurez, logrando un F1-score de 90% en un conjunto de 3,000 imágenes. Por su parte, (Nugroho, *et al*. 2022) emplearon YOLOv4, alcanzando una precisión de 94.6% sobre un dataset de 400 imágenes, con un rendimiento balanceado frente a Faster R-CNN y SSD.

En comparación, los modelos YOLOv8 entrenados en este trabajo muestran una precisión y recall competitivos, especialmente considerando que no se aplicaron modificaciones específicas a la arquitectura ni técnicas avanzadas de optimización como en los casos de (Li *et al*., 2023) o (Zeng *et al*., 2023). Aunque los valores de mAP y precisión son inferiores en términos absolutos, los modelos presentados demuestran robustez y desempeño aceptable para aplicaciones prácticas. Además, el tiempo de inferencia de YOLOv8n (240 ms) lo posiciona como una opción viable para aplicaciones casi en tiempo real en sistemas embebidos o móviles, aunque todavía por debajo de las implementaciones optimizadas específicamente para dispositivos móviles como las propuestas por (Zeng *et al*. 2023).

La matriz de confusión muestra una detección precisa para las clases de tomates maduros y verdes, pero una menor efectividad en la categoría intermedia (semi-maduros). Esto podría estar relacionado con la escasez de ejemplos en el conjunto de datos de entrenamiento o con la ambigüedad visual de esta clase. Por lo que se debería mejorar el balance de las clases en futuras iteraciones del sistema, mediante técnicas de aumento de datos o mediante la recolección de nuevas imágenes etiquetadas de este tipo.

5. Conclusiones

En términos generales, se validó la factibilidad técnica de un sistema de recolección autónoma basado en visión artificial, destacando la importancia de seleccionar correctamente los algoritmos y dispositivos según las restricciones operativas del entorno. No obstante, se identifican áreas de mejora como la robustez ante falsos positivos, la mejora en la clasificación de estados intermedios de madurez y la optimización del sistema para operar bajo recursos computacionales limitados.

Este proyecto sienta las bases para el desarrollo de sistemas de recolección autónoma en entornos agrícolas, sin embargo,

existen múltiples líneas de mejora y expansión que pueden abordarse en trabajos futuros. Uno de los principales retos identificados es la alta variabilidad del entorno de cultivo, especialmente en lo que respecta a las condiciones de iluminación, la forma y el color de los frutos, así como la oclusión parcial causada por hojas u otros elementos. Para enfrentar estos desafíos, una posible evolución del sistema consiste en la incorporación de algoritmos de detección tridimensional basados en redes neuronales entrenadas con datos espaciales. El uso de cámaras RGB-D permitiría generar nubes de puntos que, al correlacionarse correctamente, podrían mejorar tanto la precisión en la detección como la orientación de las herramientas de recolección, incluso en condiciones de oclusión.

Otro enfoque prometedor es la integración del sistema con plataformas de agricultura inteligente, lo que permitiría una gestión coordinada de todo el ciclo productivo, desde la siembra hasta la recolección. Esto incluye la posibilidad de conectar la visión artificial con herramientas de análisis agronómico que faciliten la toma de decisiones basada en datos. Donde se puede pronosticar su uso no solo para guiar el proceso de recolección, sino también para generar datos valiosos sobre el cultivo. Por ejemplo, se podrían estimar pesos aproximados de la producción (kg de tomates por planta o por superficie), detectar signos tempranos de plagas, deficiencias nutricionales o estrés hídrico, y emitir alertas automáticas para intervenciones agronómicas. Estos avances permitirían a los agricultores optimizar sus recursos y mejorar la sostenibilidad de sus cultivos.

Asimismo, la incorporación en sistemas robóticos colaborativos representa una línea de desarrollo de gran interés. El uso de esta tecnología no solo mejoraría la percepción espacial del entorno, sino que permitiría una navegación más segura y precisa en entornos compartidos con operarios humanos u otros robots. Los datos tridimensionales obtenidos por LiDAR podrían complementar la visión artificial en tiempo real, favoreciendo una mejor toma de decisiones en tareas de cosecha, poda o transporte automatizado. También sería interesante proponer la implementación de técnicas de fusión sensorial (como visión multiespectral o térmica), el uso de técnicas de active learning para refinar los modelos en campo, y la validación del sistema en condiciones reales de operación continua en invernaderos comerciales.

Referencias

Alvarez, G. A., Olguín-Rojas, J. C., Vasquez-Gomez, J. I., Uriarte-Arcia, A. V., Torres, M., 2023. Detection of tomato ripening stages using YOLOv3-Tiny. *arXiv.org*.

Benavides, M., Cantón-Garbín, M., Sánchez-Molina, J. A., Rodríguez, F., 2020. Automatic tomato and peduncle location system based on computer vision for use in robotized harvesting. Applied Sciences 10, 5887. DOI: 10.3390/app10175887

Brosque, C., Fischer, M., 2022. Safety, quality, schedule, and cost impacts of ten construction robots. Construction Robotics 6, 163–186. DOI: 10.1007/s41693-022-00065-6

Feng, Q., Wang, X., Wang, G., Li, Z., 2015. Design and test of tomatoes harvesting robot. In: 2015 IEEE International Conference on Information and Automation, pp. 949–952. IEEE. DOI: 10.1109/ICInfA.2015.7279428

Ji, W., Huang, X., Wang, S., He, X., 2023. A comprehensive review of the research of the "Eye–Brain–Hand" harvesting system in smart agriculture. Agronomy 13, 2237. DOI: 10.3390/agronomy13092237

León, R. A., Bravo, M. B., Castañeda, X., Juárez, S. E., Silva Gamboa, H. E., 2024. Desarrollo de visión artificial para la detección de la plaga Prodiplosis longifila o caracha en el cultivo del tomate. Memorias de la Vigésima Tercera Conferencia Iberoamericana en Sistemas, Cibernética e Informática: CISCI 2024, 424–431. DOI: 10.54808/CISCI2024.01.424

Li, R., Ji, Z., Hu, S., Huang, X., Yang, J., Li, W., 2023. Tomato maturity recognition model based on improved YOLOv5 in greenhouse. *Agronomy*.

Lubinus-Badillo, F., Rueda-Hernández, C. A., Narváez, B. M., Arias Trillos, Y. E., 2021. Convolutional neural networks, a model for deep learning in diagnostic imaging. A topic review. Revista Colombiana de Radiología 32, 5591–5599. DOI: 10.53903/01212095.161

Nugroho, D. P., Widiyanto, S., Wardani, D. T., 2022. Comparison of deep learning-based object classification methods for detecting tomato ripeness. *International Journal of Fuzzy Logic and Intelligent Systems*.

Ma, B., Hua, Z., Wen, Y., Deng, H., Zhao, Y., Pu, L., Song, H., 2024. Using an improved lightweight YOLOv8 model for real-time detection of multi-stage apple fruit in complex orchard environments. Artificial Intelligence in Agriculture 11, 70–82. DOI: 10.1016/j.aiia.2024.02.001

Mavridou, E., Vrochidou, E., Papakostas, G. A., Pachidis, T., Kaburlasos, V. G., 2019. Machine vision systems in precision agriculture for crop farming. Journal of Imaging 5, 89. DOI: 10.3390/jimaging5120089

Meshram, A. T., Vanalkar, A. V., Kalambe, K. B., Badar, A. M., 2022. Pesticide spraying robot for precision agriculture: A categorical literature review and future trends. Journal of Field Robotics 39, 153–171. DOI: 10.1002/rob.22000

Padhiary, M., Saha, D., Kumar, R., Sethi, L. N., Kumar, A., 2024. Enhancing precision agriculture: A comprehensive review of machine learning and AI vision applications in all-terrain vehicle for farm automation. Smart Agricultural Technology 100483. DOI: 10.1016/j.atech.2024.100483

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You Only Look Once: Unified, real-time object detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016, 779–788. DOI: 10.1109/CVPR.2016.91

Ruparelia, S., Jethva, M., Gajjar, R., 2021. Real-time tomato detection, classification, and counting system using deep learning and embedded systems. *Advances in Intelligent Systems and Computing*.

Sánchez-Molina, J. A., Rodríguez, F., Moreno, J. C., Sánchez-Hermosilla, J., Giménez, A., 2024. Robotics in greenhouses. Scoping review. Computers and Electronics in Agriculture 219, 108750. DOI: 10.1016/j.compag.2024.108750

Suresh Kumar, M., Mohan, S., 2023. Selective fruit harvesting: Research, trends and developments towards fruit detection and localization–A review. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 237, 1405–1444. DOI: 10.1177/09544062221146013

Tang, Y., Chen, M., Wang, C., Luo, L., Li, J., Lian, G., Zou, X., 2020. Recognition and localization methods for vision-based fruit picking robots: A review. Frontiers in Plant Science 11, 510. DOI: 10.3389/fpls.2020.00510

Zeng, T., Li, S., Song, Q., Zhong, F., Wei, X., 2023. Lightweight tomato real-time detection method based on improved YOLO and mobile deployment. *Computers and Electronics in Agriculture*.