



Universidad
Señor de Sipán

**FACULTAD DE INGENIERÍA, ARQUITECTURA Y
URBANISMO**

**ESCUELA PROFESIONAL DE INGENIERÍA DE
SISTEMAS**

TESIS

**Estimación de producción de frutos de mango
mediante análisis de imágenes y machine learning**

**PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERA
DE SISTEMAS**

Autora

Bach. Arcila Diaz Liliana Nataly
ORCID: <https://orcid.org/0009-0002-7668-0797>

Asesor

Mg. Mejia Cabrera Heber Ivan
ORCID: <https://orcid.org/0000-0002-0007-0928>

Línea de Investigación

**Ciencias de la información como herramientas multidisciplinares
y estratégicas en el contexto industrial y de organizaciones**

Sublínea de Investigación

**Informática y transformación digital en el contexto industrial y
organizacional**

**Pimentel – Perú
2025**

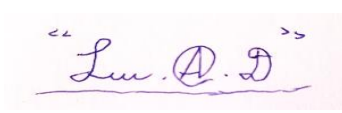
DECLARACIÓN JURADA DE ORIGINALIDAD

Quien suscribe la DECLARACIÓN JURADA, soy **Arcila Diaz Liliana Nataly** del Programa de Estudios de **Ingeniería de Sistemas** de la Universidad Señor de Sipán, declaro bajo juramento que soy autora del trabajo titulado:

ESTIMACIÓN DE PRODUCCIÓN DE FRUTOS DE MANGO MEDIANTE ANÁLISIS DE IMÁGENES Y MACHINE LEARNING

El texto de mi trabajo de investigación responde y respeta lo indicado en el Código de Ética de la Universidad Señor de Sipán, conforme a los principios y lineamientos detallados en dicho documento, en relación con las citas y referencias bibliográficas, respetando el derecho de propiedad intelectual, por lo cual informo que la investigación cumple con ser inédito, original y autentico.


En virtud de lo antes mencionado, firman:


Arcila Diaz Liliana Nataly	DNI: 48085003	
----------------------------	---------------	---


Pimentel, 15 de diciembre del 2025

Liliana Nataly Arcila Diaz

Estimación de producción de frutos de mango mediante análisis de imágenes y machine learning

 My Files

 My Files

 Universidad Señor de Sipán

Detalles del documento

Identificador de la entrega

trn:oid:::26396:426485328

Fecha de entrega

4 feb 2025, 9:50 a.m. GMT-5

Fecha de descarga

4 feb 2025, 9:52 a.m. GMT-5

Nombre de archivo

Turnitin2025final-Arcila Diaz Liliana.docx

Tamaño de archivo

7.5 MB

31 Páginas

5,950 Palabras

31,921 Caracteres



Página 2 of 36 - Descripción general de integridad

Identificador de la entrega trn:oid:::26396:426485328




13% Similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para ca...

Filtrado desde el informe

- ▶ Bibliografía
- ▶ Texto mencionado
- ▶ Coincidencias menores (menos de 8 palabras)

Fuentes principales

- 9%  Fuentes de Internet
- 1%  Publicaciones
- 5%  Trabajos entregados (trabajos del estudiante)

Marcas de integridad

N.º de alertas de integridad para revisión

No se han detectado manipulaciones de texto sospechosas.

Los algoritmos de nuestro sistema analizan un documento en profundidad para buscar inconsistencias que permitirían distinguirlo de una entrega normal. Si advertimos algo extraño, lo marcamos como una alerta para que pueda revisarlo.

Una marca de alerta no es necesariamente un indicador de problemas. Sin embargo, recomendamos que preste atención y la revise.

ESTIMACIÓN DE PRODUCCIÓN DE FRUTOS DE MANGO MEDIANTE ANÁLISIS DE IMÁGENES Y MACHINE LEARNING

Aprobación del jurado

MG. MEJIA CABRERA HEBER IVAN

Presidente de Jurado de Tesis

MG. GUEVARA ALBURQUEQUE LAURITA BELEN

Secretario de Jurado de Tesis

MG. VIDAURRE FLORES MIGUEL ANGEL

Vocal de Jurado de Tesis

DEDICATORIA

A mis hermanos, Juan Carlos, Luis Alberto y Cesy Milagros, por su cariño, su apoyo incondicional y por ser una fuente constante de motivación en cada etapa de mi vida.

A mis docentes, por las valiosas enseñanzas que me han brindado a lo largo de mi carrera universitaria. Su compromiso con mi formación académica y personal ha sido fundamental para alcanzar este logro.

La autora

AGRADECIMIENTOS

Con el amor más sincero a Dios, nuestro Padre Celestial, por concederme la salud y la fortaleza necesarias para alcanzar este objetivo tan significativo en mi vida.

A mis padres, Juan Arcila Fernandez y Ana Diaz Mundaca por apoyarme en todo momento. Todos mis logros son suyos.

A mi asesor, Mg. Heber Ivan Mejia Cabrera por su orientación en el desarrollo de esta investigación.

Al Mg. Juan Carlos Arcila Diaz, por su constante disposición para brindarme el apoyo necesario en cada momento de mi vida y con su ejemplo me impulsa a seguir adelante.

La autora

ÍNDICE

DEDICATORIA.....	v
AGRADECIMIENTOS	vi
ÍNDICE	vii
RESUMEN	ix
ABSTRACT	x
I. INTRODUCCIÓN.....	11
II. MATERIALES Y MÉTODO	17
2.1. Materiales	17
2.2. Métodos	20
III. RESULTADOS Y DISCUSIÓN.....	29
3.1. Resultados.....	29
3.2. Discusión	37
IV. CONCLUSIONES Y RECOMENDACIONES	39
4.1. Conclusiones	39
4.2. Recomendaciones	40
V. REFERENCIAS BIBLIOGRÁFICAS.....	41
VI. ANEXOS.....	45

ÍNDICE DE TABLAS

Tabla 1. Técnicas y dispositivos para la estimación de cosechas de diferentes frutos.	15
Tabla 2. Software utilizado para el entrenamiento	18
Tabla 3. Software para la aplicación web	19
Tabla 4. Entorno de entrenamiento de los modelos.....	19
Tabla 5. Resultado del proceso de aumento de datos.....	25
Tabla 6. Eficiencia de los algoritmos utilizados.....	29
Tabla 7. Desempeño de los modelos entrenados.....	30

ÍNDICE DE FIGURAS

Figura 1. Imágenes de muestra del dataset.....	18
Figura 2. Método para la estimación de producción.....	20
Figura 3. Imágenes etiquetadas con LabelImg	21
Figura 4. Imágenes segmentadas.	22
Figura 5. Formato de las etiquetas: a. YOLO, b. Pascal VOC.	24
Figura 6. Frutos de mango detectados utilizando Yolo versión 8.....	27
Figura 7. Frutos de mango detectados utilizando Faster R-CNN.....	28
Figura 8. Rendimiento de (a) YOLO versión 8; (b) Faster R-CNN.	31
Figura 9. Detección de frutos de mango con los modelos entrenados.....	32
Figura 10. Arquitectura de software de la aplicación.....	34
Figura 11. Ingreso de datos en la aplicación web	35
Figura 12. Captura de las imágenes que ingresarán a la aplicación	35
Figura 13. Imágenes por cada árbol en la aplicación.....	36
Figura 14. Frutos detectados en cada imagen de los árboles.....	36
Figura 15. Resultado de la extrapolación de frutos.....	37

ESTIMACIÓN DE PRODUCCIÓN DE FRUTOS DE MANGO MEDIANTE ANÁLISIS DE IMÁGENES Y MACHINE LEARNING

RESUMEN

La producción de mango es esencial para la economía agrícola. La estimación precisa de su producción optimiza la planificación y logística de la cosecha. Sin embargo, los métodos tradicionales son ineficientes y propensos a errores. En este estudio se analizó la detección de frutos de mango mediante algoritmos de aprendizaje automático, específicamente YOLO versión 8 y Faster R-CNN. Para el entrenamiento de los modelos, se utilizó un conjunto de datos inicial de 212 imágenes con 9,604 anotaciones, el cual fue ampliado a 2,449 imágenes y 116,654 anotaciones con el objetivo de mejorar la robustez y la capacidad de generalización de los modelos.

En la detección de frutos, YOLO alcanzó una precisión del 96.72%, un recall del 77.4% y un F1 Score del 86%, superando a Faster R-CNN, que logró una precisión del 98.57%, un recall del 63.80% y un F1 Score del 77.46%. YOLO demuestra una mayor eficiencia, siendo más rápido en el entrenamiento, consumiendo menos memoria y utilizando menos recursos de CPU. Además, en este trabajo se ha desarrollado una aplicación web con una interfaz de usuario que facilita la carga de imágenes de los árboles de mango considerados muestras. El modelo entrenado con YOLO detecta los frutos de cada árbol en la muestra representativa y utiliza técnicas de extrapolación para estimar el número total de frutos en toda la población de árboles de mango.

Palabras clave: detección de frutos, estimación de producción, YOLO, Faster R-CNN, extrapolación.

ABSTRACT

Mango production is essential to the agricultural economy. Accurate estimation of its production optimizes crop planning and logistics. However, traditional methods are inefficient and error prone. In this study, mango fruit detection was analyzed using machine learning algorithms, specifically YOLO version 8 and Faster R-CNN. For training the models, an initial dataset of 212 images with 9,604 annotations was used, which was extended to 2,449 images and 116,654 annotations in order to improve the robustness and generalization capability of the models.

In fruit detection, YOLO achieved an accuracy of 96.72%, a recall of 77.4% and an F1 Score of 86%, outperforming Faster R-CNN, which achieved an accuracy of 98.57%, a recall of 63.80% and an F1 Score of 77.46%. YOLO demonstrates higher efficiency, being faster in training, consuming less memory and using less CPU resources. In addition, this work has developed a web application with a user interface that facilitates the loading of images of mango trees considered as samples. The model trained with YOLO detects the fruits of each tree in the representative sample and uses extrapolation techniques to estimate the total number of fruits in the entire population of mango trees.

Keywords: detection of fruits, production estimation, YOLO, Faster R-CNN, extrapolation.

I. INTRODUCCIÓN

Según la Organización de las Naciones Unidas para la Agricultura y la Alimentación (FAO) dentro de las frutas tropicales el mango es el cultivo con mayor producción mundial. Esta fruta tropical no solo es apreciada por su sabor único y sus beneficios nutricionales, sino que también desempeña un papel crucial en la generación de ingresos y la estabilidad económica en diversas regiones del mundo. Como uno de los cultivos frutales más comercializados, el mango representa una fuente importante de divisas para países exportadores, contribuyendo sustancialmente a la balanza comercial. Además, el sector del mango proporciona empleo a numerosas comunidades, desde agricultores locales hasta trabajadores en la cadena de producción y distribución [1].

La producción de mango es una actividad agrícola, que representa una fuente significativa de ingresos para los agricultores. El Perú en el año 2022 logró una producción de mango de 474.000 toneladas, cultivadas en una superficie de 30.800 hectáreas. Las principales regiones productoras fueron: La región Piura con 19,867 hectáreas (64.6% del total), seguido Lambayeque con 4,311 hectáreas (14%), Áncash con 1,843 hectáreas (6%) y finalmente Cajamarca con 1,160 hectáreas (4%). La variedad de mango Kent predominó en las exportaciones, representando el 94% de los envíos [2].

La estimación precisa de la producción de árboles frutales es fundamental para la industria agrícola, ya que permite a los agricultores planificar de manera eficiente la cosecha, distribución y comercialización de los productos, de esta estimación dependen la organización y la logística necesarias para la recolección, la planificación del almacenamiento, el control, del inventario y el abastecimiento de mercados. De manera tradicional, esta tarea se ha llevado a cabo mediante métodos manuales, que son laboriosos, costosos e imprecisos debido a la variabilidad natural en el crecimiento y la producción de los árboles [3] [4].

La estimación automatizada del rendimiento en campos de frutas y hortalizas facilita una gestión más eficiente de los recursos durante la cosecha, el almacenamiento y el transporte [5]. En ausencia de sistemas automatizados para la estimación del rendimiento en cultivos

frutales, los administradores de huertos deben recurrir al recuento manual de frutos. Este método consiste en seleccionar al azar varias ramas de árboles, contar los frutos presentes y extrapolar estos resultados al total del huerto [6].

Las investigaciones científicas en este campo han explorado diversas metodologías y algoritmos de aprendizaje automático para abordar el desafío de la estimación de la producción frutal [7]. Entre estos enfoques se incluyen modelos de regresión, árboles de decisión, redes neuronales artificiales [8] [9] y técnicas de aprendizaje profundo [10], cada uno con sus ventajas y limitaciones particulares. Los estudios han demostrado que estos métodos pueden integrar diversas fuentes de datos, como imágenes satelitales, imágenes multiespectrales [11], datos climáticos, características del suelo y variables fisiológicas de los árboles, para generar estimaciones más precisas y confiables. Investigaciones similares se han llevado a cabo en diversas regiones del mundo [12] [13], con el objetivo de proporcionar información avanzada para facilitar la organización logística de esta fruta. Esto es particularmente relevante dada la corta duración entre la madurez fisiológica y la maduración en el árbol, así como la relativamente breve vida útil del producto después de la cosecha [14]. En los últimos años, el avance de las tecnologías de la información y la disponibilidad de grandes volúmenes de datos han fomentado la adopción de técnicas de aprendizaje automático (Machine Learning) en diversos ámbitos, incluida la agricultura. El aprendizaje automático ofrece una alternativa prometedora para la estimación de la producción de árboles frutales, al poder manejar grandes conjuntos de datos, identificar patrones complejos y mejorar la precisión de las predicciones [4].

En este contexto, la presente investigación formula la siguiente pregunta: ¿Es posible estimar la producción de una plantación de mango mediante el análisis de imágenes de frutos en árboles en etapa de cosecha y la aplicación de técnicas de Machine Learning?. A partir de esta interrogante, se plantea la hipótesis de que es factible estimar la producción de una plantación de mango mediante el procesamiento de imágenes de frutos en árboles en etapa de cosecha y el uso de técnicas de aprendizaje automático.

Con base en esta premisa, el objetivo general de la investigación es desarrollar un método

para la estimación de la producción de frutos de mango en la etapa de cosecha, utilizando el análisis de imágenes y técnicas de Machine Learning. Para ello, se propone como objetivos específicos la construcción de un conjunto de datos que contenga imágenes de frutos en árboles de mango en etapa de cosecha, lo que permitirá el desarrollo de un método para la detección y conteo de frutos en las plantas. Posteriormente, se evaluará la precisión y el desempeño del método propuesto con el fin de optimizar su eficacia. Finalmente, se diseñará un modelo que permita estimar la producción total de la plantación a partir del análisis de imágenes, proporcionando así una herramienta basada en inteligencia artificial para la predicción del rendimiento de los cultivos.

La realización de esta investigación se fundamenta en la necesidad de mejorar la estimación de la producción de frutos de mango en un contexto agrícola, la falta de métodos precisos y automáticos para esta estimación conlleva a ineficiencias en la cadena de suministro, pérdidas económicas y una gestión inadecuada de los recursos agrícolas. Además, las condiciones ambientales cambiantes y la falta de métodos precisos existentes plantean un desafío constante para los agricultores y la industria agrícola en su conjunto.

Esta investigación contribuirá a cubrir un vacío en el conocimiento al explorar la aplicación de técnicas de análisis de imágenes y Machine Learning en el contexto específico de la producción de mango. Los resultados obtenidos proporcionarán una base sólida para abordar un problema real y brindarán nuevas perspectivas sobre cómo mejorar la estimación de la producción agrícola

La estimación precisa de la producción de cultivos frutales no solo contribuye a una mejor planificación agrícola, sino que también optimiza la cadena de suministro, reduce el desperdicio y mejora la sostenibilidad de los sistemas agrícolas. Por lo tanto, la implementación de técnicas de aprendizaje automático en la estimación de la producción de mango surge como un enfoque innovador, que permite el desarrollo de aplicaciones para su uso en la agricultura dentro de las comunidades de países en desarrollo tecnológico.

En el trabajo [13], se desarrolló un modelo basado en redes neuronales convolucionales (CNN) para estimar la producción de cítricos utilizando imágenes RGB capturadas por un

dron, vehículo aéreo no tripulado (UAV). El entrenamiento del modelo se llevó a cabo durante tres temporadas de producción anuales (2017, 2018 y 2019) en una parcela de 4 hectáreas. Para este fin, seleccionaron aleatoriamente 20 árboles de un total de 1.654 en la parcela y tomaron fotografías desde los lados izquierdo y derecho de cada árbol. Durante las tres campañas, el modelo mostró resultados favorables, con una precisión del 96% y un error entre el 4% y el 6%, en comparación con un error de entre el 8% y el 11% por parte de un técnico especialista. No obstante, el modelo presentó dificultades para identificar frutos ocultos entre las ramas.

En la investigación [15], propusieron un método eficiente para el reconocimiento de cítricos a través de la optimización del detector de última generación You Only Look Once versión 4 (YOLOv4) [16]. Se empleó una cámara Kinect V2 para capturar imágenes RGB de los árboles de cítricos, para seleccionar automáticamente el número y tamaño de los marcos iniciales a partir de estas imágenes utilizaron los algoritmos Canopy y K-Means++. Se propuso una estructura mejorada de la red YOLOv4 para optimizar la detección de cítricos más pequeños en entornos complejos. Finalmente, el modelo de red entrenado se sometió a un entrenamiento esparso, que implicó la poda de canales o capas no esenciales y el ajuste fino de los parámetros del modelo podado para recuperar la precisión de reconocimiento. Los resultados experimentales indican que el detector mejorado YOLOv4 es eficaz para detectar diferentes etapas de crecimiento de los cítricos en entornos naturales, con un incremento promedio en la precisión del 3.15% (de 92.89% a 96.04%). Este resultado supera al YOLOv4 original, YOLOv3 y Faster R-CNN. El tiempo promedio de detección de este modelo es de 0.06 segundos por cuadro a una resolución de 1920 × 1080.

En el trabajo [12], se abordó el desafío de la detección y conteo de mangos en condiciones de oclusión, es decir en condiciones en las que el fruto de mango está parcial o completamente bloqueado por hojas, tallos u otros objetos del árbol. Se propuso un método que evaluó el filtro de color y las características específicas de la fruta, como la homogeneidad de la superficie. Se destacó la eficiencia del método al utilizar información extraída de blobs (objeto binario grande) creados después de un filtrado de histograma, incluyendo

ponderación, evaluación de la topografía del gradiente, y agrupación jerárquica, sin necesidad de determinar previamente la cantidad de grupos. Utilizando 150 imágenes de mango divididas en 30 imágenes para entrenamiento y 120 imágenes para prueba, Los resultados mostraron que este método pudo detectar mangos con precisión y tasas de error de hasta un 97.53% y 0.72%, respectivamente.

En la tabla 1, se presentan otras investigaciones pertinentes sobre el recuento de frutos y la estimación de la cosecha, los equipos empleados para la adquisición de datos entre otros.

Tabla 1. Técnicas y dispositivos para la estimación de cosechas de diferentes frutos.

Fruto	Técnica	Datos	Dispositivo	Ventajas y desventajas del dispositivo
Manzana [17]	YOLO v4-tiny	Imágenes RGB	Sensor Kinect V2 con una resolución 1920 x 1080 píxeles	El dispositivo tiene una alta resolución y un amplio campo de visión, lo que permite una captura precisa de los detalles de la fruta. Sin embargo, existen limitaciones para el uso en exteriores, ya que la luz solar puede afectar la precisión del sensor de profundidad.
Cítricos [18]	YOLO v5-CS	Imágenes RGB	DJI MAVIC Air2, con cámara SLR (Panasonic DMC-G7) y un teléfono móvil Honor 20	Proporcionan imágenes de alta calidad con excelente resolución, nitidez y control sobre parámetros como la exposición, el enfoque y la apertura. En este estudio, se utilizaron en entornos no controlados, lo que disminuye la calidad de la imagen.
Uva [19]	Artificial neural (ANN)	Imágenes satelitales	Landsat 8 OLI serie de tiempo	El sensor OLI presenta nueve bandas espectrales que cubren diferentes rangos del espectro. Sin embargo, su resolución espacial de 30 m puede ser insuficiente para estudios que requieren un análisis detallado de objetos pequeños.
Melón [20]	RetinaNet	Imágenes RGB	UAV Phantom 4 Pro equipado con una cámara a	Este dispositivo permite un análisis detallado y una cobertura eficiente a 15 m sobre el campo, ofreciendo una buena estabilidad y facilidad de

			color DJI FC6310 RGB, a 15 m sobre el campo con una resolución de 5472 × 3648 píxeles	uso. Sin embargo, su rendimiento depende de las condiciones climáticas, y el procesamiento de imágenes requiere recursos de almacenamiento sustanciales y capacidad computacional.
Manzana [21]	YOLO v5	Imágenes RGB	UAV y una cámara Raspberry Pi	La cámara Raspberry Pi es de bajo costo y altamente personalizable, facilitando su integración con algoritmos de detección de frutas como YOLOv5. Sin embargo, la calidad de imagen de los cámaras Raspberry Pi es limitada en comparación con las cámaras más avanzadas, lo que puede afectar la precisión de la detección de frutas.
Fresa [8]	R-CNN	Imágenes RGB	Teléfono móvil con una resolución de 4608 × 3456 en formato JPG	Este dispositivo captura imágenes de alta resolución, el formato JPG produce archivos livianos. Sin embargo, la proximidad requerida para la captura de imágenes (0.2–0.3 m) podría limitar la eficiencia, ya que se necesitarían múltiples tomas para cubrir un área grande del cultivo.
Fresa [11]	Regresión lineal	Imágenes multispectrales	RPA Phantom 4 Pro Mapir Survey3W cámara con una resolución: 12 megapíxeles	Este dispositivo captura imágenes multispectrales, ofreciendo datos detallados sobre la salud y el crecimiento de las fresas. Con una resolución de 12 megapíxeles y cobertura aérea, permite una evaluación precisa del cultivo. No obstante, demanda una capacidad de almacenamiento significativa y un alto poder de procesamiento para la adquisición de imágenes.
Pimiento [22]	DeepSOR T	Imágenes RGB	Cámara D435i RGB-D	Captura imágenes en color y datos de profundidad, mejorando la precisión en la detección de la madurez y el conteo de la fruta al proporcionar información de distancia y forma. Sin embargo,

				esta integración añade complejidad y requiere más potencia computacional.
Mango [9]	Faster R-CNN	Imágenes RGB	Teléfono móvil con una resolución: 1920 x 1080 píxeles	Este dispositivo es fácil de usar para los agricultores, aunque las imágenes pueden exhibir baja calidad en condiciones de iluminación variables.

Fuente: Elaboración del autor

II. MATERIALES Y MÉTODO

2.1. Materiales

1. Conjunto de datos de imágenes de árboles frutales en etapa de producción

En esta investigación se ha utilizado conjuntos de datos públicos; el primero es presentado por [23], que contiene 49 imágenes de árboles de mango de la variedad Mangifera Indica, en la etapa de producción. Se trata de imágenes JPG con dimensiones de 4000 x 3000 píxeles, adquiridas utilizando una cámara RGB de alta resolución. Estas imágenes fueron capturadas entre la 1:00 p. m. y las 3:00 p. m. en una tarde luminosa de abril, durante el verano tropical. El experimento se llevó a cabo en un huerto de mango ubicado en el pueblo de Mudimadagu, con una latitud de 13,56 N y una longitud de 78,36 E, dentro de la subdivisión Rayalpad del taluk de Srinivaspur, una región conocida por el cultivo de mango en el sureste de la India. El segundo conjunto de datos se publicó en GitHub y contiene 123 imágenes de árboles de mango en la etapa de producción. Se trata de imágenes JPG con dimensiones de 640 x 640 píxeles. Otro conjunto de datos consta de 40 imágenes con una resolución de 612 x 408 píxeles en formato JPG. En la figura 1 se aprecia dos imágenes de muestra del dataset.



Figura 1. Imágenes de muestra del dataset

Fuente: [23]

2. Herramientas de software

En la tabla 2, se enumeraron las herramientas utilizadas en el entrenamiento del modelo de detección de frutos de mango.

Tabla 2. Software utilizado para el entrenamiento

Tipo	Nombre	Licencia de Uso	Descripción
Anotaciones	LabelImg	MIT	Software para hacer las anotaciones de los objetos (fruto) en la imagen.
Arquitectura CNN	YOLO v8	MIT	Algoritmo CNN para la detección en tiempo real de objetos.
Arquitectura CNN	Faster RCNN	MIT	Algoritmo CNN para la detección en tiempo real de objetos.
Lenguaje de programación	Python	Python license	Lenguaje de programación de alto nivel, interpretado y de propósito general.
Librería	PyTorch	BSD	Biblioteca de aprendizaje automático de código abierto basada en la biblioteca de Torch.

Fuente: Elaboración del autor

En la tabla 3, se enumeraron las herramientas que se necesitó para desarrollar la aplicación web que permitió estimar la producción de frutos de mango en tiempo real.

Tabla 3. Software para la aplicación web

Tipo	Nombre	Licencia de Uso	Descripción
Lenguaje de programación	JavaScript	GNU	Lenguaje de programación del lado del cliente.
Lenguaje de programación	Python	Python License	Lenguaje de programación del lado del servidor.
Framework	Flask	Licencia BSD-3	Framework para crear aplicaciones web en Python.

Fuente: Elaboración del autor

3. Servicio

Para el entrenamiento del modelo de detección se ha utilizado el servicio de Google Collaboratory cuyas características se enumeran en la tabla 4.

Tabla 4. Entorno de entrenamiento de los modelos.

Nombre	Característica	Valor
	Velocidad del procesador	8 CPU virtuales de 3,7 GHz
Google Collaboratory	Almacenamiento	78.2 Gb
GPU NVIDIA T4	Memoria RAM	12.7 Gb
	Memoria GPU	GDDR6 de 15 GB

Fuente: Elaboración del autor

4. Hardware

En la tabla 5, se describen las características del equipo donde se ha desplegado la aplicación web.

Tabla 5. Características del servidor para el despliegue de la aplicación.

Componente	Característica	Valor
Servidor Web	Velocidad del procesador	2 CPU 2,70 GHz
	Almacenamiento	10 Gb
	Memoria RAM	8 Gb
	Servidor Web	Flask (Python 3.3)

Fuente: Elaboración del autor

2.2. Métodos

En la figura 2, se ha representado de manera esquemática las etapas que se han empleado en la investigación para estimar la producción de frutos de mango mediante análisis de imágenes y machine learning.

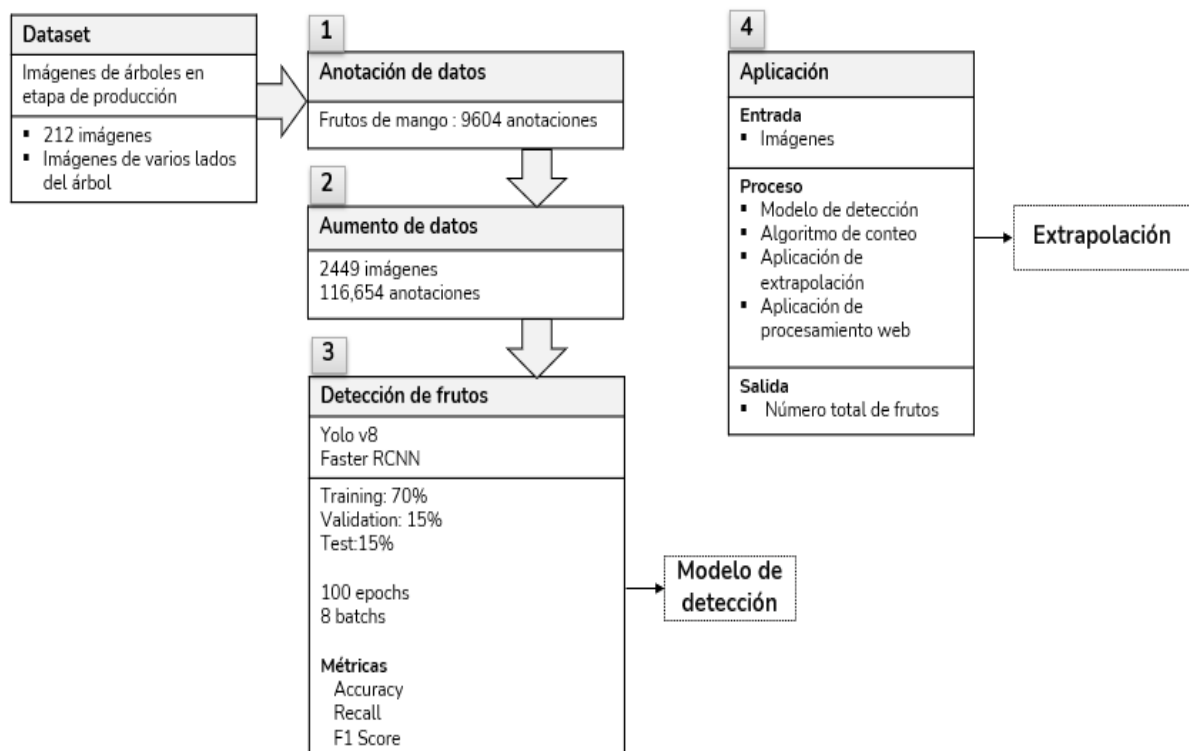


Figura 2. Método para la estimación de producción.

Fuente: Elaboración del autor

1. Anotación de datos

Las imágenes del dataset fueron etiquetadas de manera manual con la herramienta LabelImg en el formato YOLO. Para utilizar el algoritmo Faster R-CNN se ha elaborado las anotaciones en el formato Pascal VOC. En la figura 3 se aprecian dos imágenes de muestra etiquetadas utilizando la herramienta LabelImg.

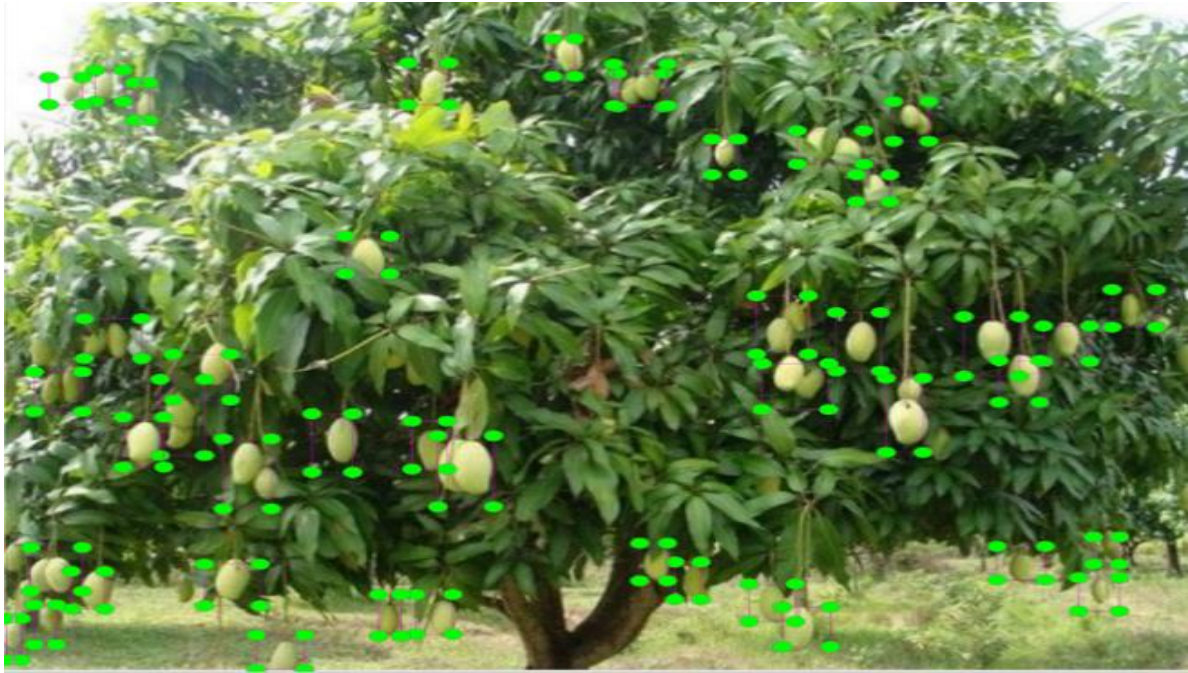


Figura 3. Imágenes etiquetadas con LabelImg

Fuente: Elaboración del autor

El dataset publicado en [23], incluye variantes segmentadas en las que se resalta la ubicación de cada mango mediante un color verde, mientras que el fondo se presenta en color negro. En la figura 4, se aprecia 1 muestra en la izquierda la imagen original y a su derecha la imagen segmentada. Se aprovechó esta segmentación para utilizarlo en un algoritmo automatizado para etiquetar los objetos identificados en color verde.



Figura 4. Imágenes segmentadas.

Fuente: [23]

Para desarrollar el algoritmo de etiquetado se realizaron los siguientes pasos:

Inicialmente se definió el rango de color verde en el espacio de color HSV (Hue, Saturation, Value):

```
verde_inferior = np.array([40, 40, 40])
```

```
verde_superior = np.array([80, 255, 255])
```

La imagen se cambió de color RGB a HSV, luego se creó una máscara binaria donde los

píxeles dentro del rango verde son blancos (255) y los demás son negros (0).

$$máscara(x, y) = \begin{cases} 255 & \text{si } verde_inferior \leq imagen_{hsv}(x, Y) \leq verde_superior \\ 0 & \text{en caso contrario} \end{cases} \quad (1)$$

Luego se detectaron los contornos de la máscara y para cada contorno se calculó el cuadro delimitador (bounding box) y se transformó sus coordenadas al formato YOLO, por consiguiente, se implementaron las ecuaciones 2, 3, 4 y 5.

$$x_{centro} = \frac{x + \frac{w}{2}}{ancho} \quad (2)$$

$$y_{centro} = \frac{y + \frac{h}{2}}{altura} \quad (3)$$

$$w_{norm} = \frac{w}{ancho} \quad (4)$$

$$h_{norm} = \frac{h}{altura} \quad (5)$$

Donde:

x: es la coordenada x del borde izquierdo del cuadro delimitador.

y: es la coordenada y del borde superior del cuadro delimitador

w: es la anchura del cuadro delimitador

h: es la altura del cuadro delimitador

ancho: es la anchura total de la imagen

altura: es la altura total de la imagen

Finalmente, esta información se almacenó en un archivo txt para ser utilizado en el entrenamiento. Se ha verificado después que el etiquetado sea correcto y en algunos casos se ha realizado el ajuste manual de la etiqueta. En la figura 5, se aprecia la captura de ambos formatos ya que son esenciales para el entrenamiento de los modelos.

```

Class_102: Bloc de notas
Archivo Edición Formato Ver Ayuda
0 0.714625 0.8548333333333333 0.00025 0.001
0 0.687125 0.8336666666666667 0.00075 0.0006666666666666666
0 0.6875 0.8331666666666667 0.0005 0.00033333333333333333
0 0.687625 0.8323333333333334 0.00025 0.0006666666666666666
0 0.591125 0.8306666666666667 0.00025 0.0026666666666666666
0 0.706625 0.8281666666666667 0.00125 0.0023333333333333335
0 0.70075 0.8583333333333333 0.0345 0.08333333333333333
0 0.556625 0.7861666666666667 0.00025 0.0003333333333333333
0 0.569 0.8241666666666667 0.045 0.10633333333333334
0 0.572125 0.7708333333333334 0.00025 0.0003333333333333333
0 0.571375 0.7703333333333333 0.00075 0.0006666666666666666
0 0.638125 0.719 0.00025 0.0006666666666666666
0 0.62225 0.6965 0.0005 0.0003333333333333333
0 0.623375 0.7376666666666667 0.03375 0.082666666666666667

```

(a)

```

voc_class_102: Bloc de notas
Archivo Edición Formato Ver Ayuda
0 2858 2563 2859 2566
0 2747 2500 2750 2502
0 2749 2499 2751 2500
0 2750 2496 2751 2498
0 2364 2488 2365 2496
0 2824 2481 2829 2488
0 2734 2450 2872 2700
0 2226 2358 2227 2359
0 2186 2313 2366 2632
0 2288 2312 2289 2313
0 2284 2310 2287 2312
0 2552 2156 2553 2158
0 2488 2089 2490 2090
0 2426 2089 2561 2337

```

(b)

Figura 5. Formato de las etiquetas: a. YOLO, b. Pascal VOC.

Fuente: Elaboración del autor

En el formato YOLO, las coordenadas del cuadro delimitador se definen normalmente como las coordenadas del centro del cuadro delimitador, junto con su anchura y altura relativas a las dimensiones de la imagen. Para utilizar el algoritmo Faster R-CNN, teniendo en cuenta el tamaño de la imagen, las anotaciones se han convertido al formato Pascal VOC [24]. Las coordenadas en formato YOLO son $(x_{center}, y_{center}, w, h)$, donde x_{center} e y_{center} están son las coordenadas del centro del cuadro delimitador (bounding box), seguido w y h son la anchura y la altura. Por consiguiente, todas están normalizadas en el rango $[0, 1]$.

Para convertir estas coordenadas al formato Pascal VOC $(x_{min}, y_{min}, x_{max}, y_{max})$, se emplearon las siguientes fórmulas:

$$x_{min} = x_{center} - \frac{w}{2} \quad (6)$$

$$y_{min} = y_{center} - \frac{h}{2} \quad (7)$$

$$x_{max} = x_{center} + \frac{w}{2} \quad (8)$$

$$y_{max} = y_{center} + \frac{h}{2} \quad (9)$$

2. Aumento de datos

Para expandir el conjunto de datos de imágenes de mangos, se ha empleado la técnica de aumento de datos, la cual ha demostrado ser eficaz para mitigar el sobreajuste [25]. El proceso de aumento incluye la aplicación de varias transformaciones geométricas y distorsiones a las imágenes como escalado, rotación, recorte aleatorio, volteo vertical, volteo horizontal y mejora del contraste. Estas transformaciones permiten incrementar la variedad de las imágenes para el entrenamiento del modelo y mejorar la precisión.

El aumento de datos se ha realizado utilizando rotaciones progresivas que van de -5° a $+5^\circ$, simulando diferentes ángulos de visión de los frutos en el árbol (Ecuación (1)), donde θ es el ángulo de rotación.

$$rotation = I_{rot}(x', y') = I(x \cos(\theta) - y \sin(\theta), x \sin(\theta) + y \cos(\theta)) \quad (10)$$

Tabla 5. Resultado del proceso de aumento de datos.

Imágenes originales		Aumento de datos	
Imágenes	Anotaciones	Imágenes	Anotaciones
212	9604	2449	116.654

Fuente: Elaboración del autor

3. Detección de frutos

Se compararon dos modelos de detección de objetos los cuales son: YOLO versión 8 y Faster R-CNN y se midió su eficiencia. El dataset se dividió un 70% para el training, 15% para la validación y 15% para el test. Ambos fueron entrenados durante 100 epochs con un batch de tamaño 8.

- **YOLO:** Es una familia de modelos de detección de objetos que divide la imagen de entrada en una cuadrícula y predice simultáneamente múltiples bounding boxes y sus correspondientes probabilidades de clase [26].

División de la Imagen en una Cuadrícula:

La imagen de entrada se divide en una cuadrícula de $S \times S$ celdas. En este trabajo se utilizaron imágenes de 4000 pixeles de ancho y 3000 pixeles de alto, entonces para esa imagen se aplicó un factor de reducción de 32 (factor de reducción típica de YOLO) la cuadrícula se calcula de la siguiente manera:

$$S_x = \frac{\text{ancho_imagen}}{\text{factor}} = \frac{4000}{32} = 125 \quad (11)$$

$$S_y = \frac{\text{alto_imagen}}{\text{factor}} = \frac{3000}{32} = 93.75 \approx 94 \quad (12)$$

La cuadrícula resultante sería aproximadamente de 125 x 94, este cálculo asegura que la imagen de alta resolución se maneje eficientemente, permitiendo a YOLO predecir objetos con alta precisión a través de múltiples celdas de la cuadrícula. En la figura 6 se aprecia las imágenes de árboles de mango con sus frutos detectados utilizando el modelo entrenado con YOLO.





Figura 6. Frutos de mango detectados utilizando Yolo versión 8.

Fuente: Elaboración del autor

- **Faster R-CNN.-** Este método combina la propuesta de regiones y la detección de objetos en un solo modelo, mejorando significativamente la velocidad y la precisión en comparación con métodos anteriores como R-CNN y Fast R-CNN [27]. El proceso general de Faster R-CNN incluye tres fases:

Extracción de características

Para extraer características de cada imagen de entrada, se emplea la red convolucional profunda ResNet50 [28].

Dada la imagen de entrada I de dimensiones $H \times W \times 3$, se obtiene un mapa de características F de dimensiones $H_f \times W_f \times D$, donde H_f y W_f son menores que H y W debido a las operaciones de reducción de la red.

$$F = \text{ConvNet}(I) \quad (13)$$

Propuestas de regiones

La red de propuestas de regiones (RPN) es una red completamente

convolucional que genera propuestas de regiones, que son posibles ubicaciones donde pueden estar presentes los objetos.

Refinamiento y clasificación: las propuestas generadas por la RPN se utilizan para extraer regiones de interés (Rois) del mapa de características. Estas Rois se normalizan a una dimensión fija y se pasan a través de una capa de agrupamiento (RoI Pooling o RoI Align), que las convierte en mapas de características de tamaño fijo. En la figura 7 se aprecia las imágenes de árboles de mango con sus frutos detectados utilizando el modelo entrenado con FASTER R-CNN.



Figura 7. Frutos de mango detectados utilizando Faster R-CNN.

Fuente: Elaboración del autor

III. RESULTADOS Y DISCUSIÓN

3.1. Resultados

1. Evaluación de los resultados para la detección y conteo de frutos de mango

➤ **Eficiencia.** - Para la evaluación de los dos algoritmos de detección de objetos, se midieron tres parámetros clave: consumo de memoria, uso de CPU y tiempo de procesamiento. Las mediciones se realizaron utilizando el lenguaje de programación Python y el siguiente enfoque:

Tiempo de Procesamiento

Se midió el tiempo total que cada algoritmo tardó en completarse utilizando la función `time.time()`.

Consumo de Memoria

Se utilizó la biblioteca `tracemalloc` para registrar el consumo máximo de memoria durante la ejecución de cada algoritmo.

Consumo de CPU

Se utilizó la biblioteca `psutil` para medir el uso de CPU del proceso correspondiente.

Los resultados de las mediciones se muestran en la siguiente tabla.

Tabla 6. Eficiencia de los algoritmos utilizados

Eficiencia		
Parámetro	YOLO	Faster R-CNN
Uso de almacenamiento (GB)	32	32
Tiempo de Procesamiento (s)	8046.65	13.551,23
Consumo de Memoria RAM (MB)	5302.20	7450.30
Consumo de Memoria GPU (GB)	13.9	16
Consumo de CPU (%)	58.70	84.75

Fuente: Elaboración del autor

➤ **Desempeño.** - Se utilizó el dataset para el entrenamiento de dos modelos de detección de objetos YOLO versión 8 y Faster R-CNN. El dataset se dividió un 70% para el training, 15% para la validation y 15% para el test. Ambos fueron entrenados durante 100 epochs con un batch de tamaño 8.

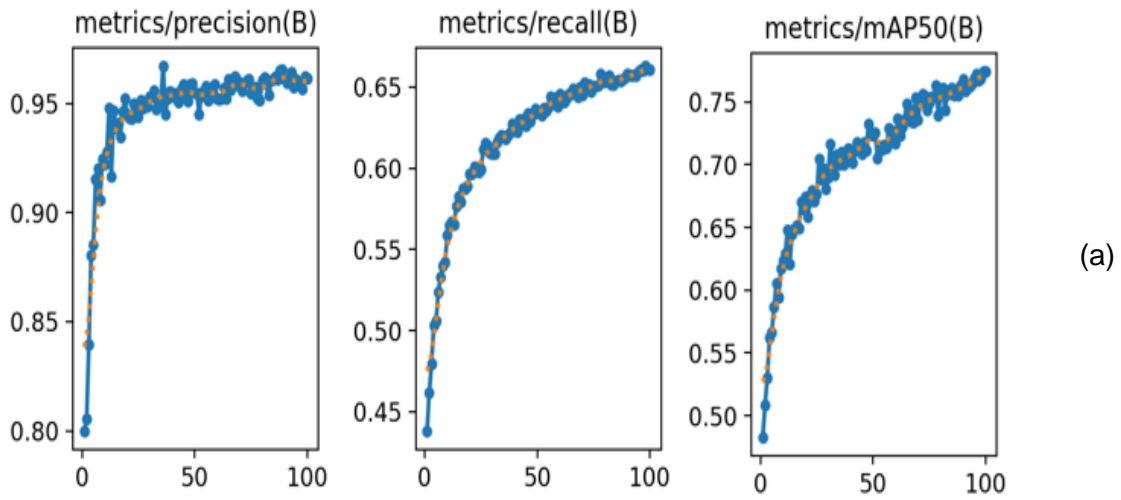
En la tabla 7, se observa el modelo de detección entrenado con YOLO versión 8 y demostró un sólido desempeño en la identificación de frutos de mangos durante la etapa de producción. Con una precisión del 96,72%, el modelo logró una alta proporción de predicciones correctas entre todas las detecciones realizadas. Un recall del 77.4% que refleja la capacidad de los modelos para identificar correctamente una parte significativa de los casos positivos del conjunto de datos. Finalmente, el puntaje F1 del 86% equilibra la precisión lo que indica un rendimiento general robusto. En cuanto a Faster R-CNN logró una precisión del 98.57%, un recall del 63.80% y un puntaje F1 del 77.46%. este modelo presentó un rendimiento inferior en comparación con el modelo entrenado con YOLO.

Tabla 7. Desempeño de los modelos entrenados

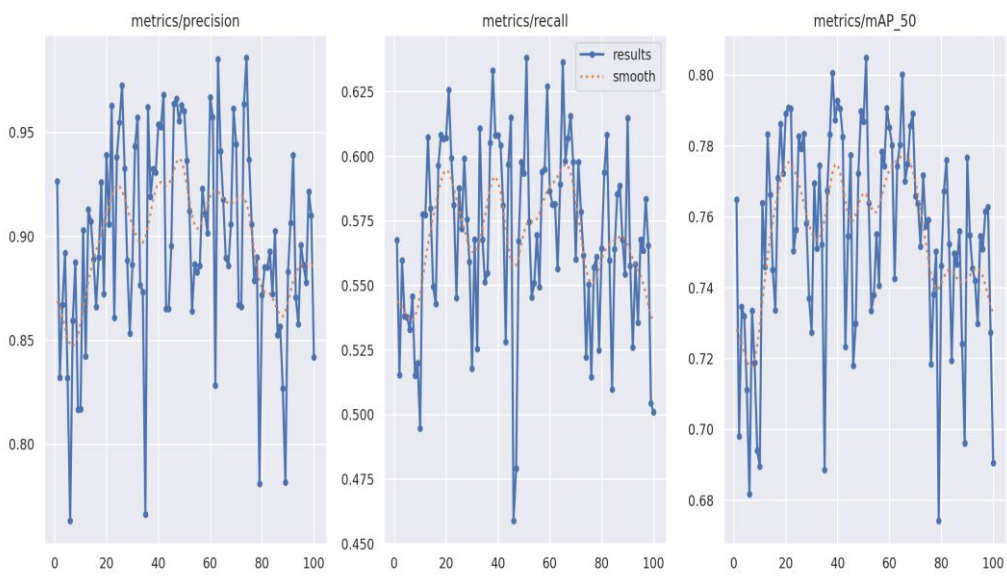
Parámetro	YOLO	Faster R-CNN
Precisión (%)	96.72	98.57
Recall (%)	77.4	63.80
F1 Score (%)	86	77.46

Fuente: Elaboración del autor

En la figura 8, se muestra a detalle el rendimiento del modelo entrenado con YOLO versión 8 y Faster R-CNN en 100 épocas.



(a)



(b)

Figura 8. Rendimiento de (a) YOLO versión 8; (b) Faster R-CNN.
Fuente: Elaboración del autor

Estos modelos entrenados pudieron procesar una imagen de prueba para detectar frutos de mango en imágenes de árboles de mangos en etapa de producción.

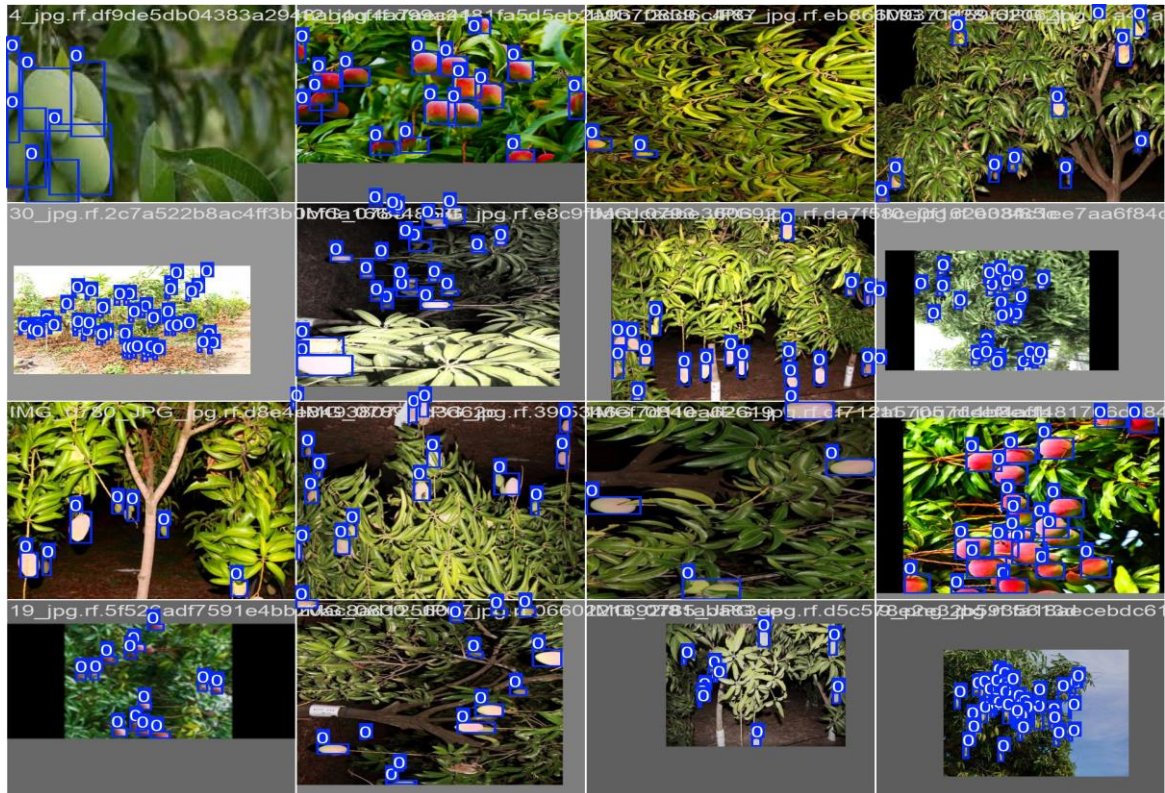


Figura 9. Detección de frutos de mango con los modelos entrenados.

Fuente: Elaboración del autor

2. Método para estimar la producción de una plantación

Para realizar el conteo de mangos se ha ingresado inicialmente la cantidad de árboles frutales, para cada árbol se agregó una o varias fotos, de tal manera que se considere todos los frutos del árbol. Luego se ha determinado la cantidad de frutos por árbol de la muestra y mediante la extrapolación se estimará la producción total de frutos de mangos de todos los árboles.

Inicialmente se ha calculado el promedio de frutos por árbol (\bar{x}):

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i \quad (14)$$

donde X_i es la cantidad de frutos en el árbol i .

Luego calculamos la desviación estándar (S), utilizando la siguiente ecuación:

$$S = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2} \quad (15)$$

Luego calculamos el Error estándar de la media (SE):

$$SE = \frac{S}{\sqrt{n}} \quad (16)$$

Ahora extrapolamos la población (N árboles):

$$L_{inf} = (\bar{X} - t \times SE) \times N \quad (17)$$

$$L_{sup} = (\bar{X} + t \times SE) \times N \quad (18)$$

El intervalo de confianza esta dado por:

$$\bar{X} \pm t \times SE \quad (19)$$

Donde t es el valor crítico del t- distribution para un nivel de confianza específico (por ejemplo, 95%) y n -1 grados de libertad.

3. Aplicación web

En la figura 10, se ha ilustrado la arquitectura de la aplicación web desarrollada, que está diseñada en base a un modelo cliente servidor. La interfaz de usuario se ha implementado utilizando el lenguaje de programación JavaScript, complementado con HTML y CSS para la creación del entorno visual. Esta interfaz facilita la ejecución de las solicitudes necesarias al servidor, permitiendo la entrada de datos y el envío de imágenes de los árboles para su posterior procesamiento. Los datos se envían desde la interfaz de usuario al servidor a través de solicitudes utilizando el Protocolo de Transferencia de Hipertexto (HTTP) en formato JavaScript Object Notation (JSON).

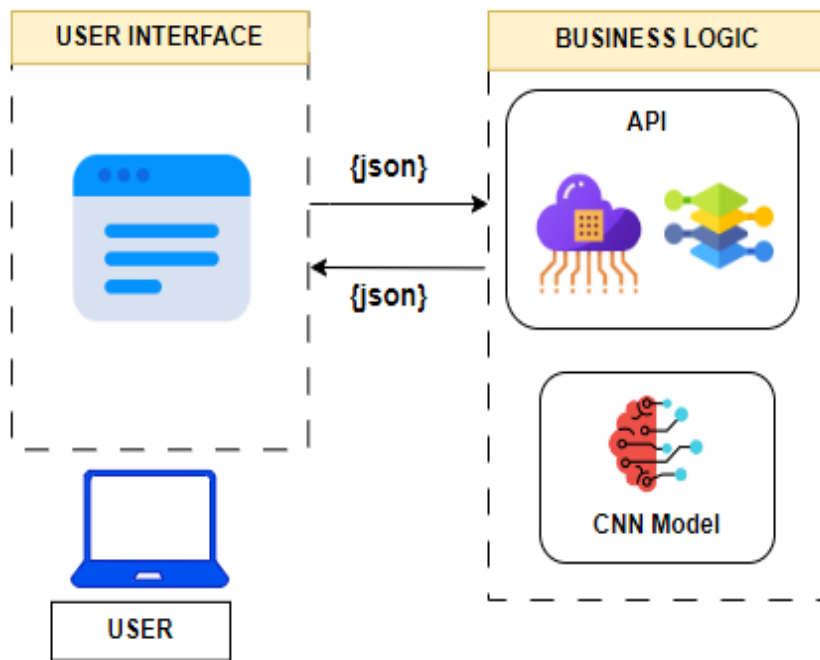


Figura 10. Arquitectura de software de la aplicación.
Fuente: Elaboración del autor

La capa de negocio se ha desarrollado en Python, utilizando el framework Flask. Esta capa es responsable de recibir solicitudes a través de puntos finales y realizar el procesamiento necesario. Dentro de esta lógica empresarial reside el modelo entrenado, que está diseñado para detectar frutos de mango en los árboles. Este modelo toma una imagen de entrada y, como resultado, proporciona el número de frutas de mango identificadas dentro de la imagen. Inicialmente, la aplicación se ha implementado en un entorno local con fines de prueba. Al acceder a la aplicación web, los usuarios pueden especificar el número total de árboles de mango presentes en su área de cultivo y también deben ingresar el número de árboles que constituirán la muestra representativa (Figura 11)

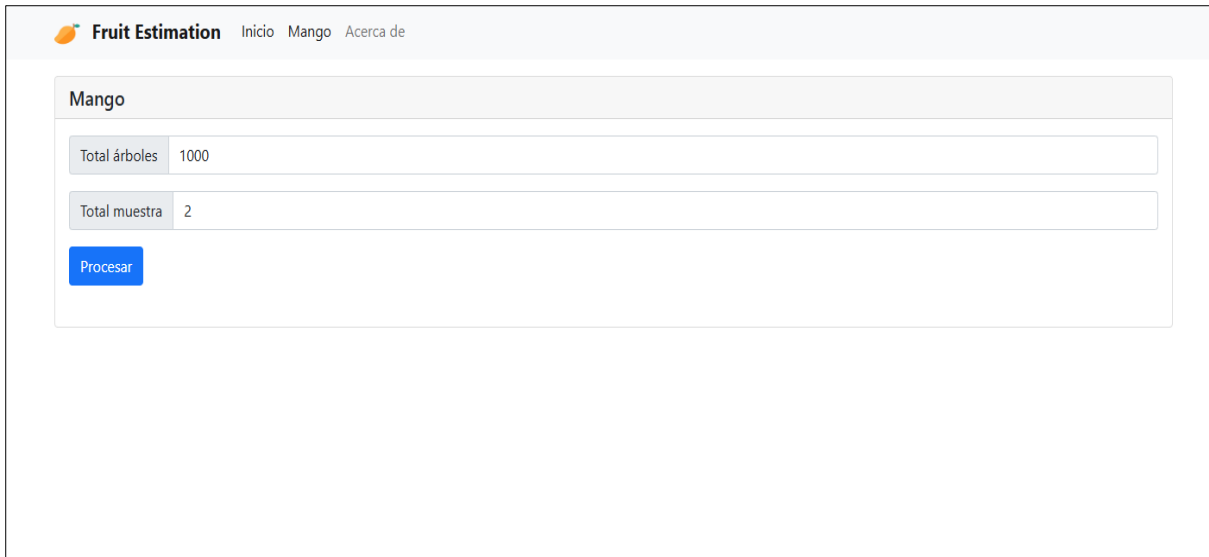


Figura 11. Ingreso de datos en la aplicación web

Fuente: Elaboración del autor

Para cada árbol, el usuario debe capturar fotografías que incluyan todas las frutas visibles en el árbol; estas imágenes deben seleccionarse específicamente para cada árbol. En la figura 12, se aprecia el ingreso de imágenes para cada árbol.

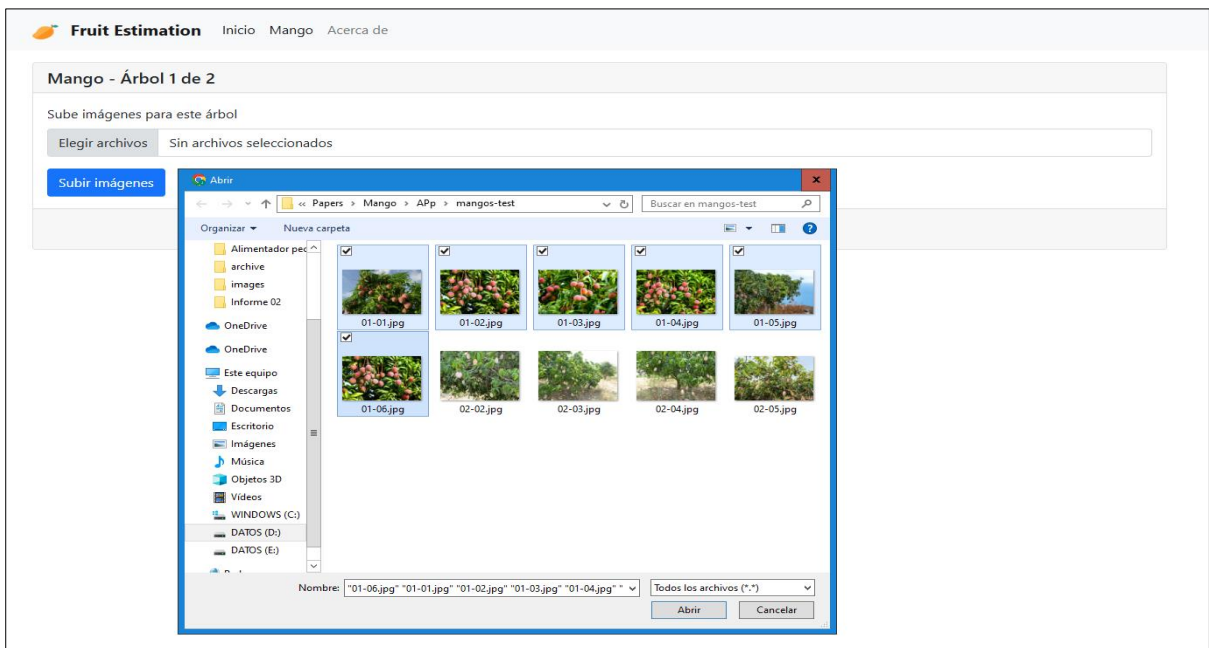


Figura 12. Captura de las imágenes que ingresarán a la aplicación

Fuente: Elaboración del autor

Antes de realizar el procesamiento se mostrará un resumen de todos los árboles con sus imágenes ingresadas. En la figura 13, se aprecia este resumen.

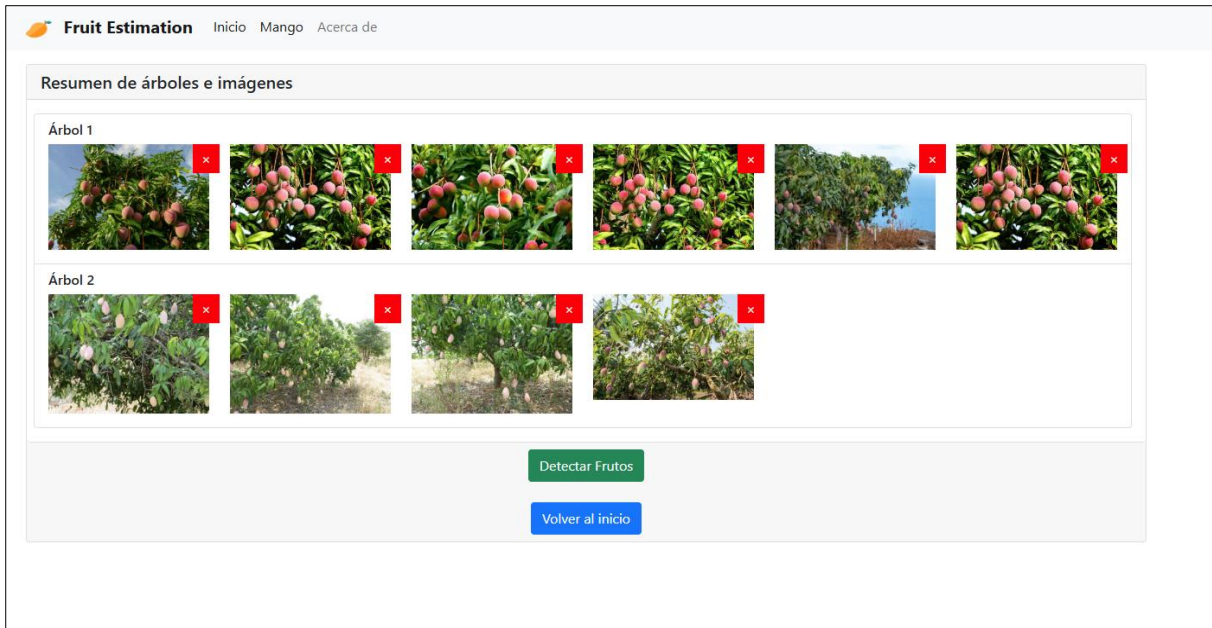


Figura 13. Imágenes por cada árbol en la aplicación

Fuente: Elaboración del autor

En la detección de frutos, la aplicación mostrará automáticamente la cantidad de frutas identificadas en cada imagen del árbol (Figura 14).

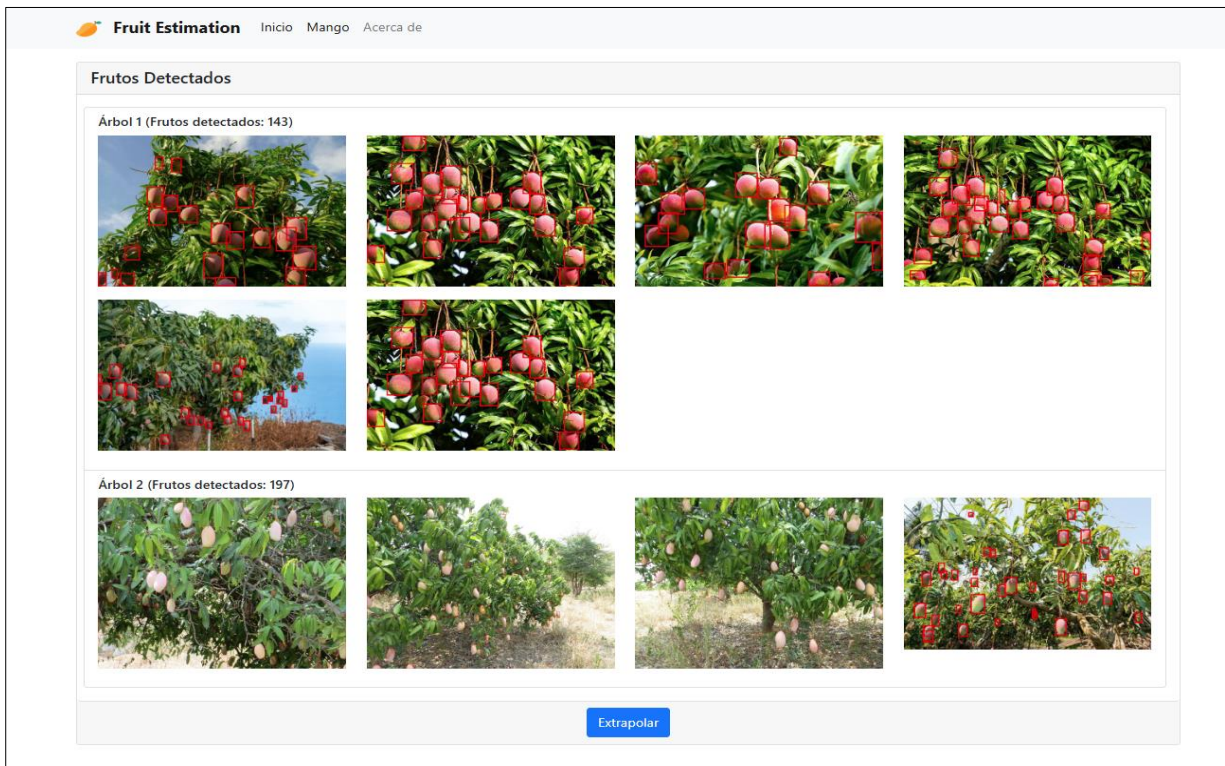


Figura 14. Frutos detectados en cada imagen de los árboles.

Fuente: Elaboración del autor

Finalmente, la aplicación presenta los resultados de la extrapolación realizada, indicando el rango promedio de frutos de mango presentes en el cultivo (Figura 15).

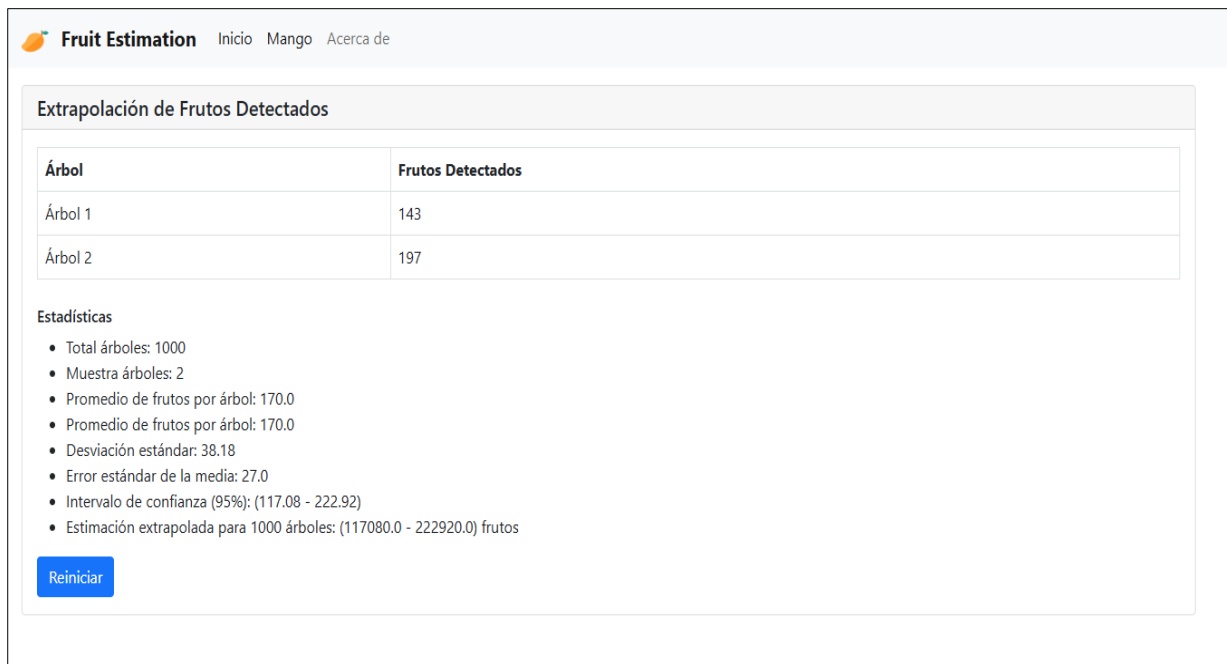


Figura 15. Resultado de la extrapolación de frutos.

Fuente: Elaboración del autor

3.2. Discusión

La estimación de la producción de frutos y el conteo de frutos se basan en el análisis de imágenes como datos de entrada. Algunos estudios emplean imágenes de alta resolución, que permiten capturar detalles precisos de los frutos; sin embargo, este enfoque necesita una mayor capacidad de almacenamiento y procesamiento de datos [17] [20]. Otros estudios utilizan imágenes obtenidas a través de un UAV [11] [18] [21]; sin embargo, la aplicación de estos resultados requiere que los agricultores inviertan en equipos, lo que a menudo implica costos excesivos. Nuestra investigación, similar a la de [8] [9], utiliza imágenes capturadas por teléfonos inteligentes, lo que facilita una adopción más económica y factible por parte de los agricultores.

En el trabajo de [13], se empleó una red neuronal convolucional (CNN) para estimar la producción de cítricos utilizando imágenes capturadas por drones. Aunque los autores lograron una precisión del 96% y una tasa de error del 4-6%, encontraron dificultades para

identificar frutos ocluidos. Por lo tanto, esta investigación se centra en la estimación de la producción de mango y utiliza el modelo YOLO, logrando una precisión del 96,72%. Sin embargo, el modelo presentado en [13] enfrentó desafíos para identificar frutos ocluidos, una limitación que también puede ser pertinente para nuestro estudio, donde la oclusión de frutos por hojas o ramas podría afectar la precisión de la detección.

En el estudio de [12], la detección del mango en condiciones de oclusión se abordó utilizando filtros de color y características específicas de la fruta, logrando una precisión del 97,53% y una tasa de error del 0,72%. Aunque nuestro método no se centra específicamente en las condiciones de oclusión, es más generalizable y adaptable a diferentes entornos, lo que lo hace robusto para diversas aplicaciones.

Mientras que el estudio de [9] emplea el modelo ResNet-50, centrándose en un enfoque de red neuronal convolucional más tradicional para la estimación del rendimiento, nuestra investigación aprovecha modelos avanzados que facilitan la detección en tiempo real, lo cual es crucial en entornos agrícolas dinámicos. Aunque ambos estudios tienen como objetivo estimar la producción agrícola, nuestra investigación no solo enfatiza la precisión, logrando una exactitud del 96,72%, sino que también prioriza la aplicación práctica al brindarles a los agricultores un servicio para estimar su producción de mango.

En comparación con el trabajo presentado en [12], nuestra investigación implementa un enfoque más sólido que utiliza técnicas de aumento de datos. El marco metodológico empleado en nuestro estudio se basa en modelos avanzados de detección de objetos, específicamente YOLO versión 8 y Faster R-CNN, que facilitan el análisis en tiempo real y mejoran la capacidad de identificar objetos en diferentes condiciones de iluminación y complejidades de fondo. Nuestra investigación no solo mejora la precisión de la detección del mango, sino que también proporciona un marco para la implementación práctica de estas tecnologías en la agricultura.

IV. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

El dataset empleado en este trabajo comprendía un total de 212 imágenes originales, cada una con 9604 anotaciones. Tras aplicar técnicas de aumento de datos, se generaron 2449 imágenes adicionales, lo que dio como resultado 116.654 anotaciones. Esto representa un aumento sustancial en la cantidad de datos disponibles para el entrenamiento, lo cual es crucial para mejorar la robustez y la generalización de los modelos de detección. El aumento significativo en el número de imágenes y anotaciones fortalece la capacidad del modelo para aprender de una mayor diversidad de ejemplos, optimizando así su rendimiento en tareas de detección y clasificación.

El algoritmo YOLO supera a Faster R-CNN en términos de tiempo de procesamiento, consumo de memoria y uso de CPU durante el entrenamiento. Estas ventajas hacen de YOLO la opción preferida para entornos de desarrollo rápido con limitaciones de recursos.

Los resultados obtenidos a partir de la evaluación de los modelos YOLO y Faster R-CNN demuestran que YOLO supera a Faster R-CNN, logrando una precisión del 96,72% y un puntaje F1 del 86%.

Se utilizó la extrapolación para calcular el número total de frutos en árboles de mango en función de la cantidad observada en una muestra representativa de árboles. Esta técnica permite una estimación precisa del número total de frutos en una población más grande a partir de los datos recopilados en la muestra, lo que proporciona una visión más completa de la producción de mango en los árboles estudiados.

Finalmente, se desarrolló una aplicación web que permite a los agricultores cargar imágenes de sus árboles y recibir una estimación de la producción de mango. Si bien esta aplicación representa un avance en la integración de tecnologías para la agricultura, se concluye que este tipo de soluciones deben evolucionar hacia servicios accesibles para los agricultores, que les permitan beneficiarse de estas estimaciones de manera consistente y eficiente.

Este proyecto representa un esfuerzo inicial en nuestra región para estimar la producción de mango, un cultivo de gran importancia local. En trabajos futuros, se desarrollará un conjunto

de imágenes de mango de la zona, asegurando que las imágenes se tomen desde una distancia consistente entre el dispositivo de captura y el árbol. En cuanto al procesamiento de imágenes, se utilizarán las últimas versiones de YOLO y se probarán otras arquitecturas de detección de objetos. La aplicación desarrollada en este estudio estará disponible para todos los agricultores a través de servicios de computación en la nube, lo que garantiza una amplia accesibilidad y escalabilidad.

4.2. Recomendaciones

- Expandir el enfoque utilizado en esta investigación para incluir otros tipos de frutas y cultivos. Adaptar y validar el modelo para diferentes productos agrícolas aumentará el impacto y la aplicabilidad de la tecnología en la agricultura de precisión, beneficiando a una mayor diversidad de cultivos.
- Fomentar colaboraciones entre ingenieros de sistemas, agrónomos y expertos en aprendizaje automático es esencial para seguir desarrollando y perfeccionando la tecnología. La combinación de conocimientos permitirá abordar problemas complejos desde diferentes perspectivas y mejorar el rendimiento del sistema.
- Se recomienda explorar técnicas avanzadas de preprocesamiento de imágenes, como el uso de filtrado adaptativo y normalización de iluminación, para mejorar la calidad de las imágenes antes del entrenamiento del modelo. Esto podría contribuir a una mayor precisión en la detección de frutos y a una reducción en los errores de predicción.
- Es fundamental realizar pruebas en entornos reales de cultivo, considerando variaciones en iluminación, clima y densidad de follaje, para evaluar la robustez del modelo en condiciones no controladas. Esto permitiría ajustar el sistema a escenarios más complejos y mejorar su aplicabilidad en la agricultura de precisión.

V. REFERENCIAS BIBLIOGRÁFICAS

- [1] Agraria.pe, “Agroindustria generó 1.3 millones de puestos de trabajo entre enero y octubre de 2022.” Accessed: Jan. 10, 2023. [Online]. Available: <https://agraria.pe/noticias/agroindustria-genero-1-3-millones-de-puestos-de-trabajo-entr-30353>
- [2] Agraria.pe, “Producción nacional de mango alcanzó las 474.000 toneladas en 2022.” Accessed: Jan. 10, 2023. [Online]. Available: <https://agraria.pe/noticias/produccion-nacional-de-mango-alcanzo-las-474-000-toneladas-e-30987>
- [3] S. Manonmani, S. Senthilkumar, U. S. A. Govind, and S. Manivannan, “Application of Artificial Intelligence in Fruit Production: A Review,” *Agricultural Science Digest - A Research Journal*, no. Of, Apr. 2022, doi: 10.18805/ag.D-5482.
- [4] A. Koirala, K. B. Walsh, Z. Wang, and C. McCarthy, “Deep learning – Method overview and review of use for fruit detection and yield estimation,” *Computers and Electronics in Agriculture*, vol. 162, pp. 219–234, Jul. 2019, doi: 10.1016/j.compag.2019.04.017.
- [5] N. Häni, P. Roy, and V. Isler, “A comparative study of fruit detection and counting methods for yield mapping in apple orchards,” *Journal of Field Robotics*, vol. 37, no. 2, pp. 263–282, 2020, doi: 10.1002/rob.21902.
- [6] A. Koirala, K. B. Walsh, Z. Wang, and C. McCarthy, “Deep learning – Method overview and review of use for fruit detection and yield estimation,” *Computers and Electronics in Agriculture*, vol. 162, pp. 219–234, Jul. 2019, doi: 10.1016/j.compag.2019.04.017.
- [7] A. A. Boatswain Jacques, V. I. Adamchuk, J. Park, G. Cloutier, J. J. Clark, and C. Miller, “Towards a Machine Vision-Based Yield Monitor for the Counting and Quality Mapping of Shallots,” *Frontiers in Robotics and AI*, vol. 8, Apr. 2021, doi: 10.3389/frobt.2021.627067.
- [8] J. Li, Z. Zhu, H. Liu, Y. Su, and L. Deng, “Strawberry R-CNN: Recognition and counting model of strawberry based on improved faster R-CNN,” *Ecological Informatics*, vol. 77, p. 102210, Nov. 2023, doi: 10.1016/j.ecoinf.2023.102210.
- [9] N. Managali Vasanth and R. PPandian, “Fast region based convolutional neural network

- ResNet-50 model for on tree Mango fruit yield estimation,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 33, no. 2, p. 1084, Feb. 2024, doi: 10.11591/ijeecs.v33.i2.pp1084-1091.
- [10] D. J. G. Romero, O. E. Apolo-Apolo, and M. P. Ruíz, “Estimating Optimal Harvest Time and Yield in Tomatoes Using Deep Learning Techniques: A Preliminary Study,” in *2023 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor)*, IEEE, Nov. 2023, pp. 81–85. doi: 10.1109/MetroAgriFor58484.2023.10424131.
- [11] L. S. de Oliveira, R. Castoldi, G. D. Martins, and M. H. Medeiros, “Estimation of Strawberry Crop Productivity by Machine Learning Algorithms Using Data from Multispectral Images,” *Agronomy*, vol. 13, no. 5, p. 1229, Apr. 2023, doi: 10.3390/agronomy13051229.
- [12] A. Hutagalung, H. Nugroho, A. Suheryadi, and P. E. Yunanto, “Detection and Counting of Mango Fruits in Occluded Condition Using Image Analysis,” in *2017 5th International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME)*, 2017, pp. 190–195. doi: 10.1109/ICICI-BME.2017.8537729.
- [13] O. E. Apolo Apolo, J. Martínez-Guanter, G. Egea, and M. Pérez-Ruiz, *Estimación de producción en cítricos usando técnicas de aprendizaje automático*. 2019. doi: 10.26754/c_agroing.2019.com.3461.
- [14] N. T. Anderson *et al.*, “Estimation of Fruit Load in Australian Mango Orchards Using Machine Vision,” *Agronomy*, vol. 11, no. 9, Art. no. 9, Sep. 2021, doi: 10.3390/agronomy11091711.
- [15] W. Chen, S. Lu, B. Liu, G. Li, and T. Qian, “Detecting Citrus in Orchard Environment by Using Improved YOLOv4,” *Scientific Programming*, vol. 2020, p. 8859237, Nov. 2020, doi: 10.1155/2020/8859237.
- [16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.

- [17] F. Gao *et al.*, "A novel apple fruit detection and counting methodology based on deep learning and trunk tracking in modern orchard," *Computers and Electronics in Agriculture*, vol. 197, p. 107000, 2022, doi: <https://doi.org/10.1016/j.compag.2022.107000>.
- [18] S. Lyu, R. Li, Y. Zhao, Z. Li, R. Fan, and S. Liu, "Green Citrus Detection and Counting in Orchards Based on YOLOv5-CS and AI Edge System," *Sensors*, vol. 22, no. 2, 2022, doi: 10.3390/s22020576.
- [19] S. T. Arab, R. Noguchi, S. Matsushita, and T. Ahamed, "Prediction of grape yields from time-series vegetation indices using satellite remote sensing and a machine-learning approach," *Remote Sensing Applications: Society and Environment*, vol. 22, p. 100485, Apr. 2021, doi: 10.1016/j.rsase.2021.100485.
- [20] A. Kalantar, Y. Edan, A. Gur, and I. Klapp, "A deep learning system for single and overall weight estimation of melons using unmanned aerial vehicle images," *Computers and Electronics in Agriculture*, vol. 178, p. 105748, 2020, doi: <https://doi.org/10.1016/j.compag.2020.105748>.
- [21] Z. Y. LI Zhijun YANG Shenghui, SHI Deshuai, LIU Xingxing, "Yield Estimation Method of Apple Tree Based on Improved Lightweight YOLOv5," *Smart Agriculture*, vol. 3, no. 2, p. 100, 2021, doi: 10.12133/j.smartag.2021.3.2.202105-SA005.
- [22] L. D. Viveros Escamilla, A. Gómez-Espinosa, J. A. Escobedo Cabello, and J. A. Cantoral-Ceballos, "Maturity Recognition and Fruit Counting for Sweet Peppers in Greenhouses Using Deep Learning Neural Networks," *Agriculture*, vol. 14, no. 3, 2024, doi: 10.3390/agriculture14030331.
- [23] R. Kestur, A. Meduri, and O. Narasipura, "MangoNet: A deep semantic segmentation architecture for a method to detect and count mangoes in an open orchard," *Engineering Applications of Artificial Intelligence*, vol. 77, pp. 59–69, Jan. 2019, doi: 10.1016/j.engappai.2018.09.011.
- [24] "Pascal VOC," CVAT. Accessed: Jan. 29, 2025. [Online]. Available: <https://docs.cvat.ai/docs/manual/advanced/formats/format-voc/>
- [25] D. A. van Dyk and X.-L. Meng, "The Art of Data Augmentation," *Journal of Computational*

and Graphical Statistics, vol. 10, no. 1, pp. 1–50, Mar. 2001, doi: 10.1198/10618600152418584.

- [26] C. Yu and Y. Shin, “An efficient YOLO for ship detection in SAR images via channel shuffled reparameterized convolution blocks and dynamic head,” *ICT Express*, Feb. 2024, doi: 10.1016/j.icte.2024.02.007.
- [27] S. Ren, K. He, R. B. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *CoRR*, vol. abs/1506.01497, 2015, [Online]. Available: <http://arxiv.org/abs/1506.01497>
- [28] R. Rani, J. Sahoo, S. Bellamkonda, S. Kumar, and S. K. Pippal, “Role of Artificial Intelligence in Agriculture: An Analysis and Advancements With Focus on Plant Diseases,” *IEEE Access*, vol. 11, pp. 137999–138019, 2023, doi: 10.1109/ACCESS.2023.3339375.

VI. ANEXOS

Anexo 1. Acta de revisión de similitud de la investigación.



ACTA DE REVISIÓN DE SIMILITUD DE LA INVESTIGACIÓN

Yo **William Joel Marín Rodríguez** docente del curso de **Investigación II** del Programa de Estudios de **Ingeniería de Sistemas**, luego de revisar la investigación del estudiante, Arcila Díaz Liliana Nataly, titulada:

Estimación de producción de frutos de mango mediante análisis de imágenes y machine learning

Dejo constancia que la investigación antes indicada tiene un índice de similitud del **19%**, verificable en el reporte de originalidad mediante el software de similitud TURNITIN. Por lo que se concluye que cada una de las coincidencias detectadas no constituyen plagio y cumple con lo establecido en la Directiva sobre índice de similitud de los productos académicos y de investigación en la Universidad Señor de Sipán S.A.C. vigente.

En virtud de lo antes mencionado, firma:

Marín Rodríguez William Joel	DNI: 40398872	
------------------------------	---------------	---

Pimentel, 18 de julio de 2024.

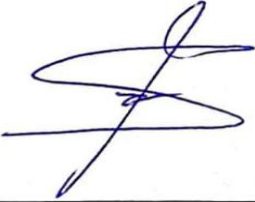
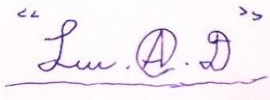
Anexo 2. Acta de aprobación de asesor.



ACTA DE APROBACIÓN DEL ASESOR

Yo **Mejía Cabrera Heber Ivan** quien suscribe como asesor designado mediante Resolución de Facultad N°0772-2023/FIAU-USS del proyecto de investigación titulado **Estimación de producción de frutos de mango mediante análisis de imágenes y machine learning**, desarrollado por el(los) estudiante(s): **Arcila Díaz Liliana Nataly**, del programa de estudios de **Ingeniería de Sistemas**, acredito haber revisado, y declaro expedito

En virtud de lo antes mencionado, firman:

Mejía Cabrera Heber Ivan (Asesor)	DNI: 41639565	
Arcila Díaz Liliana Nataly (Autor)	DNI: 48085003	

Pimentel, 15 de julio de 2024.

Anexo 3. Certificado de aceptación del manuscrito para publicación.



informatics

an Open Access Journal by MDPI



CERTIFICATE OF ACCEPTANCE



The certificate of acceptance for the manuscript (**informatics-3133371**) titled:
Estimation of Mango Fruit Production Using Image Analysis and Machine Learning Algorithms

Authored by:

Liliana Arcila-Diaz; Heber I. Mejia-Cabrera; Juan Arcila-Diaz

was accepted in *Informatics* (ISSN 2227-9709) on 23 October 2024



Academic Open Access Publishing
since 1996

Basel, October 2024

Prof. Dr. Antony Bryant
Editor-in-Chief

Anexo 4. Tabla de operacionalización de variables.

Tipo de variable	Variables de estudio	Definición conceptual	Definición operacional	Dimensiones	Indicadores	Ítems	Instrumento	Valores finales	Escala de medición
Independiente	Método para predecir la estimación de producción de cosecha de mango	Implementación de algoritmos de Machine Learning para analizar imágenes de frutos de mango en el árbol con el objetivo de predecir la estimación de producción de cosecha de mango.	Implementación práctica de algoritmos de Machine Learning aplicados a imágenes de frutos en los árboles de mango.	Eficiencia	Consumo de Memoria.	CM=CMD-CMI	Ficha de observación	0 – 100%	Porcentaje (%)
					Consumo de CPU	CCPU=CCPU D-CCPUI		1 a 3 Ghz	Gigabyte (gb)
					Tiempo de procesamiento	TDP=TF-TI		0.1 a 3000 ms	Milisegundos (ms)
Dependiente	Estimación de producción de cosecha de mango	Cantidad de frutos de mango producidos en un determinado periodo.	Medición de la cantidad total de frutos.	Desempeño	Precisión	$P=TP/(TP+FP)$	Matriz de confusión	0 – 100%	Porcentaje (%)
					Exactitud	$ACC=(TP+TN)/(TP+TN+FP+FN)$		0 – 100%	Porcentaje (%)
					Recall	$R=TP/(TP+FN)$		0 – 100%	Porcentaje (%)
					F1 Score	$F=2*(P*R)/(P+R)$		0 – 100%	Porcentaje (%)

Anexo 5. Instrumentos de recolección de datos.

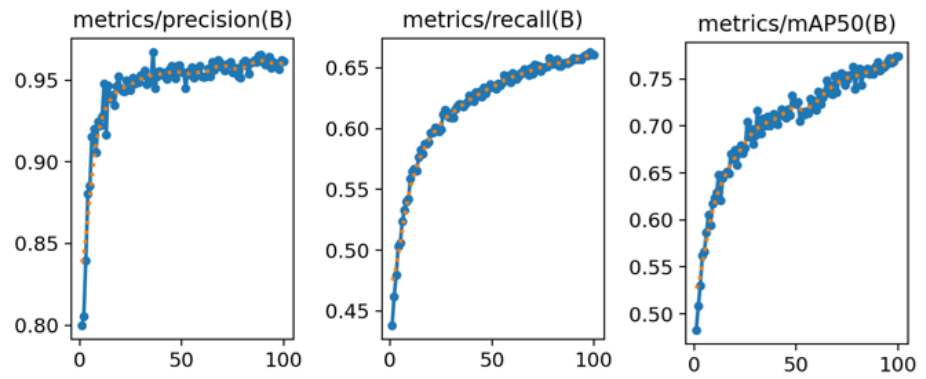
01 ficha de observación – Entrenamiento													
Nombre	Prueba para medir la eficiencia y desempeño												
Descripción	Esta prueba permitirá medir la eficiencia y desempeño del método propuesto para detectar frutos de mango y estimar la producción de un cultivo de mangos.												
Fecha de la prueba	14-07-2024												
Escenario	<ul style="list-style-type: none"> - Se ha utilizado un dataset de 2449 imágenes para el entrenamiento de nuestro método, estas imágenes JPG de dimensiones 4000 x 3000, de frutos de mango de la variedad Magnifera Indica, en etapa de producción, que fueron adquiridas utilizando una cámara de espectro de visión de alta resolución (RGB). - Las imágenes fueron divididas en 3 grupos. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Grupo</th> <th>Porcentaje</th> <th>Imágenes</th> </tr> </thead> <tbody> <tr> <td>Val</td> <td>15 %</td> <td>367</td> </tr> <tr> <td>Train</td> <td>70 %</td> <td>1741</td> </tr> <tr> <td>Test</td> <td>15 %</td> <td>368</td> </tr> </tbody> </table> <ul style="list-style-type: none"> - Valores de entrenamiento: Batch: 8 Epoch: 100 - Arquitectura utilizada: Yolo Versión 8, Faster R-CNN. 	Grupo	Porcentaje	Imágenes	Val	15 %	367	Train	70 %	1741	Test	15 %	368
Grupo	Porcentaje	Imágenes											
Val	15 %	367											
Train	70 %	1741											
Test	15 %	368											
Prerrequisitos	<ul style="list-style-type: none"> - Las imágenes deben estar agrupadas y ubicadas en el directorio del proyecto. 												

Insumos	<table border="1" style="width: 100%; text-align: center;"> <tr> <th colspan="3">Hardware</th> </tr> <tr> <th>Servicio</th> <th>Característica</th> <th>Valor</th> </tr> <tr> <td rowspan="4">Google Collaboratory GPU NVIDIA T4</td> <td>Velocidad del procesador</td> <td>8 CPU virtuales de 3,7 GHz</td> </tr> <tr> <td>Almacenamiento</td> <td>78.2 Gb</td> </tr> <tr> <td>Memoria RAM</td> <td>12.7 Gb</td> </tr> <tr> <td>Memoria GPU</td> <td>GDDR6 de 15 GB</td> </tr> </table>			Hardware			Servicio	Característica	Valor	Google Collaboratory GPU NVIDIA T4	Velocidad del procesador	8 CPU virtuales de 3,7 GHz	Almacenamiento	78.2 Gb	Memoria RAM	12.7 Gb	Memoria GPU	GDDR6 de 15 GB												
	Hardware																													
Servicio	Característica	Valor																												
Google Collaboratory GPU NVIDIA T4	Velocidad del procesador	8 CPU virtuales de 3,7 GHz																												
	Almacenamiento	78.2 Gb																												
	Memoria RAM	12.7 Gb																												
	Memoria GPU	GDDR6 de 15 GB																												
<table border="1" style="width: 100%; text-align: center;"> <tr> <th colspan="4">Software</th> </tr> <tr> <th>Tipo</th> <th>Nombre</th> <th>Licencia de Uso</th> <th>Descripción</th> </tr> <tr> <td>Anotaciones</td> <td>Labellmg</td> <td>MIT</td> <td>Software para hacer las anotaciones de los objetos (fruto) en la imagen.</td> </tr> <tr> <td>Arquitectura CNN</td> <td>Yolo V8</td> <td>MIT</td> <td>Algoritmo CNN para la detección en tiempo real de objetos.</td> </tr> <tr> <td>Arquitectura CNN</td> <td>Faster RCNN</td> <td>MIT</td> <td>Algoritmo CNN para la detección en tiempo real de objetos.</td> </tr> <tr> <td>Lenguaje de programación</td> <td>Python</td> <td>Python license</td> <td>Lenguaje de programación de alto nivel, interpretado y de propósito general.</td> </tr> <tr> <td>Librería</td> <td>PyTorch</td> <td>BSD</td> <td>Biblioteca de aprendizaje automático de código abierto basada en la biblioteca de Torch.</td> </tr> </table>			Software				Tipo	Nombre	Licencia de Uso	Descripción	Anotaciones	Labellmg	MIT	Software para hacer las anotaciones de los objetos (fruto) en la imagen.	Arquitectura CNN	Yolo V8	MIT	Algoritmo CNN para la detección en tiempo real de objetos.	Arquitectura CNN	Faster RCNN	MIT	Algoritmo CNN para la detección en tiempo real de objetos.	Lenguaje de programación	Python	Python license	Lenguaje de programación de alto nivel, interpretado y de propósito general.	Librería	PyTorch	BSD	Biblioteca de aprendizaje automático de código abierto basada en la biblioteca de Torch.
Software																														
Tipo	Nombre	Licencia de Uso	Descripción																											
Anotaciones	Labellmg	MIT	Software para hacer las anotaciones de los objetos (fruto) en la imagen.																											
Arquitectura CNN	Yolo V8	MIT	Algoritmo CNN para la detección en tiempo real de objetos.																											
Arquitectura CNN	Faster RCNN	MIT	Algoritmo CNN para la detección en tiempo real de objetos.																											
Lenguaje de programación	Python	Python license	Lenguaje de programación de alto nivel, interpretado y de propósito general.																											
Librería	PyTorch	BSD	Biblioteca de aprendizaje automático de código abierto basada en la biblioteca de Torch.																											
Indicadores de aceptación	<table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Consumo de memoria</td> <td>Los valores deben estar entre 0.0 y 1.0 La arquitectura con el menor valor es la más eficiente.</td> </tr> <tr> <td>Consumo</td> <td>La arquitectura con el menor valor es la más eficiente.</td> </tr> </table>			Consumo de memoria	Los valores deben estar entre 0.0 y 1.0 La arquitectura con el menor valor es la más eficiente.	Consumo	La arquitectura con el menor valor es la más eficiente.																							
Consumo de memoria	Los valores deben estar entre 0.0 y 1.0 La arquitectura con el menor valor es la más eficiente.																													
Consumo	La arquitectura con el menor valor es la más eficiente.																													

	<table border="1"> <tr> <td>de CPU</td> <td></td> </tr> <tr> <td>Tiempo</td> <td>La arquitectura con el menor valor es la más eficiente.</td> </tr> <tr> <td>Precisión</td> <td>Los valores deben estar entre 0.0 y 1.0 Un valor aceptable debe ser mayor a 0.9.</td> </tr> <tr> <td>Recall</td> <td>Los valores deben estar entre 0.0 y 1.0 Un valor aceptable debe ser mayor a 0.9.</td> </tr> <tr> <td>F1 Score</td> <td>Los valores deben estar entre 0.0 y 1.0 Un valor aceptable debe ser mayor a 0.9.</td> </tr> </table>	de CPU		Tiempo	La arquitectura con el menor valor es la más eficiente.	Precisión	Los valores deben estar entre 0.0 y 1.0 Un valor aceptable debe ser mayor a 0.9.	Recall	Los valores deben estar entre 0.0 y 1.0 Un valor aceptable debe ser mayor a 0.9.	F1 Score	Los valores deben estar entre 0.0 y 1.0 Un valor aceptable debe ser mayor a 0.9.																										
de CPU																																					
Tiempo	La arquitectura con el menor valor es la más eficiente.																																				
Precisión	Los valores deben estar entre 0.0 y 1.0 Un valor aceptable debe ser mayor a 0.9.																																				
Recall	Los valores deben estar entre 0.0 y 1.0 Un valor aceptable debe ser mayor a 0.9.																																				
F1 Score	Los valores deben estar entre 0.0 y 1.0 Un valor aceptable debe ser mayor a 0.9.																																				
Resultados	<table border="1"> <thead> <tr> <th colspan="3">Eficiencia</th> </tr> <tr> <th>Parámetro</th> <th>YOLO</th> <th>Faster R-CNN</th> </tr> </thead> <tbody> <tr> <td>Uso de almacenamiento (GB)</td> <td>32</td> <td>32</td> </tr> <tr> <td>Tiempo de Procesamiento (s)</td> <td>8046.65</td> <td>3551</td> </tr> <tr> <td>Consumo de Memoria RAM (MB)</td> <td>5302.20</td> <td>7450.30</td> </tr> <tr> <td>Consumo de Memoria GPU (GB)</td> <td>13.9</td> <td>16</td> </tr> <tr> <td>Consumo de CPU (%)</td> <td>58.70</td> <td>84.75</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="3">Desempeño</th> </tr> <tr> <th>Parámetro</th> <th>YOLO</th> <th>Faster R-CNN</th> </tr> </thead> <tbody> <tr> <td>Precisión (%)</td> <td>96.72</td> <td>98.57</td> </tr> <tr> <td>Recall (%)</td> <td>77.7</td> <td>63.80</td> </tr> <tr> <td>F1 Score (%)</td> <td>86</td> <td>77.46</td> </tr> </tbody> </table>	Eficiencia			Parámetro	YOLO	Faster R-CNN	Uso de almacenamiento (GB)	32	32	Tiempo de Procesamiento (s)	8046.65	3551	Consumo de Memoria RAM (MB)	5302.20	7450.30	Consumo de Memoria GPU (GB)	13.9	16	Consumo de CPU (%)	58.70	84.75	Desempeño			Parámetro	YOLO	Faster R-CNN	Precisión (%)	96.72	98.57	Recall (%)	77.7	63.80	F1 Score (%)	86	77.46
Eficiencia																																					
Parámetro	YOLO	Faster R-CNN																																			
Uso de almacenamiento (GB)	32	32																																			
Tiempo de Procesamiento (s)	8046.65	3551																																			
Consumo de Memoria RAM (MB)	5302.20	7450.30																																			
Consumo de Memoria GPU (GB)	13.9	16																																			
Consumo de CPU (%)	58.70	84.75																																			
Desempeño																																					
Parámetro	YOLO	Faster R-CNN																																			
Precisión (%)	96.72	98.57																																			
Recall (%)	77.7	63.80																																			
F1 Score (%)	86	77.46																																			

Evidencias

Entrenamiento durante 100 épocas.
Yolo versión 8



Faster R-CNN

