

UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN MECATRÓNICA



**INFORME FINAL DEL TRABAJO DE INTEGRACIÓN CURRICULAR, MODALIDAD
DE PROYECTO DE INVESTIGACIÓN.**

TEMA:

**“SISTEMA DE SUPERVISIÓN DEL CULTIVO DE LA SPONDIAS PURPUREA
(OVO) MEDIANTE VISIÓN ARTIFICIAL.”**

**LÍNEA DE INVESTIGACIÓN: VISIÓN POR COMPUTADORA Y APLICACIONES
EN LA AGRICULTURA**

AUTOR: ALEJANDRA PATRICIA CHUQUÍN REALPE

DIRECTOR: PHD, IVÁN DANILO GARCÍA SANTILLÁN

ASESORA: PHD, BRIZEIDA NOHEMÍ GÁMEZ APARICIO, ING.

IBARRA – JULIO 2024



**UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA**

**AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD
TÉCNICA DEL NORTE**

1. IDENTIFICACIÓN DE LA OBRA

El cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DEL AUTOR			
CÉDULA DE IDENTIDAD:	1004575542		
APELLIDOS Y NOMBRES:	Chuquín Realpe Alejandra Patricia		
DIRECCIÓN	Vía Urcuquí, km 8, Los Perales 1-49		
EMAIL:	apchuquir@utn.edu.ec		
TELÉFONO FIJO:		TELÉFONO MÓVIL:	0998255560
DATOS DE LA OBRA			
TÍTULO:	Sistema de supervisión del cultivo de la Spondias purpurea (ovo) mediante visión artificial.		
AUTOR:	Chuquín Realpe Alejandra Patricia		
FECHA (DD/MM/AAAA):	22/07/2024		
PROGRAMA:	Pregrado		
TÍTULO POR EL QUE OPTA:	Ingeniero en Mecatrónica		
ASESOR/DIRECTOR:	Ing. Iván Danilo García Santillán		

2. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de tercero, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 26 días del mes de julio de 2024

EL AUTOR:

(Firma).....
Nombre: Chuquín Realpe Alejandra Patricia

CERTIFICACIÓN DIRECTOR DEL TRABAJO DE INTERGRACIÓN CURRICULAR

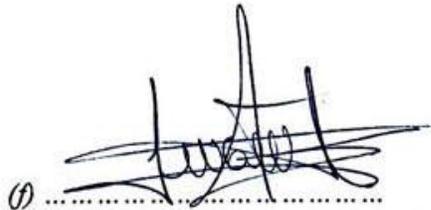
Ibarra, 26 de Julio de 2024

Ing. Iván Danilo García Santillán, PhD.

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

Haber revisado el presente informe final del trabajo de titulación, el mismo que se ajusta a las normas vigentes de la Unidad Académica de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.

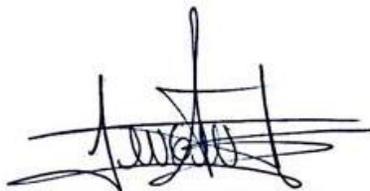


Ing. Iván Danilo García Santillán, PhD.

C.C.: 1002292603

APROBACIÓN DEL COMITÉ CALIFICADOR

El Tribunal Examinador del trabajo de titulación “SISTEMA DE SUPERVISIÓN DEL CULTIVO DE LA SPONDIAS PURPUREA (OVO) MEDIANTE VISIÓN ARTIFICIAL” elaborado por CHUQUIN REALPE ALEJANDRA PATRICIA, previo a la obtención del título de Ingeniero en Mecatrónica, aprueba el presente informe de investigación en nombre de la Universidad Técnica del Norte:



(f):.....
Ing. Iván Danilo García Santillán, PhD.
C.C.: 1002292603



(f):.....
Ing. Brizeida Nohemí Gámez Aparicio, PhD
C.C.: 1758387383

DEDICATORIA

A mi Madre

Que con su ternura y amor me impulso a entregar toda mi capacidad, que supo sembrar en mí el anhelo de la superación.

“Solo se pierde cuando dejamos de luchar”

Su hija

Alejandra

AGRADECIMIENTO

Mi primero primer agradecimiento a Dios, cuya gracia y favor inmérito me han permitido completar este objetivo.

A mi madre, quiero agradecerle por siempre creer en mí y ser no solo una madre, sino también una amiga. A mis hermanos María José y Rafael, gracias por amarme, cuidarme, y ser mis compañeros de vida siendo mis cómplices en las mayores locuras. A mis sobrinos Valentina y Alejandro, por ser ese rayo de luz en los días más nublados. A mi alma gemela Stefy, por estar siempre a mi lado y por ser una parte tan especial de mi vida. Y finalmente, a mi padre, por su cariño.

A mis amigos quiero expresar mi más sincero agradecimiento, su compañía, comprensión y ánimo han hecho que los desafíos sean más llevaderos y los éxitos aún más significativos. En especial, quiero agradecer a Santiago, quien ha estado a mi lado con una lealtad y un apoyo invaluable, desempeñando un papel crucial en este proceso. Valoro profundamente la amistad y el apoyo que me has brindado en cada paso del camino.

A R.D gracias por tu apoyo incondicional durante la última etapa de este proceso. Este logro también es fruto de tu empuje y aliento constante. Te quiero más de lo que las palabras pueden expresar.

Un grato agradecimiento a los profesores de la Universidad Técnica del Norte de la carrera de Mecatrónica, cuya dedicación y enseñanzas han sido esenciales para mi formación académica y profesional. De manera especial, agradezco a los PhD. García Santillán Iván Danilo y PhD. Gaméz Aparicio Brizeida Nohemí por su excepcional orientación y apoyo en mi trabajo de titulación. Su sabiduría, guía y paciencia en transmitir cada uno de sus conocimientos siendo fundamentales para la exitosa culminación de este logro.

Alejandra

RESUMEN

En el presente trabajo de integración curricular previo a la obtención del título de ingeniero mecatrónico se desarrolla un Sistema de supervisión del cultivo de la Spondias Purpurea (ovo) mediante visión artificial, la investigación aborda la necesidad de determinar el grado de maduración y calcular el volumen de producción de la Spondias purpurea. Se desarrolló un sistema de monitoreo avanzado utilizando YOLOv8, una técnica de aprendizaje profundo, para detectar con precisión los estados de maduración mediante el procesamiento de imágenes. Se recopilaron manualmente imágenes de Spondias purpurea en diferentes estados de maduración, y se entrenó el modelo YOLOv8 con este dataset, ajustando hiperparámetros y empleando técnicas de aumento de datos para mejorar la generalización. El modelo fue evaluado utilizando métricas de precisión, recall, F1 y mAP, obteniendo una precisión promedio de 0.98 con una confianza de 1.0, y 0.93 a una confianza menor, un puntaje F1 de 0.80 con una confianza de 0.797, y un mAP de 0.801 a un umbral de 0.5. Los resultados demuestran un rendimiento superior en la identificación de los estados de maduración, proporcionando una forma objetiva y eficiente de detección y facilitando una toma de decisiones más efectiva en el manejo de la Spondias purpurea. Para determinar el volumen de producción, se realizó un archivo Excel en donde se recolectaron los datos. Se evalúa la efectividad de la solución tecnológica empleando las métricas establecidas por la ISO 25010. Finalmente, se presentan los hallazgos del sistema de supervisión, analizando el impacto de esta nueva herramienta de detección de madurez en el ámbito de la agricultura.

Palabras clave: *Spondias purpurea, YOLOv8, visión por computadora, visión artificial, OpenCV, Python, Ultralytics.*

ABSTRACT

The research addresses the need to determine the ripening degree and calculate the production volume of *Spondias purpurea*. An advanced monitoring system was developed using YOLOv8, a deep learning technique, to accurately detect ripening stages through image processing. Images of *Spondias purpurea* in different ripening stages were manually collected, and the YOLOv8 model was trained with this dataset, adjusting hyperparameters and employing data augmentation techniques to improve generalization. The model was evaluated using precision, recall, F1, and mAP metrics, achieving an average precision of 0.98 with a confidence of 1.0, and 0.93 at a lower confidence level, an F1 score of 0.80 with a confidence of 0.797, and an mAP of 0.801 at a 0.5 threshold. The results demonstrate superior performance in identifying ripening stages, providing an objective and efficient detection method, and facilitating more effective decision-making in the management of *Spondias purpurea*. To determine the production volume, an Excel file was created to collect the data. The effectiveness of the technological solution is evaluated using the metrics established by ISO 25010. Finally, the findings of the monitoring system are presented, analyzing the impact of this new ripeness detection tool in the agricultural field.

Keywords: *artificial vision, computer vision, Python, OpenCV, YOLOv8, ultralytics, Spondias Purpurea.*

ÍNDICE GENERAL

1	<i>CAPÍTULO I: INTRODUCCIÓN</i>	15
1.1	Problema.....	15
1.1.1	Objetivos.....	16
1.1.2	Objetivo General.....	16
1.1.3	Objetivos Específicos	16
1.2	Alcance	16
1.3	Justificación	17
2	<i>CAPÍTULO II: MARCO TEÓRICO</i>	19
2.1	Antecedentes.....	19
2.2	Bases teóricas.....	20
2.2.1	Redes Neuronales Artificiales	20
2.2.2	Neuronas Artificiales	20
2.2.3	Inteligencia Artificial.....	21
2.2.4	Visión por Computadora.....	22
2.2.5	Redes Neuronales Convolucionales	23
2.2.6	Componentes, Arquitectura y Técnica de Entrenamiento	24
2.2.7	Funciones de Activación y Normalización.....	24
2.2.8	Aprendizaje Supervisado y Optimización	25
2.2.9	Herramientas y Bibliotecas para Trabajos.....	25
2.2.10	Conjunto de Datos y Desafíos en la Investigación	25
2.2.11	YOLOv8	26
2.2.12	Python Orientado a la Visión Artificial	27
2.2.13	Librería OpenCV	28

2.2.14	Librería PyQt	28
2.2.15	Librería Ultralytics.....	29
2.2.16	Spondias Purpurea	29
2.3	ISO/IEC 25010	35
3	<i>CAPÍTULO III: MARCO METODOLÓGICO</i>	37
3.1	Materiales y Métodos	37
3.1.1	Hardware.....	37
3.1.2	Software.....	38
3.2	Estructura de la Investigación.....	38
3.3	Nivel de la Investigación	38
3.4	Análisis de la Investigación	39
4	<i>CAPÍTULO IV: RESULTADOS Y DISCUSIÓN</i>	40
4.1	Recolección de Datos	40
4.2	Adquisición de las Imágenes	40
4.3	Etiquetado de Imágenes	41
4.4	Elaboración del <i>Dataset</i>	43
4.5	Despliegue y Validación del Modelo	44
4.5.1	Configuración del Entorno de Entrenamiento	44
4.5.2	Ajuste de Hiperparámetros	45
4.6	Anaconda Navigator	46
4.7	Spyder.....	47
4.8	Librerías	48
4.10	Interfaz de Usuario	51
4.10.1	QT Designer.....	51

4.10.2	Verificación y Adherencia a la Norma ISO 25010.....	52
4.11	Supervisión del Volumen de Producción	52
4.12	Métricas del Modelo YOLOv8.....	53
4.12.1	Fase de Entrenamiento.....	53
4.12.2	Fase de Validación.....	55
4.12.3	Fase de Resultados.....	56
4.12.4	Precisión	58
4.12.5	Recall	59
4.12.6	Mean Average Precision.....	61
4.12.7	F1 – Confianza.....	63
4.13	Descripción de la Programación.....	66
4.13.1	Configuración del Entorno de Trabajo	67
4.13.2	Diseño de Interfaces con Qt Designer	67
4.13.3	Implementación del Modelo YOLOv8.....	68
4.13.4	Recolección de Datos en Excel.....	69
4.14	Desarrollo de la Interfaz Gráfica de Usuario	70
4.14.1	Pruebas.....	72
4.15	Datos de Producción de Ovos: Clasificación y Resultados en Excel	75
4.16	Limitaciones del Sistema.....	78
4.17	Recomendaciones y líneas de trabajo futuro:	79
5	<i>CONCLUSIONES</i>	81
6	<i>RECOMENDACIONES</i>	82
7	<i>REFERENCIAS</i>	84
8	<i>ANEXOS</i>	90

ÍNDICE DE TABLAS

Tabla 1 <i>Ordenador para desarrollo de programación</i>	37
Tabla 2 <i>Hiperparámetros influyentes</i>	45
Tabla 3 <i>Dependencias del Script de Detección</i>	48
Tabla 4 <i>Métricas en validación</i>	66
Tabla 5 <i>Métricas en validación</i>	66

ÍNDICE DE FIGURAS

Figura 1 <i>Arquitectura de una neurona artificial</i>	21
Figura 2 <i>Ramas de la inteligencia artificial</i>	22
Figura 3 <i>Procedimiento de Detección usando Redes Neuronales Convolucionales</i>	23
Figura 4 <i>Fruto de Spondias purpurea con distintos estados de maduración</i>	30
Figura 5 <i>Estados de maduración de la Spondias Purpurea</i>	34
Figura 6 <i>Obtención de imágenes de Spondias Purpurea</i>	41
Figura 7 <i>Uso de herramienta Label Studio</i>	42
Figura 8 <i>Formato de exportación Label Studio</i>	43
Figura 9 <i>Interfaz de Anaconda Navigator</i>	47
Figura 10 <i>Instalación de Spyder en Anaconda</i>	48
Figura 11 <i>Diagrama de flujo de algoritmo principal</i>	50
Figura 12 <i>Áreas de trabajo QT Designer</i>	51
Figura 13 <i>Métricas en entrenamiento</i>	54
Figura 14 <i>Métricas en validación</i>	55
Figura 15 <i>Gráfica Precisión - Confianza</i>	58
Figura 16 <i>Gráfica Recall – Confianza</i>	60
Figura 17 <i>Resultados del modelo</i>	62
Figura 18 <i>Curva F1 - Confianza</i>	64
Figura 19 <i>Recall</i>	65
Figura 20 <i>Librerías importadas en Spyder</i>	67
Figura 21 <i>Código exportado de QT Designer</i>	68
Figura 22 <i>Código de implementación de YOLOv8</i>	69

Figura 23 <i>Código exportar a Excel</i>	70
Figura 24 <i>Portada de la interfaz gráfica</i>	71
Figura 25 <i>Interfaz principal del sistema</i>	71
Figura 26 <i>Diálogo del sistema para abrir imagen</i>	73
Figura 27 <i>Imagen procesada</i>	75
Figura 28 <i>Recolección de resultados - Excel</i>	76
Figura 29 <i>Volumen de producción</i>	76
Figura 30 <i>Cantidad de ovos analizados</i>	77
Figura 31 <i>Porcentaje de ovos analizados</i>	78

CAPÍTULO I: INTRODUCCIÓN

En esta sección se detallan los requisitos indispensables para llevar a cabo este trabajo de titulación, los cuales abarcan el tema escogido, la problemática identificada, los objetivos planteados, la extensión del estudio y su justificación, con el fin de culminar exitosamente este proyecto.

1.1 Problema

Con base en el cambio en la estructura productiva de Ecuador, es necesario reducir las importaciones y aumentar las exportaciones, lo cual implica incrementar la producción nacional de productos procesados o materias primas, como las frutas exóticas [1]. Para lograr esto, es fundamental que en Ecuador se adopten nuevas tecnologías que maximicen los beneficios. Sin embargo, muchos agricultores carecen de conocimientos técnicos y una cultura de incorporación de tecnología en la agricultura, lo que les impide comprender las ventajas que puede brindar, como la agricultura de precisión [2].

Esta innovación permite aplicar técnicas específicas adaptadas a los requerimientos de cada región, utilizando el análisis de imágenes para digitalizar y georreferenciar los cultivos, y determinar la cantidad adecuada de fertilizantes para cada punto exacto del terreno [3]. En el caso de la calidad de las frutas, especialmente la *Spondias purpurea* (ovo), es necesario realizar un riguroso control, puesto que se trata de un producto destinado al consumo humano y debe cumplir con altos estándares para garantizar la salud del consumidor.

La visión artificial por computadora es un campo en crecimiento con diversas aplicaciones, como la inspección automatizada, la identificación de objetos, las mediciones y la robótica [5]. En

este contexto, se pretende utilizar algoritmos de procesamiento de imágenes y visión artificial para identificar el estado del cultivo de *Spondias purpurea* (ovo) mediante la detección de objetos, especialmente sus frutos. Esto permitirá a los agricultores obtener mejores resultados y optimizar su tiempo en la cosecha [3].

1.1.1 Objetivos

1.1.2 Objetivo General

- Utilizar los sistemas de visión artificial y su uso en la agricultura de precisión para determinar los distintos estados de la madurez de la *Spondias purpurea*.

1.1.3 Objetivos Específicos

- Seleccionar el algoritmo de visión artificial a utilizar en el sistema de supervisión del cultivo de ovo.
- Implementar el sistema de supervisión del cultivo de ovo.
- Validar el algoritmo y sistema de visión artificial en el cultivo de ovo.

1.2 Alcance

En este trabajo de titulación se propuso crear un sistema de supervisión del cultivo de la *Spondias purpurea* (ovo) mediante visión artificial. El objetivo fue seleccionar un algoritmo que permita calcular el grado de madurez de la *Spondias purpurea* (ovo), utilizando algoritmos de visión por computadora para identificar tanto el estado de madurez como el volumen de producción [6].

1.3 Justificación

La *Spondias purpurea*, conocida comúnmente como “ovo”, es una fruta exótica con un significativo potencial económico en Ecuador. En concordancia con la modificación de la matriz productiva del país, es imperativo aumentar la producción nacional y reducir las importaciones mediante la optimización de cultivos autóctonos y su procesamiento para la exportación. Sin embargo, muchos agricultores ecuatorianos enfrentan desafíos técnicos debido a la falta de conocimientos en tecnologías avanzadas como la agricultura de precisión. Esta brecha tecnológica limita su capacidad para maximizar la productividad y calidad de sus cultivos.

Desde este marco, la adopción de un sistema de monitoreo enfocado en visión artificial ofrece una solución innovadora y eficiente. La visión artificial, una disciplina dentro de la inteligencia artificial (IA), permite la automatización de procesos de monitoreo a través del análisis de imágenes y videos. Este enfoque puede proporcionar una supervisión continua y precisa, mejorando tanto optimización de los recursos como la toma de decisiones. Además, permite un control riguroso y constante del estado de madurez y calidad de los frutos, garantizando que los productos cumplan con los altos estándares necesarios para el consumo humano y para la exportación [7].

La adopción de sistemas de visión artificial en la agricultura de precisión representa una solución innovadora y eficiente para abordar estos desafíos. Al aplicar esta tecnología al cultivo de *Spondias purpurea*, se obtienen diversos beneficios, como una optimización de recursos, una mejora en la calidad del producto, una reducción de la dependencia en métodos manuales y un incremento de la productividad. La automatización de la supervisión del cultivo reduce la carga de trabajo manual y minimiza los errores humanos, aumentando la consistencia y fiabilidad de los procesos agrícolas.

Por esta razón, el desarrollo e implementación de un sistema de supervisión del cultivo de *Spondias purpurea* mediante visión artificial no solo contribuirá a mejorar la eficiencia y calidad de la producción de esta fruta, sino que también fomentará la adopción de tecnologías avanzadas en el sector agrícola ecuatoriano, alineándose con los objetivos nacionales de transformación productiva y desarrollo sostenible. Esta innovación permitirá a los agricultores obtener mejores resultados, optimizar su tiempo durante el periodo de cosecha y contribuir al desarrollo económico del país al incrementar la competitividad de sus productos en el mercado global.

CAPÍTULO II: MARCO TEÓRICO

Esta investigación tuvo como objetivo crear un sistema de monitoreo para el cultivo de la *Spondias purpurea* (ovo), utilizando técnicas de visión artificial y su implementación en una aplicación específica. Para garantizar la comprensión precisa de este proyecto, es crucial que los términos y conceptos relacionados se expliquen de manera clara y concisa, como se detalló en el problema planteado.

2.1 Antecedentes

En Ecuador se cultivan tres tipos de ovo: *Spondias cytherea* L., *Spondias purpurea* L. y *Spondias mombin* L., siendo la segunda cultivada principalmente en Ambuquí, Imbabura. La producción, que varía entre 20 000 y 30 000 kg por hectárea, es limitada debido a la delicadeza del fruto y su difícil comercialización fuera de la zona. Los agricultores locales, dedicados mayormente al cultivo de *Spondias Purpurea* necesitan conocer los estados de madurez del fruto para optimizar su comercialización y producción de subproductos. Sin embargo, la falta de conocimiento técnico impide la adopción de tecnologías avanzadas, como la agricultura de precisión, que podrían maximizar los beneficios.

Investigaciones anteriores, como el estudio de Valdivia [4] Diseño de un sistema de visión artificial para la clasificación de chirimoyas basado en medidas, desarrollaron un sistema encargado de medir *Annona Cherimola* Mill (Chirimoya) del ecotipo Aurora, originarias de la comunidad de Callahuanca, con el objetivo de clasificar aquellas que contaran con dimensiones de 10 x 12 cm. Esto se lleva a cabo durante la fase de procesamiento a través del ajuste de contraste, y en la fase de segmentación, que abarca técnicas de apertura, cierre y código cadena; los métodos seleccionados satisfacen los tiempos de cálculo requerido para el diseño.

Por su parte, González y Parra [9] elaboraron una investigación de Diseño e implementación de un sistema de reconocimiento de naranjas para el robot gio 1 usando visión asistida por computador, en la que crearon una plataforma robótica creada en la Universidad Católica de Colombia. La finalidad fue desarrollar un sistema de recolección de frutas, con el fin de asistir a los trabajadores del campo que realizan tareas manuales.

En esta misma línea, Aguilar [10], en su trabajo Diseño de un sistema clasificador de manzanas por color mediante visión artificial para la empresa Fresh & Natural C.I., desarrolló un sistema de clasificación de manzanas utilizando visión artificial y capturando imágenes con una cámara web, con el propósito de automatizar un proceso manual realizado en las empresas. El diseño del sistema se llevó a cabo empleando la metodología del modelo en V, especificando los requisitos que debía cumplir el sistema.

2.2 Bases teóricas

2.2.1 Redes Neuronales Artificiales

Las redes neuronales artificiales están fundamentadas en algoritmos informáticos que integran la IA, el aprendizaje automático y el aprendizaje profundo para analizar datos y encontrar soluciones a problemas complejos, identificando patrones y relaciones ocultas en la información.

2.2.2 Neuronas Artificiales

Una neurona artificial es como un pequeño procesador cerebral, diseñado para imitar el funcionamiento de las neuronas humanas. Cada una de estas unidades de procesamiento es como un pequeño “centro de decisión” que recibe y procesa información de manera análoga a como lo hacen las neuronas biológicas en nuestro cerebro. Al conectar estas neuronas artificiales en redes

más grandes, es posible crear sistemas que pueden aprender y adaptarse a nuevos datos, de manera similar a cómo el cerebro aprende y se adapta a nuevas experiencias (ver Figura 1). Además, una función específica calcula la salida de la neurona, que a menudo está determinada por un acceso [11].

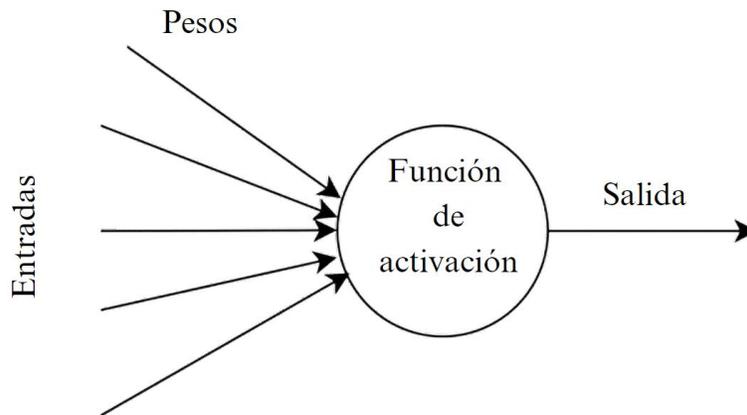


Figura 1 *Arquitectura de una neurona artificial.* Adaptado de Gershenson [12].

2.2.3 *Inteligencia Artificial*

La IA es una rama científica que tiene como propósito crear sistemas y algoritmos que puedan emular la cognición humana, abarcando aspectos como el aprendizaje, el razonamiento, la percepción y la interacción en lenguaje natural [13].

La IA se compone de diversas subáreas, incluyendo el aprendizaje profundo (*deep learning*), el aprendizaje automático (*machine learning*), la visión por computadora, la robótica y el procesamiento del lenguaje natural (PNL), entre otras. En años recientes, la IA ha avanzado considerablemente, gracias a la creación de algoritmos más avanzados [14].

La disponibilidad de vastos volúmenes de datos y el incremento en la capacidad de procesamiento (ver Figura 2).

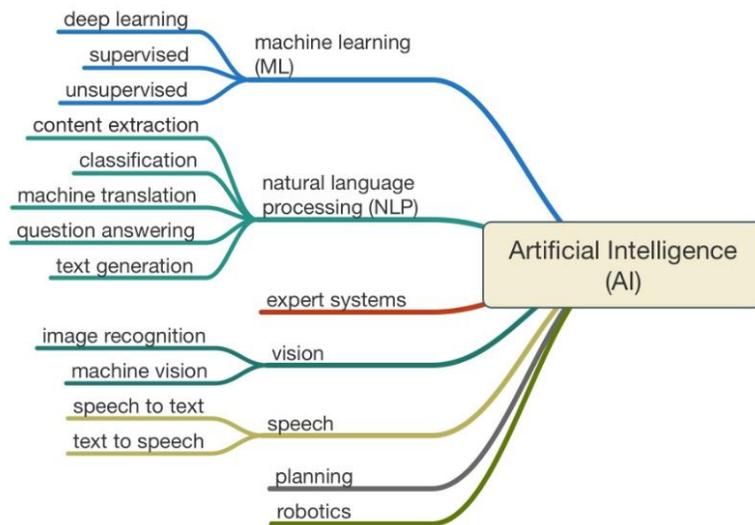


Figura 2 *Ramas de la inteligencia artificial.* Tomado de Stanford University [15].

2.2.4 *Visión por Computadora*

La visión por computadora es un campo dentro de la IA, que se centra en dotar a las máquinas para que interpreten y analicen datos visuales, como imágenes y videos. Se utiliza para detectar objetos, clasificar imágenes y reconocer rostros, entre otras aplicaciones. Para este propósito, se emplean diversos algoritmos y técnicas, incluyendo el procesamiento de imágenes, la obtención de características, el reconocimiento de objetos y el aprendizaje automático.

Entre las técnicas empleadas se encuentran la transformación de imágenes para facilitar su análisis, la identificación de características importantes y la detección y clasificación de objetos mediante redes neuronales convolucionales (CNN) y otros métodos avanzados. La detección de objetos, en particular, se encarga de identificar y clasificar elementos presentes en imágenes o videos. Para lograr esto, se emplean técnicas avanzadas como las CNN, las redes neuronales convolucionales basadas en regiones (R-CNN) y el método conocido como *You Only Look Once* (YOLO) [17, 18].

2.2.5 Redes Neuronales Convolucionales

Las CNN permiten son adecuadas para analizar datos organizados en estructuras de cuadrícula, como imágenes. Son ideales para tareas de visión por computadora, como clasificar imágenes, detectar objetos y segmentar semánticamente; esto es, en visión artificial. En la Figura 3 se muestra el procedimiento llevado a cabo por este tipo de red neuronal.

Es importante mencionar que las CNN diseñadas para aprender patrones jerárquicos de características, desde las básicas hasta las avanzadas, lo que las hace adecuadas para diversas aplicaciones, como la organización de imágenes, la identificación de objetos y el PNL. La capacidad de las CNN para aprender automáticamente características relevantes ha dado lugar a un rendimiento excepcional en numerosas tareas y ha sido crucial para el progreso en el ámbito de la IA [19].

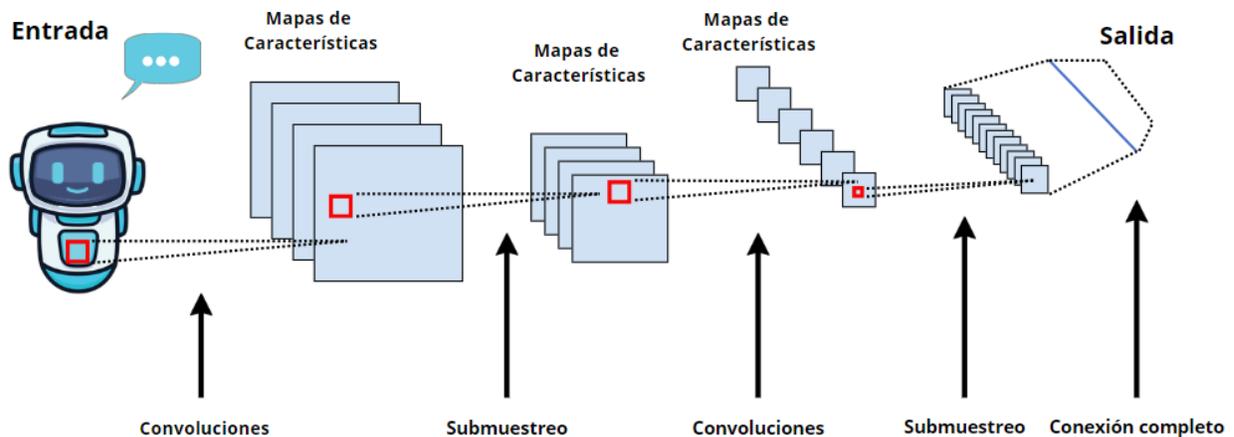


Figura 3 Procedimiento de Detección usando Redes Neuronales Convolucionales. Tomado de Mohamed [20]

2.2.6 Componentes, Arquitectura y Técnica de Entrenamiento

Las redes neuronales artificiales están diseñadas para replicar la manera en que funciona el cerebro humano, con el fin de desarrollar la capacidad de tomar decisiones inteligentes utilizando recursos computacionales. Estas redes están compuestas por componentes clave, como una estructura definida, un mecanismo de aprendizaje efectivo, la capacidad de asociar información y representarla de manera significativa. Al otorgar parámetros específicos a estas redes durante el entrenamiento, se puede desarrollar un modelo matemático que, a su vez, se puede implementar en un sistema informático para aplicaciones de *machine learning* [8].

2.2.7 Funciones de Activación y Normalización

Las funciones de activación son esenciales en las CNN, puesto que incorporan no linealidades en el modelo, lo que permite a la red aprender representaciones más complejas y avanzadas de los datos. Entre las funciones de activación más comunes se encuentran ReLU (*rectified linear unit*), Leaky ReLU, ELU (*exponential linear unit*) y Swish. Estas funciones contribuyen a prevenir dificultades como el desvanecimiento y la explosión del gradiente, promoviendo el entrenamiento de modelos más profundos y eficaces[11].

Otra técnica crucial en las CNN es la normalización, cuyo propósito no es otro que estabilizar y acelerar el proceso de entrenamiento. La normalización por lotes es una técnica habitual que ajusta la entrada de cada capa de acuerdo con la dispersión del lote actual y la medida. Este método no solo optimiza la rapidez de convergencia y la estabilidad del entrenamiento, sino que también permite emplear tasas de aprendizaje mucho más elevadas y disminuye la dependencia de la configuración inicial de los pesos [11].

2.2.8 Aprendizaje Supervisado y Optimización

El aprendizaje supervisado en CNN utiliza datos etiquetados para entrenar y configurar los parámetros del modelo. Por otro lado, el aprendizaje no supervisado, que no necesita etiquetas, investiga estructuras y patrones en los datos con técnicas como *autoencoders* y aprendizaje contrastivo. Estos dos enfoques pueden integrarse para mejorar la eficiencia y el rendimiento de las CNN. Asimismo, la configuración inicial de los pesos, la regularización y los algoritmos de optimización, como el descenso de gradiente estocástico y Adam, son cruciales para acelerar la convergencia, prevenir el sobreajuste y optimizar el entrenamiento del modelo[11].

2.2.9 Herramientas y Bibliotecas para Trabajos

Para trabajar con CNN, se utilizan diversas herramientas y bibliotecas especializadas. *Open Source Computer Vision Library* (OpenCV) es fundamental para el procesamiento y análisis de imágenes, proporcionando una amplia gama de funciones para la manipulación y detección de características visuales. NumPy y Pandas son esenciales para la manipulación y preprocesamiento de datos, proporcionando soporte para arrays multidimensionales y estructuras de datos flexibles. Ultralytics YOLO facilita la detección de objetos con modelos preentrenados y optimización para detección en tiempo real, permitiendo la construcción, entrenamiento y despliegue de modelos eficientes[46].

2.2.10 Conjunto de Datos y Desafíos en la Investigación

En la investigación de CNN con YOLOv8 en el ámbito de detección de objetos, se utilizan Python y las bibliotecas OpenCV, NumPy y YOLOv8 para enfrentar desafíos como el preprocesamiento eficiente de imágenes, el ajuste de hiperparámetros y la optimización de

modelos para detección en tiempo real. Conjuntos de datos como COCO, PASCAL VOC y Open Images son cruciales para entrenar modelos robustos, mientras que instrumentos como OpenCV y NumPy garantizan el procesamiento y la manipulación de imágenes. YOLOv8 permite la construcción y entrenamiento de modelos de alta precisión. La integración de estas tecnologías permite abordar desafíos técnicos y computacionales, mejorando la eficiencia y precisión de los sistemas de detección de objetos [46].

2.2.11 YOLOv8

YOLO es una arquitectura de detección de objetos que se distingue por su eficiencia y precisión en la identificación de objetos en imágenes y videos. Su rapidez la hace ideal para aplicaciones en tiempo real en seguridad, robótica y visión por computadora. Se basa en una CNN que predice cajas delimitadoras y la probabilidad de la clase de objeto en esas cajas. YOLO fracciona la imagen en una cuadrícula y realiza la identificación del objeto de manera única, lo que lo hace más rápido que otras técnicas. Además, puede detectar múltiples objetos en una imagen y proporcionar la probabilidad de acierto de la clase de objeto detectado[46].

2.2.11.1 Arquitectura de YOLOv8

La arquitectura de YOLOv8 optimiza la eficiencia y precisión mediante una combinación de tres componentes principales: el *backbone*, el *neck* y el *head*. El *backbone* utiliza CNN avanzadas para extraer características jerárquicas, mientras que el *neck* fusiona características de diferentes escalas utilizando FPN y PAN. El *head* emplea un diseño sin anclas para realizar las predicciones finales de cuadros delimitadores, clases y puntuaciones de confianza, lo que optimiza la precisión al detectar objetos de diversos tamaños y formas [46].

2.2.11.2 Detección sin Anclas en YOLOv8

YOLOv8 adopta una detección sin anclas, eliminando la necesidad de definir anclas predefinidas de diferentes tamaños y relaciones de aspecto. Este enfoque simplifica el diseño del modelo y mejora la precisión al predecir cuadros delimitadores basados en puntos clave en lugar de anclas. La detección sin anclas permite una mejor adaptabilidad a objetos de diversas escalas y formas, reduciendo la tasa de falsos positivos y negativos y mejorando la capacidad del modelo para detectar objetos pequeños e irregulares [46].

2.2.11.3 Función de Pérdida y Reglas de Actualización

La medida de error de YOLOv8 combina varias componentes clave: pérdida de localización, pérdida de clasificación y pérdida de confianza. La pérdida total se calcula como una combinación ponderada de estas pérdidas, utilizando métricas como IoU y entropía cruzada para evaluar la precisión de las predicciones. El modelo se entrena utilizando optimizadores avanzados como Adam o SGD con momento, y se aplican técnicas de regularización como *dropout* y *batch normalization* para prevenir el sobreajuste, asegurando una actualización eficiente y precisa de los parámetros del modelo [43].

2.2.12 Python Orientado a la Visión Artificial

Python es un lenguaje de programación altamente utilizado en el procesamiento de imágenes debido a su facilidad de uso y las numerosas bibliotecas especializadas disponibles, como OpenCV, NumPy y scikit-image. Estas bibliotecas proporcionan una variedad de funciones y algoritmos para cargar, manipular y examinar imágenes [23]. Con Python, los desarrolladores pueden llevar a cabo operaciones básicas como ajuste de brillo, cambio de tamaño y recorte, así

como tareas más complejas como detección de bordes, segmentación y reconocimiento de patrones. Python también se integra bien con otras tecnologías y es adecuado para aplicaciones más grandes que requieran procesamiento de imágenes eficiente [24].

2.2.13 Librería OpenCV

OpenCV es una biblioteca de código abierto comúnmente utilizada en el procesamiento de imágenes y visión por computadora [25]. En Python, la biblioteca principal es cv2, que proporciona una extensa variedad de funciones para cargar, manipular y analizar imágenes. OpenCV se combina comúnmente con numpy para cálculos numéricos eficientes y matplotlib para visualización de imágenes. Con características como ajuste de brillo, detección de bordes y reconocimiento de patrones, OpenCV se utiliza en aplicaciones como reconocimiento facial, seguimiento de objetos y robótica. En resumen, OpenCV es una biblioteca esencial en Python para el procesamiento de imágenes y la visión por computadora [24].

2.2.14 Librería PyQt

PyQt es un módulo GUI que une el marco multiplataforma Qt C++ con Python. Qt no solo ofrece herramientas GUI, sino también abstracciones para subprocesos de red, Unicode, SQL, bases de datos, SVG, OpenGL, XML, un navegador web, un sistema de servicios y una amplia gama de widgets GUI. La clase “Qt” funciona con un mecanismo de ranuras que facilita la comunicación entre componentes de *software*, lo que permite diseñar elementos reutilizables. Además, Qt incluye Qt Designer, una herramienta que funciona como una interfaz gráfica de usuario. Con PyQt, es posible diseñar código Python desde Qt Designer y agregar nuevos controles GUI utilizando tanto Qt Designer como Python como lenguaje de programación [26].

2.2.15 Librería Ultralytics

Ultralytics es una destacada librería de *software* especializada en la identificación de objetos y otras labores de visión por computadora, utilizando la arquitectura YOLO. Su facilidad de uso la hace accesible para desarrolladores sin experiencia previa. Destaca por su rendimiento excepcional, ideal para aplicaciones que requieren detección en tiempo real. Puede integrarse en proyectos académicos e industriales en el ámbito de la visión por computadora. Enfocada en optimización y precisión, es una herramienta eficiente para detectar objetos con rapidez y precisión, siendo ideal para aplicaciones en seguridad, robótica y visión por computadora [27].

2.2.16 Spondias Purpurea

La *Spondias purpurea*, también conocida como ovo o ciruela jocote, es un árbol frutal nativo de América Central y Ecuador. Sus frutos son drupas comestibles de sabor agridulce. Como se mencionó anteriormente, se propuso desarrollar un método de supervisión utilizando visión artificial para mejorar el cultivo de la *Spondias purpurea* y asegurar su crecimiento adecuado [1]. En la Figura 4 se observa el fruto de la *Spondias Purpurea* con distintos estados de maduración



Spondias purpurea con distintos estados de maduración



Clase 1 (verde)



Clase 2 (amarillo)



Clase 3 (rojo)

Figura 4 Fruto de *Spondias purpurea* con distintos estados de maduración

2.2.16.1 Características de la *Spondias Purpurea* (Ovo)

La *Spondias purpurea* es un árbol frutal perenne que puede llegar a medir hasta 20 metros. Sus hojas son alternas, compuestas y pinnadas, con folíolos ovalados y bordes dentados. Produce inflorescencias terminales con pequeñas flores verdosas. Sus frutos son drupas ovaladas o redondas, de tamaño variable y colores que van desde amarillo hasta rojo al madurar. Posee un sabor agridulce y jugoso. La *Spondias purpurea* es apreciada por su valor alimenticio y propiedades medicinales debido a su contenido de antioxidantes, vitaminas y minerales, que la hacen nutricionalmente beneficiosa para la salud [28].

2.2.16.2 Origen y Distribución

La *Spondias purpurea*, conocida como ciruela ovo o ciruela china, es una especie originaria de América Central, pero también se encuentra en varias regiones de Ecuador. En este país, se distribuye principalmente en la zona costera y en la región amazónica. Se cultiva tanto a nivel comercial como en huertos familiares debido a la demanda de sus frutos. El clima cálido y húmedo de Ecuador brinda condiciones favorables para el crecimiento de la *Spondias purpurea*, y es común encontrarla en provincias como Esmeraldas, Manabí, Guayas, Los Ríos, Imbabura, El Oro, Sucumbíos y Napo, entre otras [3].

Al norte de la provincia de Imbabura se encuentra la parroquia de Ambuquí, que tiene una temperatura de 21 °C, una humedad relativa del 80 % y un clima cálido y seco. Esta parroquia, que cuenta con alrededor de seis mil habitantes, es conocida en el ámbito turístico por su producción de ovo [3].

2.2.16.3 Descripción Taxonómica

A continuación, se presenta la descripción taxonómica de la *Spondias purpurea*:

Reino: Plantae (Plantas).

División: Magnoliophyta (Plantas con flores).

Clase: Magnoliopsida (Dicotiledóneas).

Orden: Sapindales (Sapindales).

Familia: Anacardiaceae (Anacardiáceas).

Género: *Spondias*.

Especie: *Spondias purpurea*.

La *Spondias purpurea* es un árbol frutal perenne de tamaño intermedio a grande, con un tronco erguido y ramificado y una corteza lisa de tonalidad grisácea. Sus hojas son alternas, compuestas y pinnadas, con folíolos ovales y margen dentado. Las flores son diminutas y verdosas, y se integran en inflorescencias terminales. Los frutos son drupas de forma ovalada o redonda, de colores que varían entre amarillo, naranja y rojo al madurar. Cada fruto contiene una única semilla en su interior. La *Spondias purpurea* se caracteriza por su sabor agridulce y jugoso, siendo apreciada por su valor alimenticio y propiedades medicinales [29].

2.2.16.4 Fruto

- Tamaño: el tamaño del fruto tiende a variar, pero generalmente tiene una longitud de 2 a 4 centímetros y un diámetro similar.
- Color: el color del fruto varía según el grado de madurez. Inicialmente, es verde, y a medida que madura, puede cambiar a tonalidades amarillas, anaranjadas o rojizas,

siendo más intenso en algunos individuos. El color rojizo o púrpura es común en los frutos maduros.

- Textura: la piel del fruto es delgada, pero resistente. Puede presentar una superficie lisa y brillante.
- Pulpa: la pulpa del fruto es jugosa y carnosa, con una consistencia suave. Tiene un sabor característico agrídulce que varía en intensidad según la madurez.
- Semilla: en el centro del fruto se encuentra una única semilla grande y dura. La semilla es de forma ovalada y suele ser de color marrón oscuro [28].

2.2.16.5 Maduración de la Futa

El proceso mediante el cual una fruta alcanza su máximo desarrollo en términos de sabor, textura y color se conoce como maduración. Este proceso involucra cambios químicos y físicos, como el aumento de azúcares, la disminución de ácidos y la transformación de la estructura celular, que hacen que la fruta sea más dulce, jugosa y atractiva para el consumo. Dependiendo del tipo de fruta, la maduración puede continuar después de la cosecha (frutas climatéricas como plátanos y tomates) o detenerse una vez cosechada (frutas no climatéricas como uvas y cítricos). La correcta gestión de la maduración es crucial para asegurar la calidad y el disfrute óptimo de la fruta [4].

2.2.16.6 Tiempo de Maduración de la *Spondias Purpurea*

El tiempo de maduración del fruto de la *Spondias purpurea* en la parroquia de Ambuquí, donde la temperatura promedio es de 21 °C, con una humedad relativa del 80 % y un clima cálido-seco, suele ser de 3 a 4 meses desde la floración hasta que la fruta alcanza su madurez en el árbol. Para que el fruto pase de estar “jecho” (tonos verdes amarillentos) a adquirir tonos amarillos, toma aproximadamente 2 días, y para llegar a estar completamente maduro y listo para cosechar,

requiere 4 días más. Durante este periodo, la fruta experimenta cambios en su tamaño, color y composición química, volviéndose más dulce y desarrollando su característico sabor y aroma. El principal indicador de maduración de esta fruta es el color de la cáscara (ver Figura 5)[4].

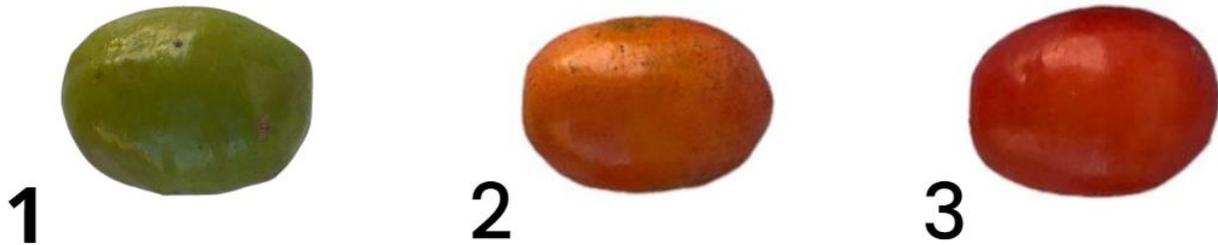


Figura 5 Estados de maduración de la *Spondias Purpurea*. Se presentan los distintos estados de madurez de la *Spondias purpurea*. Clase 1 (verde), clase 2 (amarillo), clase 3 (rojo).

2.2.16.7 Clasificación y Supervisión del Volumen de Producción

La supervisión del volumen de producción de ovos es un proceso crucial que permite evaluar y gestionar la cantidad de producto disponible en diferentes etapas de maduración. Mediante la aplicación de sistemas de detección enfocados en el análisis de imágenes, es posible clasificar los ovos en categorías como rojos, verdes y amarillos, reflejando así su estado de desarrollo. Esta clasificación precisa permite ajustar las estrategias de recolección y distribución de manera eficiente, asegurando que los ovos se cosechen y distribuyan en el momento óptimo. Al llevar un control riguroso del volumen de producción, se mejora la toma de decisiones en la gestión de recursos, se optimizan los procesos de producción y se garantiza una alta calidad del producto final [30].

2.3 ISO/IEC 25010

Para que un *software* sea valioso y cumpla con los requisitos de los usuarios, es necesario que tenga un nivel de calidad que englobe los siguientes requisitos [48]:

1. El *software* cumple con completitud funcional, lo que significa que todas las funcionalidades cubren los objetivos propuestos y entregan los resultados deseados con precisión. Además, es pertinente en términos de tareas y funciones propuestas.
2. En cuanto al desempeño, el *software* cumple con el comportamiento temporal establecido y los recursos empleados al interior del *software* cumplen su función, respetando los límites máximos de los parámetros establecidos.
3. En términos de compatibilidad, el *software* puede convivir con otros programas sin dañarlos y puede intercambiar información con otros sistemas sin problemas.
4. La usabilidad del *software* se refiere a su capacidad para satisfacer las necesidades del usuario. Este puede aprender a aplicar, utilizar y controlar el *software* con facilidad, y cuenta con una interfaz amigable y accesible, incluso para usuarios con discapacidades.
5. El *software* es fiable en condiciones normales, se recupera de fallos y mantiene la seguridad mediante protección contra accesos no autorizados y capacidad de registrar acciones y demostrar identidades.
6. Con respecto a la mantenibilidad, el *software* permite cambios con un impacto reducido en el sistema en su conjunto y puede utilizarse en varios sistemas sin problemas futuros. Permite evaluar el impacto de los cambios, diagnosticar fallos y determinar las pruebas necesarias.

7. Finalmente, el *software* es portátil y puede adaptarse a distintos entornos de *hardware* y *software*, pudiendo ser instalado y utilizado como reemplazo de otros productos con el mismo objetivo.

CAPÍTULO III: MARCO METODOLÓGICO

En el capítulo del marco metodológico se describe el proceso de programación y desarrollo del sistema diseñado para detectar la madurez de la *Spondias purpurea*. Para este sistema, se empleó el lenguaje de programación Python, un lenguaje de alto nivel que no solo garantiza el proceso de aprendizaje, sino que también ofrece numerosas librerías especializadas en visión por computadora. En particular, se empleó YOLOv8 para la detección de imágenes, aprovechando su eficacia y precisión en la identificación de objetos.

3.1 Materiales y Métodos

En este apartado, se evalúan los materiales empleados en la investigación, así como los procedimientos, enfoques, diseños y métodos aplicados durante el estudio.

3.1.1 Hardware

En la Tabla 1 se describe el *hardware* empleado en este trabajo de grado.

Tabla 1

Ordenador para desarrollo de programación

Modelo	Lenovo Legion 5 15IMH05
Procesador	Intel (R) Core (TM) i7-10750H CPU 2.50 GHz, 2592 Mhz
Memoria instalada (RAM)	8.00 GB.
Tipo de sistema	Sistema operativo de 64 bits – Windows 11.

3.1.2 Software

- Sistema operativo de 64 bits – Windows 11.
- Anaconda.
- Spyder.
- Python.
- YOLOv8.

3.2 Estructura de la Investigación

Considerando que se busca explorar y explicar las características de un nuevo proceso de estudio, el enfoque de este estudio es experimental y descriptivo. Su objetivo principal es optimizar los materiales, los productos y la tecnología involucrados. La experimentación es fundamental en este contexto, dado que es el único enfoque que permite establecer relaciones causales entre las diferentes variables analizadas.

3.3 Nivel de la Investigación

El grado de investigación empírica hace referencia al nivel de profundidad de conocimiento que se busca obtener a través de la investigación. En este caso, el nivel de investigación es exploratorio, lo que quiere decir que el propósito es visualizar rutas innovadoras en el desarrollo intelectual humano. La meta de esta investigación es diseñar un sistema que proporciona un enfoque innovador y una alternativa efectiva para la producción en el cultivo de ovo, utilizando un procedimiento de análisis de imágenes [31]. De esta manera, se busca allanar el camino para futuras soluciones y avances en este campo.

3.4 Análisis de la Investigación

Etapa 1: Selección del algoritmo: implicó realizar investigaciones informativas y consultar diversas fuentes para fundamentar el objeto de estudio y examinar los posibles algoritmos que se empleará en el proceso.

Etapa 2: desarrollo del algoritmo: se inició el desarrollo de algoritmos utilizando el lenguaje de programación Python y las librerías OpenCV, 'numpy' y YOLOv8 para la detección de imágenes.

Etapa 3: Implementación del sistema: se centró en crear una interfaz de usuario amigable que pudiera mostrar los resultados de manera comprensible.

Etapa 4: Validación del algoritmo: se llevó a cabo la validación de la investigación mediante la recolección de muestras para realizar pruebas de funcionamiento, seguida de un análisis de los resultados y la revisión y corrección del desempeño del sistema.

CAPÍTULO IV: RESULTADOS Y DISCUSIÓN

Este capítulo proporciona una descripción detallada del curso de acción propuesto para abordar y resolver el problema actual, con el fin de comprobar el adecuado funcionamiento del algoritmo y de todos los componentes de la interfaz gráfica de usuario.

4.1 Recolección de Datos

La recolección de datos abarca los métodos y herramientas empleados para recopilar información que valide las hipótesis.

Técnicas utilizadas:

- Revisión de directorios y bases de datos:

Esta técnica facilitó la recopilación y sistematización de la información requerida, puesto que en internet se encuentran disponibles foros, blogs, artículos, tesis, tutoriales en video, libros, entre otras fuentes de información.

- Observación:

Esta técnica es fundamental para recopilar datos esenciales, dado que facilita la interacción con la realidad a través de observaciones, como capturas de pantalla, el desarrollo de algoritmos, registros de simulación y fotografías.

4.2 Adquisición de las Imágenes

Un paso esencial para conseguir las muestras requeridas para entrenar el modelo de clasificación fue la obtención de imágenes de la *Spondias purpurea* en sus distintas etapas de maduración (ver Figura 6). Este fruto presenta variaciones en su color según su grado de madurez

[32], lo que puede manifestarse como un cambio en la tonalidad o en la superficie del fruto. Las imágenes descritas en los apartados siguientes se obtuvieron de conjuntos de datos preexistentes.



Distintos estados de maduración



Clase 1 (verde)



Clase 2 (amarillo)



Clase 3 (rojo)

Figura 6 *Obtención de imágenes de Spondias Purpurea*

4.3 Etiquetado de Imágenes

Se realiza el etiquetado de acuerdo con el tipo de maduración del ovo, en tres clases: clase 1: verde clase 2: amarillo, y clase 3: rojo (ver figura 7). Este procedimiento de etiquetado resultó fundamental para el entrenamiento del modelo de CNN, dado que proporcionó a la red la información necesaria para aprender a diferenciar los estados de maduración del ovo.

Para el proceso de etiquetado, se eligió la herramienta Label Studio por su flexibilidad y capacidad de adaptación. Label Studio es un instrumento de etiquetado de datos de código abierto que permite a los usuarios etiquetar datos de manera eficiente para entrenar modelos de aprendizaje automático. Es versátil y admite una amplia gama variedad tipos de datos y tareas de etiquetado, mientras que su naturaleza en línea facilita la colaboración entre equipos y es compatible con diversos sistemas y formatos de archivo.

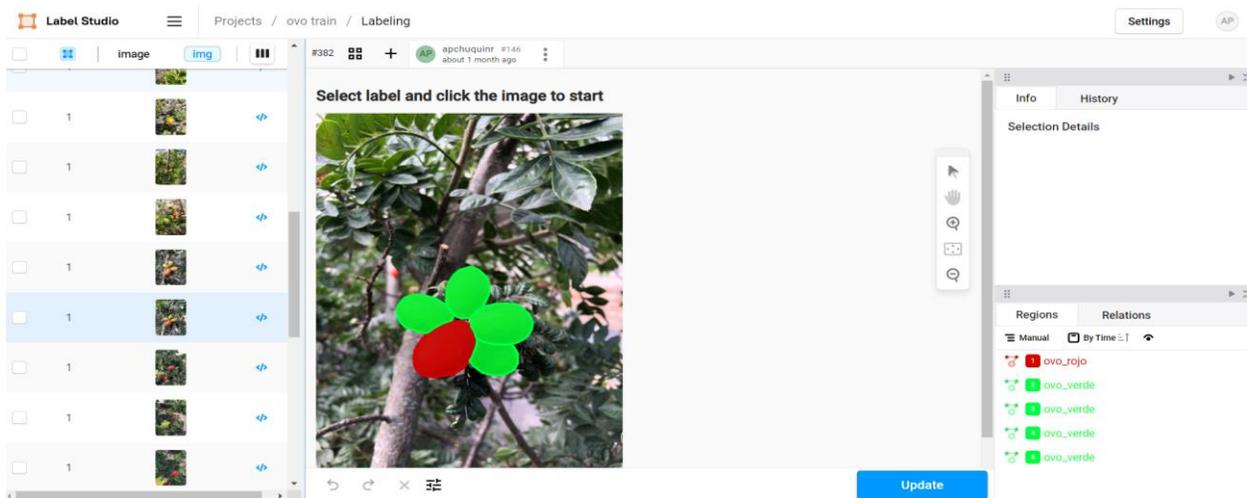


Figura 7 *Uso de herramienta Label Studio.* Se muestra el uso de etiquetas por polígonos de Label Studio.

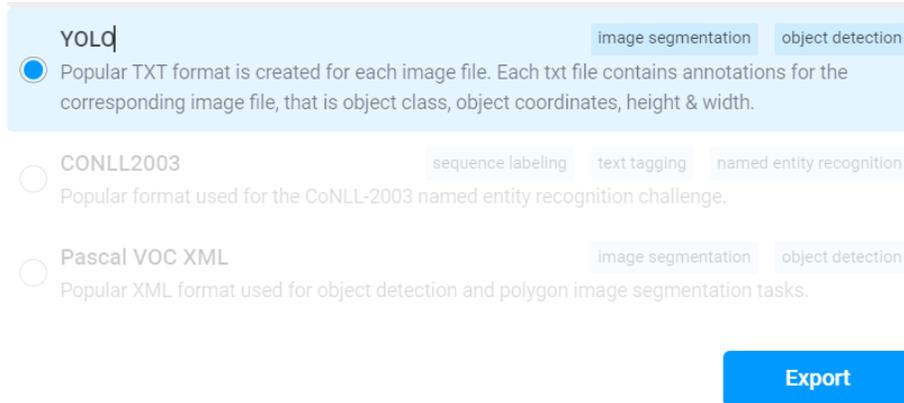


Figura 8 *Formato de exportación Label Studio*

Con respecto a cómo se almacenan los archivos de etiquetas (ver Figura 8), YOLO utiliza un formato de archivo de texto para cada imagen con anotaciones en el siguiente formato: **class_id x_center y_center width height**. En este esquema, **class_id** representa el identificador de la clase del objeto, mientras que **x_center** y **y_center** son las coordenadas del centro del *bounding box* en relación con el tamaño de la imagen (valores normalizados entre 0 y 1), y **width** y **height** indican las dimensiones del *bounding box* (también normalizados).

4.4 Elaboración del *Dataset*

La creación de un conjunto de datos es un proceso fundamental para el entrenamiento y evaluación de modelos de aprendizaje profundo. Es un proceso que requiere tiempo para generar conjuntos de datos suficientemente grandes que permitan entrenar un modelo y garantizar su efectividad [33].

Al completar el proceso de anotación del conjunto de datos que consta de 127 imágenes de entrenamiento y 23 imágenes de validación, cada una con una resolución de 640x640 píxeles, se generaron las estadísticas del *dataset*, lo que permitió obtener una visión general del total de imágenes, etiquetas y número de clases etiquetadas [34].

4.5 Despliegue y Validación del Modelo

En cuanto a la implementación del modelo, se utilizó la plataforma de Jupyter Notebook para el entrenamiento de imágenes. Proporciona recursos computacionales avanzados y gratuitos, como GPU, los cuales son esenciales para el entrenamiento. Esta plataforma es eficaz para desarrollar, entrenar y evaluar modelos de detección de objetos de manera efectiva, aprovechando la interactividad y flexibilidad. El conjunto de datos de imágenes y sus etiquetas, el modelo entrenado, los resultados obtenidos, las imágenes de prueba y las métricas arrojadas por YOLOv8 se emplean para evaluar y mejorar la precisión y eficiencia de los sistemas de detección de objetos.

El conjunto de datos se dividirá en 80 % para entrenamiento y 20 % para validación. Se buscará la configuración óptima de hiperparámetros (épocas, tamaño de redimensionamiento y lote) durante el entrenamiento. Luego, se probará el modelo con imágenes fuera del conjunto de entrenamiento para evaluar su eficacia y generalización en nuevas imágenes de la maduración de la *Spondias purpurea*.

4.5.1 Configuración del Entorno de Entrenamiento

Configurar un entorno de entrenamiento adecuado es crucial para el desarrollo y la eficacia de todo modelo de aprendizaje automático. En Jupyter Notebook, se organiza una estructura de directorios específica que comprende carpetas diferenciadas para las imágenes de entrenamiento y validación, así como para los archivos de texto con las etiquetas correspondientes. Además, se crea una carpeta destinada al almacenamiento de las imágenes destinadas a pruebas posteriores.

Una vez configurado el entorno de trabajo y la estructura de directorios, se procede a instalar la biblioteca Ultralytics en el entorno local de Jupyter. Esta biblioteca es esencial para ofrecer la implementación de YOLOv8 requerida para el entrenamiento del modelo.

Tras instalar la biblioteca Ultralytics, se transfieren las imágenes del conjunto de datos extraído que se encuentran en el entorno local de ejecución de Jupyter [35]. Los datos se dividen en dos partes: el 80 % de las imágenes se destinan al entrenamiento, mientras que el 20 % restante se utiliza para la validación del modelo. Estos conjuntos se organizan en los directorios correspondientes de acuerdo con la estructura previamente establecida.

4.5.2 Ajuste de Hiperparámetros

Durante el entrenamiento del modelo YOLOv8 para la detección de objetos, se hizo énfasis en una selección meticulosa de hiperparámetros para optimizar tanto el rendimiento como la precisión. Aunque YOLOv8 ofrece una amplia variedad de hiperparámetros configurables, se optó por concentrarse en los más críticos, además de algunos adicionales, para mejorar la eficacia del modelo (ver Tabla 2).

Tabla 2

Hiperparámetros influyentes

Hiperparámetros del entrenamiento	
Hiperparámetros	Descripción
Épocas (epochs)	Se ajusta la cantidad de épocas para lograr un equilibrio entre un aprendizaje exhaustivo y la prevención del sobreajuste.
Tamaño del lote (batch)	Se establece un tamaño de lote apropiado, fundamental para la eficiencia del procesamiento y la actualización eficaz de los parámetros.
Tamaño de la imagen (imgsz)	Se define un tamaño ideal de imagen para asegurar una detección precisa en diferentes escalas.
Modelo preentrenado (pretrained)	Se utiliza un modelo previamente entrenado para aprovechar el conocimiento adquirido y aumentar la exactitud.
Optimizador	Se selecciona el optimizador más adecuado para alcanzar una convergencia rápida y efectiva.
Tasa de aprendizaje inicial (lr0)	Se ajusta minuciosamente para permitir modificaciones precisas en los pesos del modelo.

Decaimiento de peso (weight_decay):	Este parámetro ayuda a regularizar el modelo, ayudando a evitar el sobreajuste.
Precisión mixta automática (amp):	Se emplea para acelerar el entrenamiento, manteniendo la exactitud mediante el uso de precisión mixta.
Validación (val):	Es crucial para evaluar el rendimiento del modelo en datos no vistos y garantizar una buena capacidad de generalización.

Nota. En esta tabla se detallan los parámetros para maximizar las capacidades de YOLOv8.

4.6 Anaconda Navigator

Anaconda es una plataforma de computación científica y procesamiento de datos basada en Python, que integra una amplia variedad de bibliotecas de terceros altamente útiles. Al instalar Anaconda, se instala automáticamente Python junto con algunas bibliotecas comunes como Numpy, Pandas, Scrapy, Matplotlib y Ultralytics, simplificando así el proceso en comparación con una instalación estándar de Python.

Al ser un *software* libre, solo se requiere ingresar a la página oficial de Anaconda para descargar e instalarlo según el sistema operativo necesario. Tras la instalación de Anaconda, se crea un nuevo entorno en el que se instalarán las bibliotecas y aplicaciones necesarias para la programación (ver Figura 9). En esta instancia, se hallan instaladas 263 librerías (observar recuadro morado)

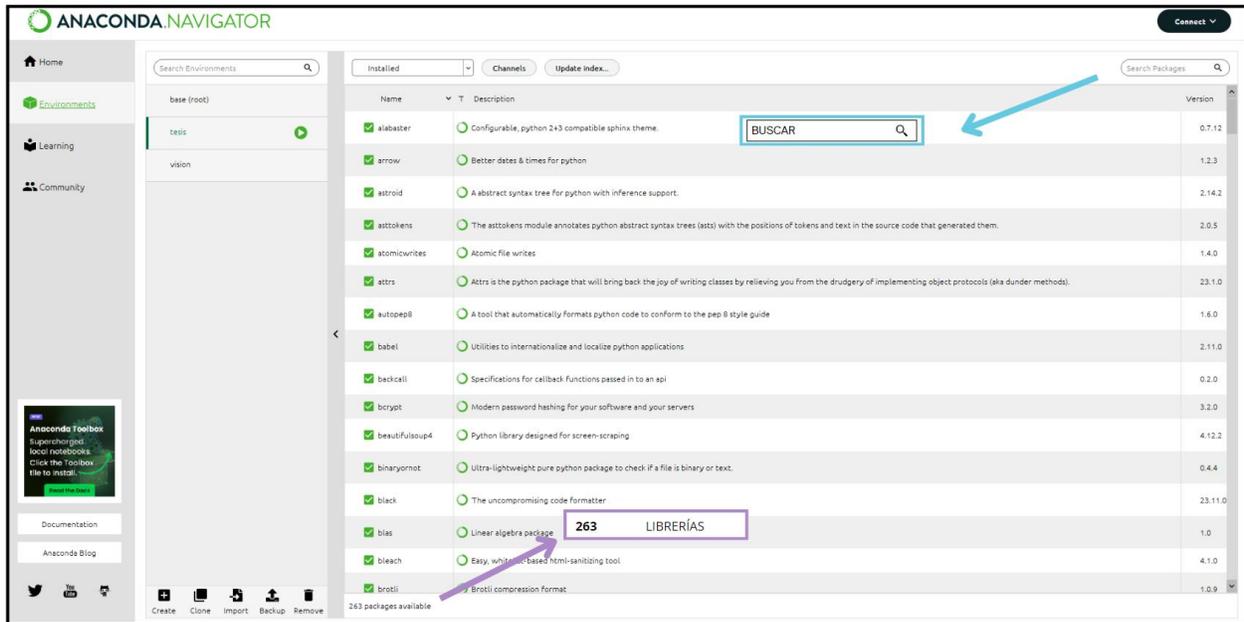


Figura 9 Interfaz de Anaconda Navigator.

4.7 Spyder

Spyder es un entorno de desarrollo integrado (IDE) multiplataforma y de código abierto, cuyo código fuente está totalmente desarrollado en Python. Fue diseñado por científicos y se enfoca exclusivamente en satisfacer las necesidades de científicos, analistas de datos e ingenieros [36].

Para llevar a cabo la instalación de Spyder en Anaconda, es importante seguir los siguientes pasos: dirigirse a “Home” y, al hacer clic en la aplicación en la palabra “Install”, Spyder se descargará e instalará en el sistema operativo seleccionado (ver Figura 10). Para efectos de este proyecto, se instaló Spyder 5.4.3.

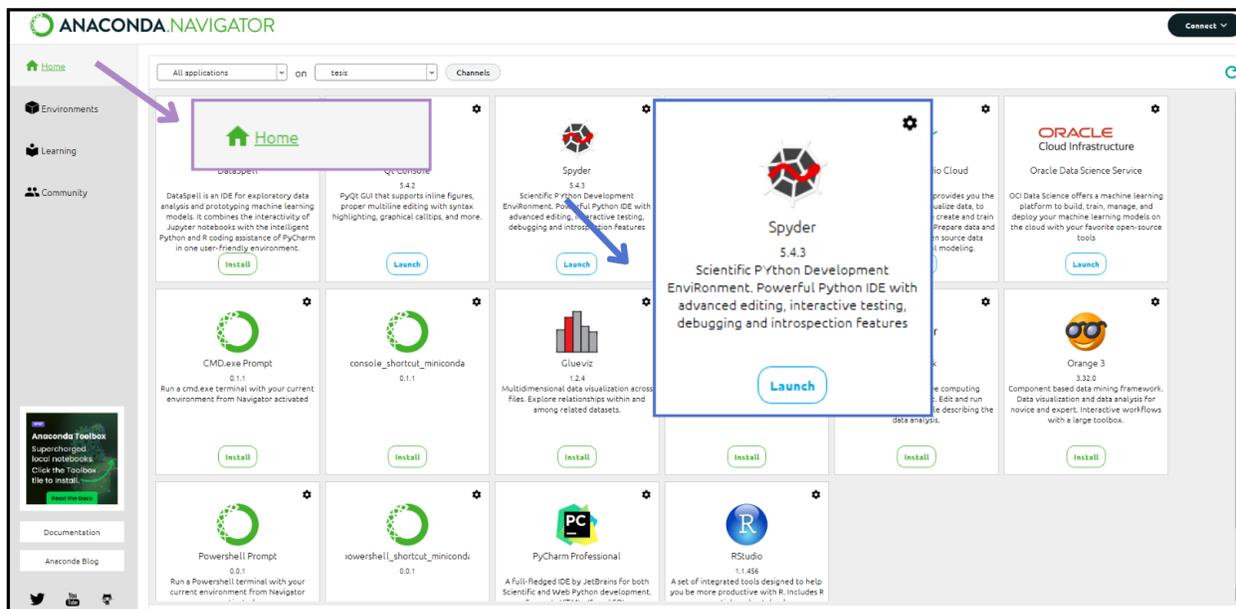


Figura 10 *Instalación de Spyder en Anaconda*

4.8 Librerías

Las librerías de Python son distintos módulos que proporcionan acceso a funcionalidades concretas del sistema, ofreciendo soluciones estandarizadas a problemas de programación. En este caso, las bibliotecas que se instalaron fueron numpy, OpenCV, sys, cv2, win32api, Ultralytics y PyQt5, junto con sus respectivas herramientas [37] (ver Tabla 3).

Tabla 3

Dependencias del Script de Detección

Librerías utilizadas	
Librerías	Descripción
NumPy	Es una librería de programación que se enfoca en facilitar la manipulación de vectores y matrices.
OpenCV	Proporciona una biblioteca que ofrece herramientas diversas para el procesamiento de imágenes y videos, diseñadas para la programación de visión por computadora y optimizadas para el rendimiento en tiempo real.
PyQT5 Designer	Se utiliza para crear una interfaz gráfica de usuario que muestra los resultados y permite la generación de reportes

sys	Se emplea para la gestión de excepciones y otros utilitarios del sistema, ofreciendo funciones y variables que permiten interactuar con el entorno del intérprete de Python y controlar la ejecución del programa.
win32api	Se utiliza para interactuar con la <i>Application Programming Interface</i> (API) de Windows cuando es necesario, permitiendo el acceso a diversas funcionalidades del sistema operativo, como la gestión de archivos, el control de dispositivos y la manipulación de ventanas y procesos.
Ultralytics	Se emplea para la detección de objetos en las imágenes, ofreciendo capacidades avanzadas para identificar y localizar diferentes tipos de objetos dentro de una imagen.

Nota. En esta tabla se detallan las librerías utilizadas dentro del *script* de detección de imágenes y conexión con la base de datos.

4.9 Diagrama de Flujo del Algoritmo

En la Figura 11 se muestra el diagrama de flujo del algoritmo principal que se aplicará en la interfaz, detallando el proceso de ejecución y operación del sistema realizado en la herramienta de diagramación en línea Microsoft Visio.

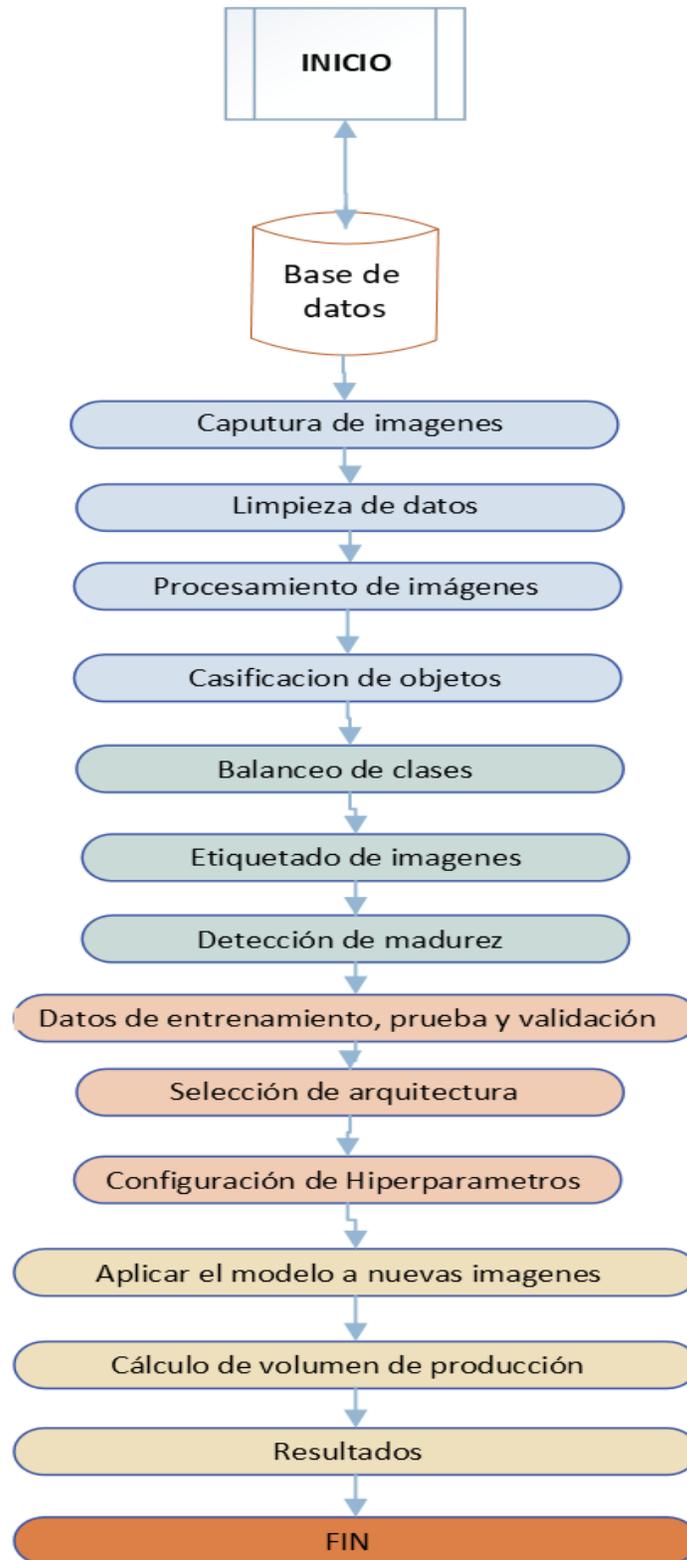


Figura 11 Diagrama de flujo de algoritmo principal

4.10 Interfaz de Usuario

Es esencial que la interfaz de usuario sea fácil de comprender y utilizar, por lo que se debe minimizar la cantidad de texto presente. En su lugar, se priorizó el uso de elementos gráficos como botones y etiquetas que sean intuitivos y de fácil manejo.

4.10.1 QT Designer

Es un instrumento que actúa como una interfaz gráfica de usuario. PyQt facilita el desarrollo del código Python utilizando Qt Designer y la incorporación de nuevos controles de interfaz gráfica al emplear tanto Qt Designer como el lenguaje de programación Python [38]. La interfaz se divide en tres áreas principales: el área de control, en color celeste, que contiene todos los widgets disponibles para el diseño del sistema; el área de proyecto, en color morado, donde se colocan los elementos deseados; y el área de propiedades, en color durazno, que permite inspeccionar y ajustar los atributos de cada objeto añadido a la interfaz (ver Figura 12).

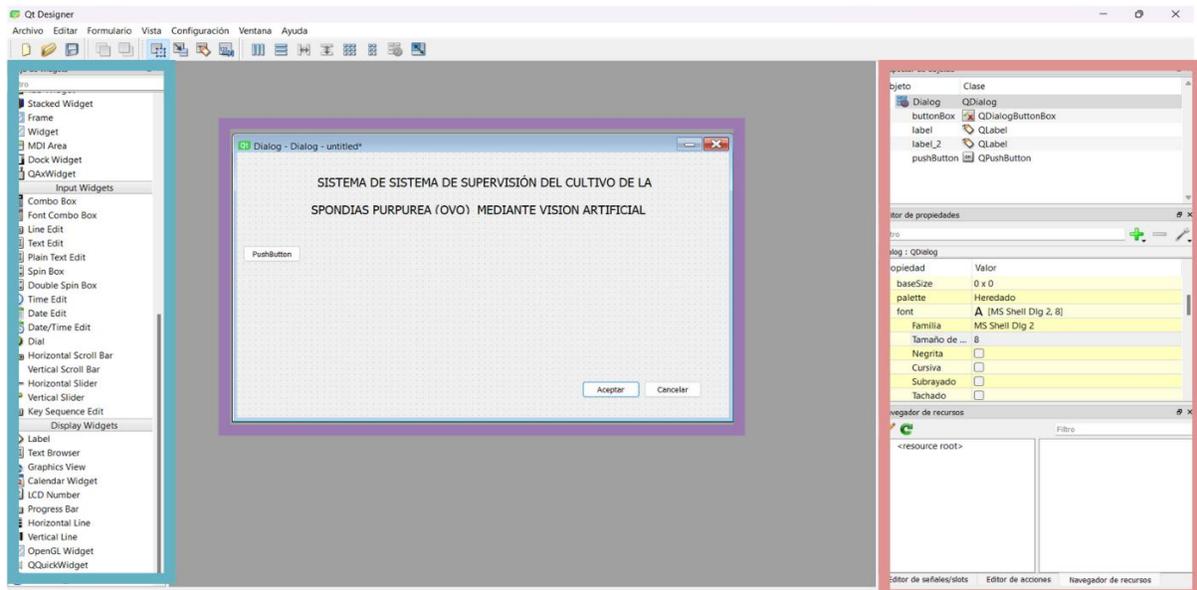


Figura 12 Áreas de trabajo QT Designer

4.10.2 Verificación y Adherencia a la Norma ISO 25010

Se realizaron pruebas exhaustivas para evaluar la capacidad del sistema para detectar y asegurar que pueda identificar con exactitud y confianza los distintos estados de maduración de la *Spondias purpurea* en las imágenes suministradas. Estas evaluaciones permitieron comprobar la efectividad del modelo YOLOv8 y confirmar que proporciona resultados de alta calidad.

Además, se llevó a cabo un análisis de eficiencia para comprobar el cumplimiento de la norma ISO 25010 en este contexto. Se evaluaron el tiempo de respuesta, el consumo de memoria y otros factores para evaluar la eficiencia del sistema.

Se tomaron medidas para mejorar la eficiencia del sistema, reduciendo el uso de recursos de computación y aumentando la capacidad de respuesta. Al adherirse a los criterios de eficiencia definidos en la norma ISO 25010, se pretende ofrecer a los usuarios una experiencia óptima y maximizar el rendimiento del sistema en cuanto al tiempo y los recursos utilizados.

4.11 Supervisión del Volumen de Producción

La supervisión del volumen de producción de ovos es un proceso crucial que permite evaluar y gestionar la cantidad de producto disponible en diferentes etapas de maduración. A través de la implementación de sistemas de detección basados en la detección de imágenes, se clasificaron los ovos en categorías como rojos, verdes y amarillos, reflejando así su estado de desarrollo. Esta clasificación precisa permitió ajustar las estrategias de recolección y distribución de manera eficiente, asegurando que los ovos sean cosechados y distribuidos en el momento óptimo. Todos los datos recopilados sobre el volumen de producción se registran y se exportan a un archivo Excel, lo que facilitó su análisis y seguimiento. Al llevar un control riguroso del volumen de producción y disponer de estos datos en un formato accesible, se mejora la toma de decisiones en la gestión

de recursos, se optimizan los procesos de producción y se garantiza una alta calidad del producto final.

4.12 Métricas del Modelo YOLOv8

4.12.1 Fase de Entrenamiento

En la etapa de entrenamiento, se llevó a cabo un análisis meticuloso de las métricas de pérdida del modelo. Se utilizaron 127 imágenes de entrenamiento y 23 imágenes de validación.

Después del entrenamiento se obtuvieron resultados que incluyeron:

- La pérdida de caja, que evalúa la precisión con la que el modelo estima la ubicación y el tamaño de las cajas delimitadoras de los frutos de la *Spondias purpurea*.
- La pérdida de clasificación, que mide la capacidad del modelo para distinguir los frutos de la *Spondias purpurea* de hojas y otros objetos.
- La pérdida de distribución de puntos de anclaje, que indica la exactitud en la predicción de puntos específicos dentro de las cajas delimitadoras [39].

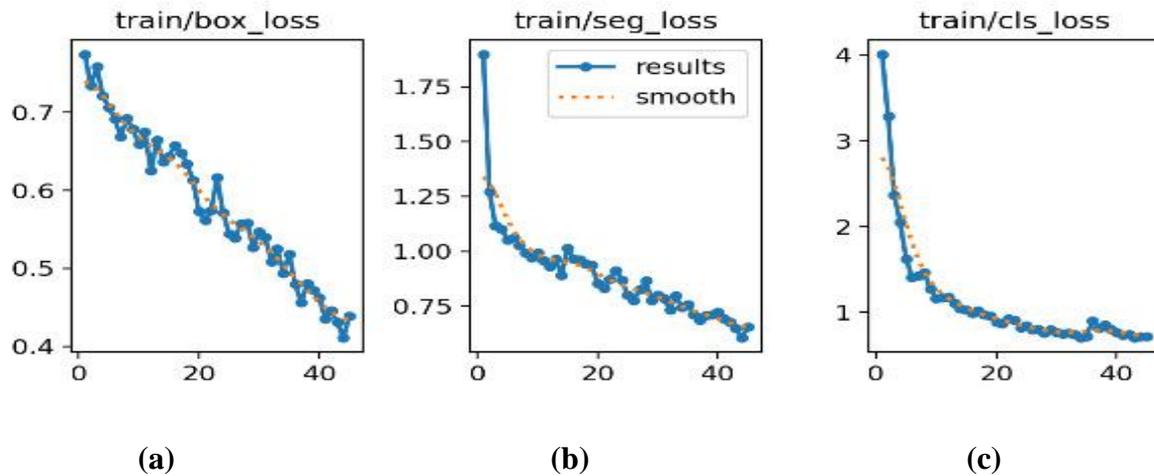


Figura 13 Métricas en entrenamiento (a) pérdida de caja, (b) pérdida de clasificación y (c) pérdida de distribución.

Se presentan tres gráficos de pérdida durante el entrenamiento, cada uno representando un tipo diferente de pérdida en función de las épocas del entrenamiento, aparentemente hasta la época 50. El primer gráfico, "train/box_loss", muestra la pérdida de la caja durante el entrenamiento. Se observa una disminución constante de la pérdida, comenzando alrededor de 0.7 y descendiendo a aproximadamente 0.4, con algunas fluctuaciones menores, lo que indica una mejora en la predicción de las cajas delimitadoras con el tiempo. El segundo gráfico, "train/seg_loss", ilustra la pérdida de segmentación, que también disminuye de manera constante desde aproximadamente 1.75 hasta 0.75, sugiriendo una mejora continua en la precisión de segmentación del modelo. Finalmente, el tercer gráfico, "train/cls_loss", muestra la pérdida de clasificación. La pérdida comienza alrededor de 4 y disminuye de manera significativa y rápida, estabilizándose alrededor de 1 después de varias épocas. Esta disminución pronunciada inicial indica una mejora rápida en la clasificación durante las primeras épocas de entrenamiento, seguida de una estabilización a medida que se aproxima a una pérdida más baja. En conjunto, los tres gráficos demuestran una

tendencia decreciente en las pérdidas correspondientes, lo que sugiere que el modelo está mejorando su rendimiento en términos de predicción de cajas, segmentación y clasificación a medida que avanza el entrenamiento. Estas métricas fueron esenciales para ajustar y afinar el modelo, destacando la precisión en la identificación de los frutos de las *Spondias purpurea*.

4.12.2 Fase de Validación

Durante la fase de validación, se evaluó el modelo con las mismas métricas de pérdida, pero aplicadas a un conjunto de datos distinto del utilizado en el entrenamiento. Este procedimiento fue esencial para verificar la capacidad del modelo de generalizar su aprendizaje a nuevas situaciones, lo cual es crucial para prevenir el sobreajuste [40]. De esa manera, se garantizó que el modelo no solo memorizara los datos de entrenamiento, sino que también pudiera aplicar su conocimiento a nuevas imágenes para identificar la madurez de la *Spondias purpurea* (ver Figura 14).

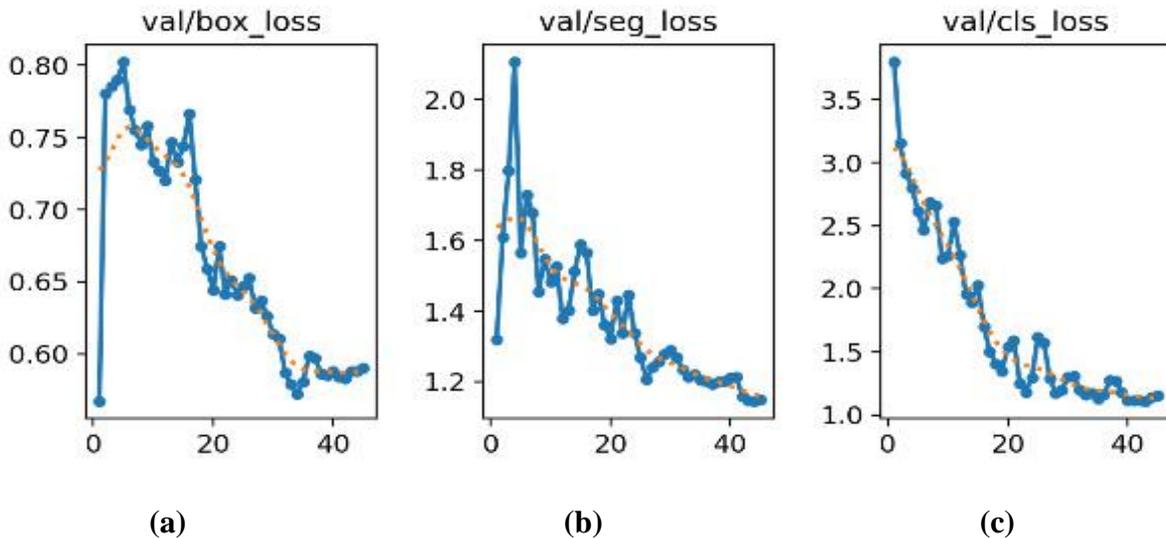


Figura 14 Métricas en validación. (a) Validación en cajas, (b) validación de clasificación y (c) validación de pérdida.

Se observa tres gráficos de pérdida durante la validación del modelo, cada uno representando un tipo diferente de pérdida en función de las épocas del entrenamiento (hasta la época 50). En el gráfico "val/box_loss" se observa una pérdida que comienza alrededor de 0.8, presenta una ligera subida inicial y luego disminuye de manera constante hasta estabilizarse alrededor de 0.6, sugiriendo una mejora en la predicción de las cajas delimitadoras. El gráfico "val/seg_loss" ilustra una pérdida de segmentación que empieza cerca de 2.0 y disminuye significativamente, estabilizándose alrededor de 1.2, indicando una rápida y continua mejora en la precisión de segmentación. Por último, el gráfico "val/cls_loss" muestra una pérdida de clasificación que comienza alrededor de 3.5 y disminuye de manera constante, estabilizándose alrededor de 1.0, lo que sugiere una mejora continua en la precisión de clasificación del modelo durante

4.12.3 Fase de Resultados

La fase de resultados se centra en métricas cruciales como la precisión, el recall, el *mean Average Precision* (mAP), matriz de confusión y la curva F1-Confidence.

La matriz de confusión es una herramienta útil para evaluar el rendimiento del modelo, evidenciando el conteo de verdaderos positivos (VP), falsos positivos (FP), verdaderos negativos (VN) y falsos negativos (FN). Esta matriz facilita la comprensión de las situaciones en las que el modelo tiene éxito y en las que falla. En este caso, permitió determinar si el modelo estaba funcionando adecuadamente y evaluar la precisión de la predicción de las distintas clases [41].

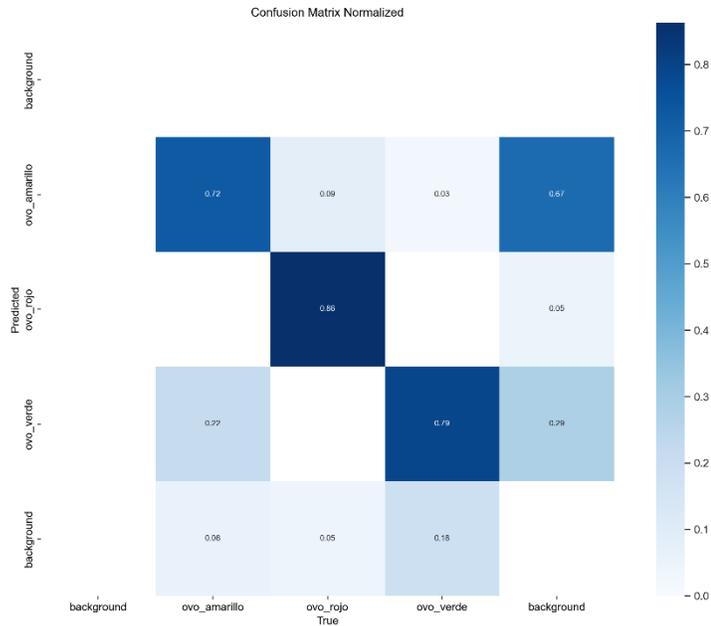


Figura 15 *Matriz de confusión*

Se visualiza una matriz de confusión normalizada, utilizada para evaluar el rendimiento de un modelo de clasificación. La matriz compara las predicciones del modelo (eje vertical) con las etiquetas verdaderas (eje horizontal). En "background", se observa una precisión del 72%, con confusiones con "ovo_amarillo" (9%), "ovo_rojo" (3%) y "ovo_verde" (67%). La clase "ovo_amarillo" presenta una alta precisión del 86%, con confusiones con "background" (5%) y "ovo_rojo" (5%). La clase "ovo_rojo" tiene una precisión del 79%, con confusiones con "background" (7%), "ovo_amarillo" (12%) y "ovo_verde" (29%). Finalmente, la clase "ovo_verde" muestra una precisión del 67%, con confusiones con "background" (18%), "ovo_amarillo" (15%) y "ovo_rojo" (16%). En conjunto, la matriz de confusión indica que el modelo tiene alta precisión en "ovo_amarillo" y razonable precisión en "ovo_rojo", "background" y "ovo_verde", sugiriendo que el modelo podría beneficiarse de mejoras para reducir las confusiones entre estas clases.

4.12.4 Precisión

Una alta precisión mostró que el modelo posee un desempeño sólido al reconocer correctamente los distintos estados de maduración del ovo, reduciendo al mínimo la confusión con objetos que no son el fruto (ver Figura 15).

$$\text{Precisión} = \frac{\text{Verdaderos Positivos (VP)}}{\text{Verdaderos Positivos (VP)} + \text{Falsos Positivos (FP)}} \quad \text{Ecuación 1}$$

Donde:

- **Verdaderos positivos (VP):** hace referencia a los casos en los que se identifica correctamente la clase de objetivo (fruto de la *Spondias purpurea*).
- **Falsos Positivos (FP):** se refiere a situaciones en las que se clasifica incorrectamente la clase objetivo en objetos que no pertenecen a ella.

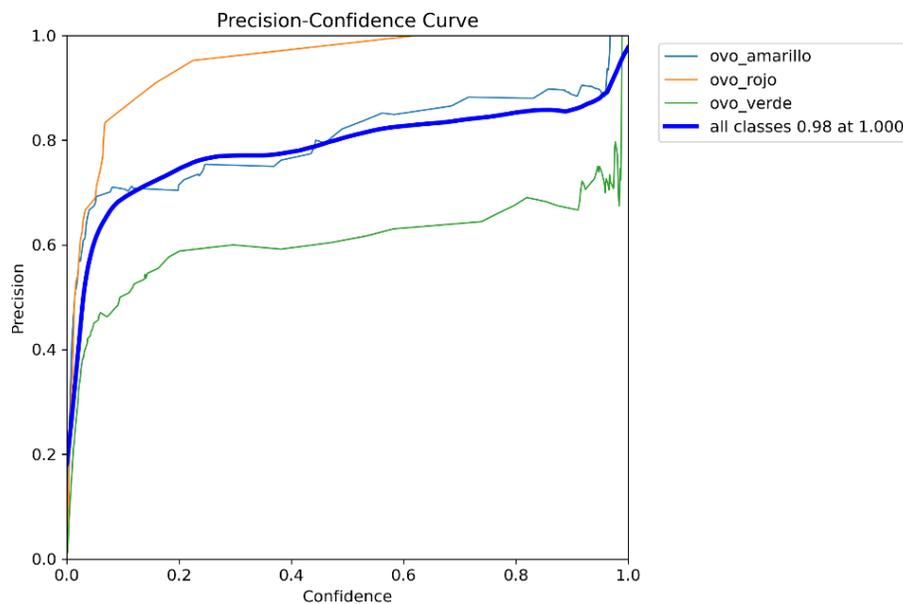


Figura 15 Gráfica Precisión - Confianza

La gráfica muestra una curva de precisión-confianza del modelo de detección de objetos. En ella, se observa que la precisión para la clase "ovo_amarillo" aumenta rápidamente con la confianza y se mantiene alta a partir de una confianza de aproximadamente 0.3. La clase "ovo_rojo" presenta una precisión muy alta que se mantiene cerca de 1.0 a lo largo de todo el rango de confianza. Por otro lado, la precisión de la clase "ovo_verde" es menor y más variable en comparación con las otras dos clases, mejorando gradualmente con la confianza, pero sin alcanzar los niveles de las demás. La línea azul gruesa, que representa la precisión promedio para todas las clases, muestra que la precisión global del modelo es alta, alcanzando un valor de 0.98 a una confianza de 1.0. Esto sugiere que el modelo tiene un rendimiento robusto en general, aunque podría necesitar mejoras en la detección de la clase "ovo_verde". La precisión representa la proporción de verdaderos positivos sobre el total de predicciones positivas, y un umbral de confianza más alto implica que el modelo será más estricto al hacer predicciones, reduciendo los falsos positivos, pero posiblemente aumentando los falsos negativos. En resumen, el modelo es efectivo en general, pero presenta áreas para mejorar, especialmente en la detección de "ovo_verde".

4.12.5 Recall

Evalúa la habilidad de un modelo para identificar correctamente todos los casos positivos en un conjunto de datos. Este valor oscila entre 0 y 1, donde un valor de 1 indica que el modelo ha detectado correctamente todos los casos positivos sin producir falsos negativos. La peor situación es la contraria: un valor de 0 indica que no se ha detectado ningún caso positivo.

$$Recall = \frac{Verdaderos\ Positivos\ (VP)}{Verdaderos\ Positivos\ (VP) + Falsos\ Negativos\ (FN)} \quad \text{Ecuación 1}$$

En esta fórmula:

- **Verdaderos Positivos (VP):** se trata de situaciones en las que se identifica correctamente la clase de interés (ejemplo: distintos estados de maduración del ovo).
- **Falsos Negativos (FN):** se refiere a los casos en los que el modelo no detecta la clase de interés, a pesar de que esta está efectivamente presente en la imagen.

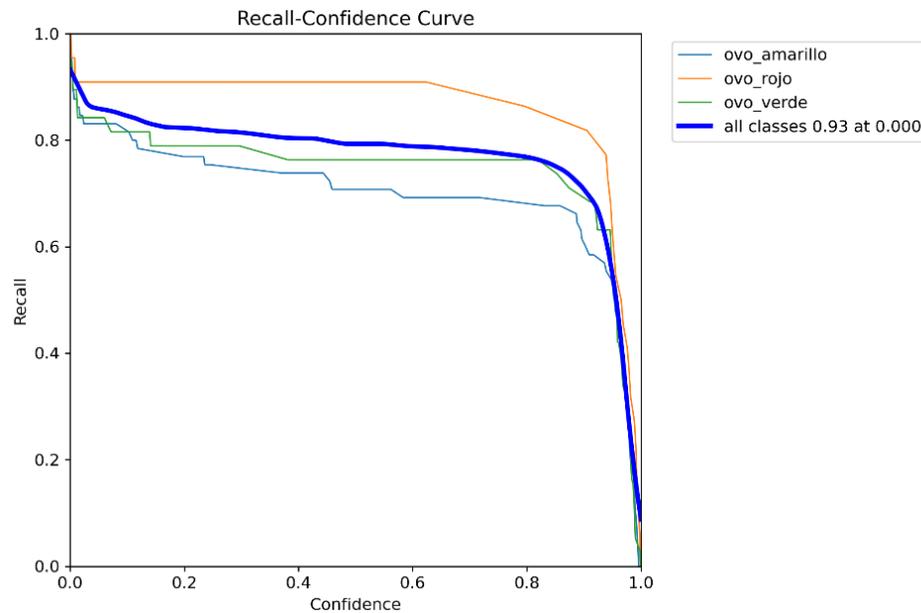


Figura 16 Gráfica Recall – Confianza

Se evalúa una curva de recall-confianza para el modelo de detección de objetos. En ella, se observa que el recall para la clase "ovo_amarillo" comienza alto y disminuye gradualmente con el aumento de la confianza, manteniéndose relativamente estable hasta una confianza de aproximadamente 0.8, después de lo cual disminuye rápidamente. La clase "ovo_rojo" presenta un recall muy alto que se mantiene cerca de 1.0 a lo largo de la mayoría del rango de confianza, pero

disminuye bruscamente cerca de una confianza de 1.0. Por otro lado, el recall de la clase "ovo_verde" es más variable, comenzando alto, pero disminuyendo gradualmente con la confianza. La línea azul gruesa, que representa el recall promedio para todas las clases, indica que el modelo tiene una buena capacidad para identificar objetos correctamente en general, con un valor de 0.93 a una confianza de 0.0. Sin embargo, se observa una disminución en el recall a medida que aumenta el umbral de confianza. Esto sugiere que el modelo es robusto en términos generales, aunque podría mejorarse la detección de la clase "ovo_verde" y mantener un alto recall a niveles altos de confianza.

4.12.6 Mean Average Precision

El mAP ofreció una evaluación integral al combinar precisión y recall en varios umbrales de confianza, brindando una visión global del desempeño del modelo en diferentes condiciones. Esta métrica fue fundamental para validar la aplicabilidad del modelo en la detección práctica de los estados de maduración del ovo en escenarios agrícolas reales (ver Figura 17).

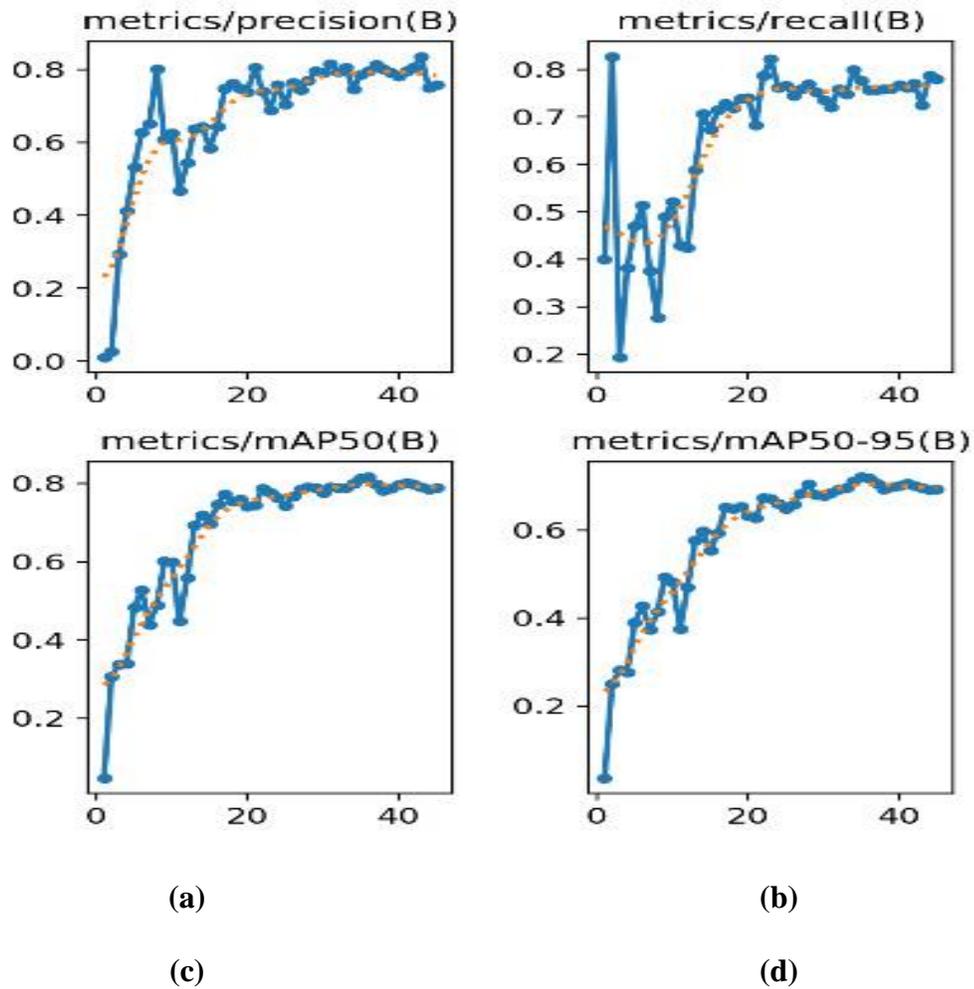


Figura 17 Resultados del modelo (a)métrica – precisión, (b)métricas - recall, (c)métricas - mAP50, (d) métricas - mAP50-95.

La serie de gráficos proporcionada muestra las métricas de rendimiento del modelo de detección de objetos en función de las épocas de entrenamiento. En la gráfica superior izquierda, se observa que la precisión del modelo aumenta progresivamente y se estabiliza alrededor de un valor de 0.8 después de aproximadamente 20 iteraciones. En la gráfica superior derecha, el recall del modelo muestra una tendencia inicial volátil, pero mejora y se estabiliza cerca de 0.7 después de 20 iteraciones. La gráfica inferior izquierda, que representa el mAP50, indica que la media de

precisión promedio al 50% de intersección sobre la unión también incrementa gradualmente, alcanzando un valor cercano a 0.8. Finalmente, la gráfica inferior derecha muestra que el mAP50-95, que evalúa la precisión promedio en un rango de umbrales de intersección sobre la unión, sigue una tendencia similar, incrementando de manera constante y estabilizándose alrededor de 0.6. En conjunto, estas métricas sugieren que el modelo mejora su rendimiento a lo largo del entrenamiento y logra una precisión y recall satisfactorios, así como buenos valores de mAP50 y mAP50-95 después de 20 iteraciones.

4.12.7 F1 – Confianza

Se trata de una métrica que integra los valores de precisión y recall, ofreciendo una medida más completa del rendimiento. Los valores de esta métrica oscilan entre 0 y 1, siendo los valores cercanos a 1 indicativos de un rendimiento óptimo del modelo. Una adecuada calibración de esta curva es esencial para la aplicación práctica del modelo en la identificación de los estados de maduración del ovo en contextos agrícolas (ver Figura 18).

$$F1 = 2 \times \frac{\textit{Precisión} \times \textit{Recall}}{\textit{Precisión} + \textit{Recall}} \qquad \text{Ecuación 2}$$

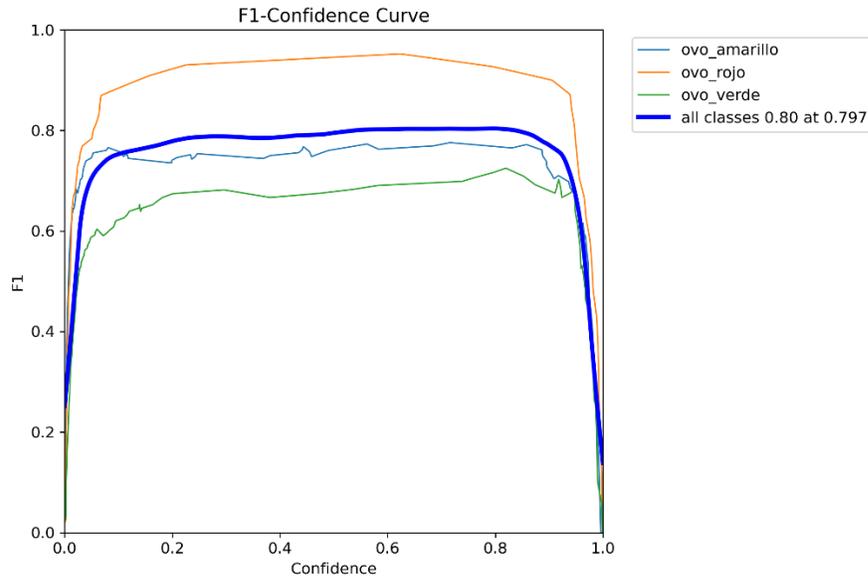


Figura 18 Curva F1 - Confianza

La gráfica muestra la relación entre el puntaje F1 y la confianza para tres clasificaciones distintas ("ovo_amarillo", "ovo_rojo", "ovo_verde") y una combinación de todas las clases ("all classes"). Se observa que "ovo_amarillo" tiene un comportamiento más consistente en comparación con las otras clases, manteniendo un puntaje F1 relativamente alto a lo largo de una amplia gama de valores de confianza. "Ovo_rojo" presenta el puntaje F1 más alto en general, especialmente en la zona de alta confianza (cerca de 1.0), indicando que esta clase se predice con mayor precisión. Por otro lado, "ovo_verde" tiene un puntaje F1 más bajo, lo que sugiere que es la clase más difícil de predecir correctamente. La curva combinada, representada por el azul oscuro, mantiene un puntaje F1 alto en una amplia gama de valores de confianza, alcanzando un valor de 0.80 a una confianza de 0.797, lo que sugiere un buen desempeño general del modelo para todas las clases en esta confianza. A medida que la confianza aumenta, el puntaje F1 también aumenta inicialmente, alcanzando un pico antes de estabilizarse o disminuir ligeramente. En los extremos de la curva de confianza (cerca de 0 y 1), se observa una caída en el puntaje F1, lo que puede indicar que en estas áreas el modelo es menos seguro y, por lo tanto, menos preciso, y el

rendimiento global del modelo es robusto en la mayoría de los rangos de confianza, alcanzando un puntaje F1 de 0.80 cuando la confianza es 0.797.

Por otro lado, se ha utilizado para comparar con modelos de trabajos similares en el ámbito de la detección con IA, dado que permite evaluar la eficacia del modelo en la detección de la mayor cantidad posible de ovos con distintos estados de maduración. Además, una vez identificado un ovo, esta medida permite determinar con precisión los distintos estados de maduración encontrados, al combinar el recall y la precisión (ver Figura 19).

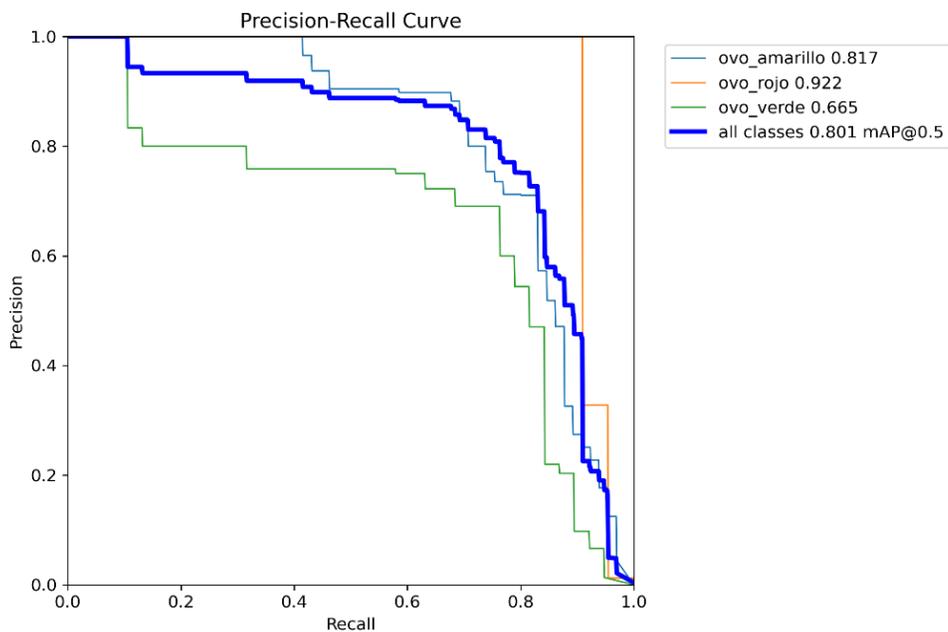


Figura 19 *Precisión – Recall*

Se analiza la curva de precisión-recall para las tres clasificaciones ("ovo_amarillo", "ovo_rojo", "ovo_verde") y una combinación de todas las clases. La clase "ovo_rojo" alcanza la mayor precisión con un área bajo la curva (AP) de 0.922, indicando un excelente desempeño. "Ovo_amarillo" también muestra un buen desempeño con un AP de 0.817. En contraste, "ovo_verde" presenta el menor AP de 0.665, lo que sugiere mayor dificultad para predecir correctamente esta clase. La curva combinada para todas las clases, representada en azul oscuro,

muestra un promedio de precisión-recall (mAP) de 0.801 a un umbral de 0.5, reflejando un buen rendimiento global del modelo. En general, se observa que las predicciones son más precisas cuando el recall es bajo, disminuyendo la precisión a medida que el recall aumenta.

Tabla 4

Métricas en validación

Class	Imágenes	Intances	Box(P)	R
all	23	142	0,797	0,753
ovo_amarillo	11	78	0,807	0,59
ovo_rojo	16	23	0,952	0,913
ovo_verde	13	41	0,632	0,756
Speed: 0.4ms preprocess	26,5ms	inference, 0.	loss	2,9ms postprocess

Tabla 5

Métricas en validación

Class	mAP50-95)	Mask (P	R	mAP50	Map50-95):100
all	0,715	0,797	0,753	0,185	0,687
ovo_amarillo	0,661	0,807	0,59	0,796	0,614
ovo_rojo	0,888	0,952	0,913	0,956	0,85
ovo_verde	0,598	0,632	0,756	0,692	0,598
Speed: 0.4ms preprocess					

4.13 Descripción de la Programación

La implementación y evaluación del modelo se realizaron empleando el entorno de desarrollo Spyder, una plataforma de código abierto que es comúnmente utilizada para la programación en Python debido a sus características de fácil uso y sus potentes herramientas integradas.

4.13.1 Configuración del Entorno de Trabajo

Para la sección de lógica del proyecto, primero se importaron todas las librerías y dependencias necesarias. Se instaló y configuró el entorno de desarrollo Spyder, asegurándose de que todas las dependencias y bibliotecas necesarias estuvieran correctamente instaladas (ver Figura 20). Enlace directo para la programación: <https://goo.su/veyA9>

```
8
9  from PyQt5 import QtCore, QtGui, QtWidgets
10 import numpy as np
11 import sys
12 import cv2
13 from PyQt5.QtCore import *
14 from PyQt5.QtGui import *
15 from PyQt5.QtWidgets import QDialog, QMainWindow
16 from PyQt5.Qt import Qt
17 from win32api import GetSystemMetrics
18 from PyQt5.QtWidgets import (QApplication, QDialog, QPushButton, QTableWidgetItem,
19                             QTableWidgetItem, QAbstractItemView, QHeaderView, QMenu,
20                             QActionGroup, QAction, QMessageBox)
21 from ultralytics import YOLO
22 import pandas as pd
23 import os
24 from datetime import datetime
25 import locale
```

Figura 20 Librerías importadas en Spyder

4.13.2 Diseño de Interfaces con Qt Designer

Se empleó Qt Designer para crear la interfaz gráfica que facilita la interacción con el modelo y la visualización de los resultados. Esta interfaz permitió la carga de datos, la configuración de parámetros y la visualización de las detecciones realizadas por el modelo de manera intuitiva y accesible [42]. La interfaz diseñada se integró con los *scripts* de Python utilizando PyQt, lo que permitió una interacción fluida entre la lógica del modelo y la presentación visual. Se realizó desde la línea de código 30 hasta la línea de código 391 (ver Figura 21). Enlace directo para la programación: <https://goo.su/veyA9>

```

30 class Ui_MainWindow(object):
31     def setupUi(self, MainWindow):
32         MainWindow.setObjectName("MainWindow")
33         MainWindow.resize(1029, 644)
34         palette = QtGui.QPalette()
35         brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
36         brush.setStyle(QtCore.Qt.SolidPattern)
37         palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Midlight, brush)
38         brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
39         brush.setStyle(QtCore.Qt.SolidPattern)
40         palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Base, brush)
41         brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
42         brush.setStyle(QtCore.Qt.SolidPattern)
43         palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Window, brush)
44         brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
45         brush.setStyle(QtCore.Qt.SolidPattern)
46         palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Midlight, brush)
47         brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
48         brush.setStyle(QtCore.Qt.SolidPattern)
49         palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Base, brush)
50         brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
51         brush.setStyle(QtCore.Qt.SolidPattern)
52         palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Window, brush)
53         brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
54         brush.setStyle(QtCore.Qt.SolidPattern)
55         palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Midlight, brush)
56         brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
57         brush.setStyle(QtCore.Qt.SolidPattern)
58         palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Base, brush)
59         brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
60         brush.setStyle(QtCore.Qt.SolidPattern)
61         palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Window, brush)
62         MainWindow.setPalette(palette)

```

Figura 21 Código exportado de QT Designer

4.13.3 Implementación del Modelo YOLOv8

Utilizando la biblioteca Ultralytics, se implementó el modelo YOLOv8 [43]. Esta biblioteca ofrece una interfaz sencilla para el entrenamiento, evaluación y ajuste fino de modelos de detección de objetos. Se desarrollaron *scripts* en Python para cargar y preprocesar los datos, configurar los parámetros del modelo y ejecutar el entrenamiento. Estos *scripts* incluían funciones para la normalización de imágenes, la generación de lotes de datos y la aplicación de aumentación de datos para mejorar la robustez del modelo. Se realizó desde la línea de código 425 hasta la línea de código 475 (ver Figura 22). Enlace directo para la programación: <https://goo.su/veyA9>

```

425 def prProcesar(self):
426     global filename
427     self.captured = cv2.imread(str(filename))
428     if not hasattr(self, "captured"):
429         return
430     imagen = self.captured
431
432     model = YOLO('save_best.pt') # Cargamos el modelo de IA en formato (pytorch)
433
434     results = model(imagen) # Iniciamos la prediccion del modelo y lo guardamos en results
435
436     detections = []
437     excel_path = 'resultados.xlsx' # Ruta al archivo excel
438
439     for result in results: # Revisamos los resultados
440         for box in result.boxes: # Assuming 'result.boxes' contains the detection boxes
441             detection = {
442                 'confidence': box.conf.item(),
443                 'class': box.cls.item(),
444             }
445             detections.append(detection)
446             result.save(filename="Procesada.jpg") # Guardar la imagen procesada con el nombre "Procesada.jpg" en la ubicación del archivo
447
448     self.guardar_resultados_en_excel(detections, excel_path) # Guardar los resultados en un excel
449
450     # Guardar la imagen procesada en la variable captured
451     captured = cv2.imread('Procesada.jpg')
452
453     # Cambiar el formato de imagen de BGR a RGB
454     img1 = cv2.cvtColor(captured, cv2.COLOR_BGR2RGB)
455
456     # Guardar los datos de la imagen procesada
457     rows, cols, channels = self.captured.shape
458
459     # Opera con los valores de las columnas y los canales para obtener el numero de bytes de la imagen
460     bytesPerLine = channels * cols

```

Figura 22 Código de implementación de YOLOv8

4.13.4 Recolección de Datos en Excel

Todos los datos recopilados sobre el volumen de producción se registraron y se exportaron a un archivo Excel, lo que facilitó su análisis y seguimiento. Al llevar un control riguroso del volumen de producción y disponer de estos datos en un formato accesible, se mejora la toma de decisiones en la gestión de recursos, se optimizan los procesos de producción y se garantiza una alta calidad del producto final (ver Figura 23). Enlace directo para la programación:

<https://goo.su/veyA9>

```

393 def guardar_resultados_en_excel(self, detections, excel_path): # Funcion para guardar detecciones en el excel
394     # Obtener la fecha y el día actual
395     now = datetime.now()
396     fecha = now.strftime("%Y-%m-%d")
397     dia = now.strftime("%A")
398
399     # Contar la cantidad de ovos de cada color
400     cantidad_amarillo = sum(1 for det in detections if det['class'] == 1)
401     cantidad_rojo = sum(1 for det in detections if det['class'] == 2)
402     cantidad_verde = sum(1 for det in detections if det['class'] == 3)
403     cantidad_ovos_analizados = len(detections)
404
405     # Crear un DataFrame con los resultados
406     data = {
407         'fecha': [fecha],
408         'dia': [dia],
409         'cantidad de ovos analizados': [cantidad_ovos_analizados],
410         'verde': [cantidad_verde],
411         'amarillo': [cantidad_amarillo],
412         'rojo': [cantidad_rojo]
413     }
414     df = pd.DataFrame(data)
415
416     # Si el excel no existe, crearlo y guardar el Dataframe
417     if not os.path.isfile(excel_path):
418         df.to_excel(excel_path, index=False)
419     else:
420         # Si el excel existe, leerlo y agregar nuevas filas
421         existing_df = pd.read_excel(excel_path)
422         updated_df = pd.concat([existing_df, df], ignore_index=True)
423         updated_df.to_excel(excel_path, index=False)

```

Figura 23 Código exportar a Excel

4.14 Desarrollo de la Interfaz Gráfica de Usuario

Como resultado de finalizar la fase de desarrollo de la interfaz gráfica de usuario, se creó una propuesta de interfaz para la identificación de los diversos estados de maduración de la Spondias purpurea. A continuación, se detalla el contenido de la GUI desarrollada.

En la Figura 24 se muestra la ventana resultante del diseño creado en QT Designer, y en la Figura 25 se presenta la programación de los elementos de la interfaz mediante Spyder.

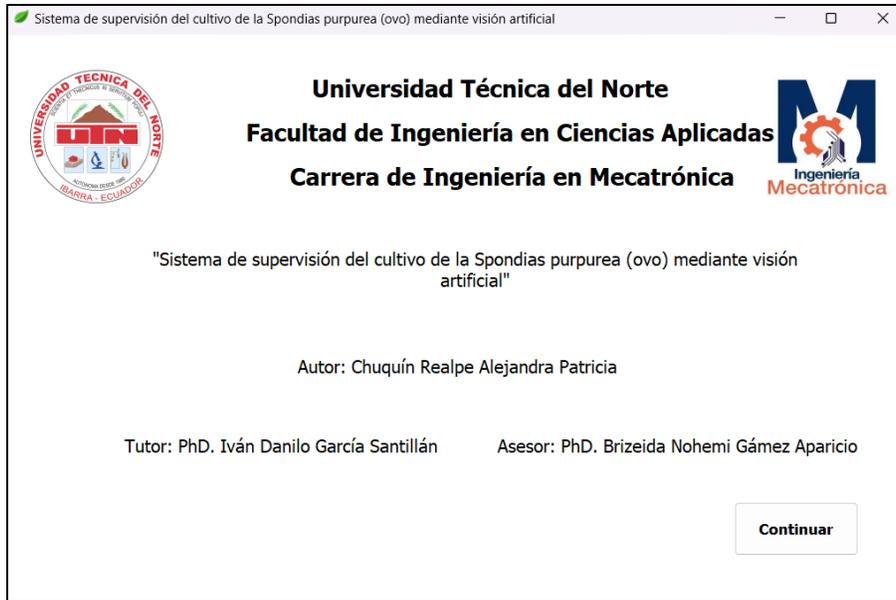


Figura 24 Portada de la interfaz gráfica



Figura 25 Interfaz principal del sistema

Desde una perspectiva preliminar, se observan los *labels* destinados al título y a las imágenes a mostrar. Los *labels* de color blanco son los que exhiben la imagen a procesar y la imagen procesada. Además, se disponen de tres botones, cada uno con una función específica:

- **Abrir Imagen:** abre un navegador de archivos que permite seleccionar una, según el requerimiento del usuario. Una vez pulsado, la imagen seleccionada se muestra en el *label* blanco de menor tamaño.
- **Procesar Imagen:** una vez pulsado, se puede observar la detención de maduración de los ovos con sus respectivas etiquetas.
- **Visualizar Imagen:** abre la imagen con el programa de visualización predeterminado del sistema operativo, lo que permite una visualización más detallada de la imagen procesada.

4.14.1 Pruebas

Con el fin de desarrollar el algoritmo de visión artificial, se capturaron imágenes de prueba utilizando un iPhone X con una resolución de 2337x3116 en formato .jpg. Para recopilar datos para la detección de imágenes mediante visión artificial, se tomaron 200 fotografías al aire libre durante la tarde con luz del día, capturando imágenes desde distintos ángulos y con diferentes sombras. Aunque no se trata de un ambiente estrictamente controlado, estas condiciones permiten al algoritmo de visión artificial aprender y detectar variaciones con precisión.

Una vez completado el diseño y desarrollo de la interfaz gráfica de usuario, se realizaron pruebas operativas para confirmar que cada botón de procesamiento y etiquetado realizara la función para la que fue programado (ver Figura 26).

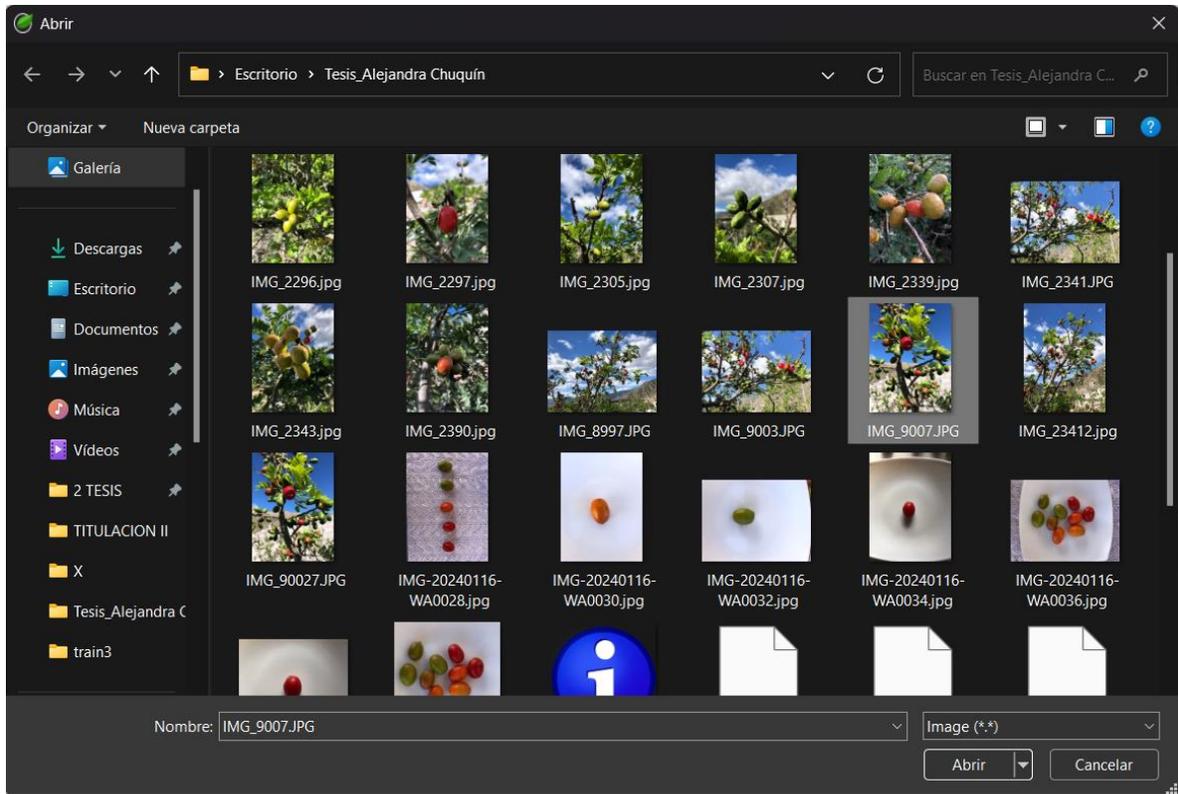


Figura 26 *Diálogo del sistema para abrir imagen*

En la Figura 27 se observa el funcionamiento del botón “Procesar”, el cual detalla la imagen procesada junto con la detección correspondiente. Para esta prueba, se tomó una imagen con un solo ovo, para poder observar de mejor forma el etiquetado de imágenes (ver Figura 27).



Clase 2 (amarillo) con 0.98 de fiabilidad

Figura 27 *Imagen procesada*

4.15 Datos de Producción de Ovos: Clasificación y Resultados en Excel

Una vez realizada la detección de los ovos y su clasificación en las respectivas etapas de maduración (rojos, verdes y amarillos), los datos resultantes se almacenan en un archivo Excel. Este registro sistemático incluye la cantidad de ovos en cada categoría para diferentes periodos de tiempo [44].

La tabla presentada en la Figura 21 ofrece un resumen detallado de la producción de ovos, clasificados en tres categorías de maduración: rojos, verdes y amarillos. La tabla muestra datos recopilados para cinco fechas específicas durante los meses de junio y julio de 2024. Cada fila representa un periodo de tiempo, detallando la cantidad de ovos en cada categoría durante ese periodo. La columna “Cantidad de ovos analizados” proporciona el total de ovos producidos, sumando los ovos rojos, verdes y amarillos. Este formato permite una rápida visualización de la

producción diaria. Los datos reflejan cómo el sistema categoriza los ovos en función de su estado de desarrollo y calcula el total de producción para cada fecha (ver Figura 28).

	H	I	J	K	L	M
13						
14	fecha	dia	cantidad de ovos analizados	verde	amarillo	rojo
15	15/06/2024	sabado	30	8	5	17
16	20/06/2024	jueves	47	30	3	14
17	16/07/2024	martes	25	13	2	10
18	19/07/2024	viernes	36	17	8	11
19	21/07/2024	domingo	42	7	17	18

Figura 28 *Recolección de resultados - Excel*

A continuación, sirve como ejemplo ilustrativo del funcionamiento del sistema de detección de ovos. En este ejemplo, se presentan los resultados obtenidos después de aplicar el sistema de clasificación basado en imágenes (ver Figura 29).

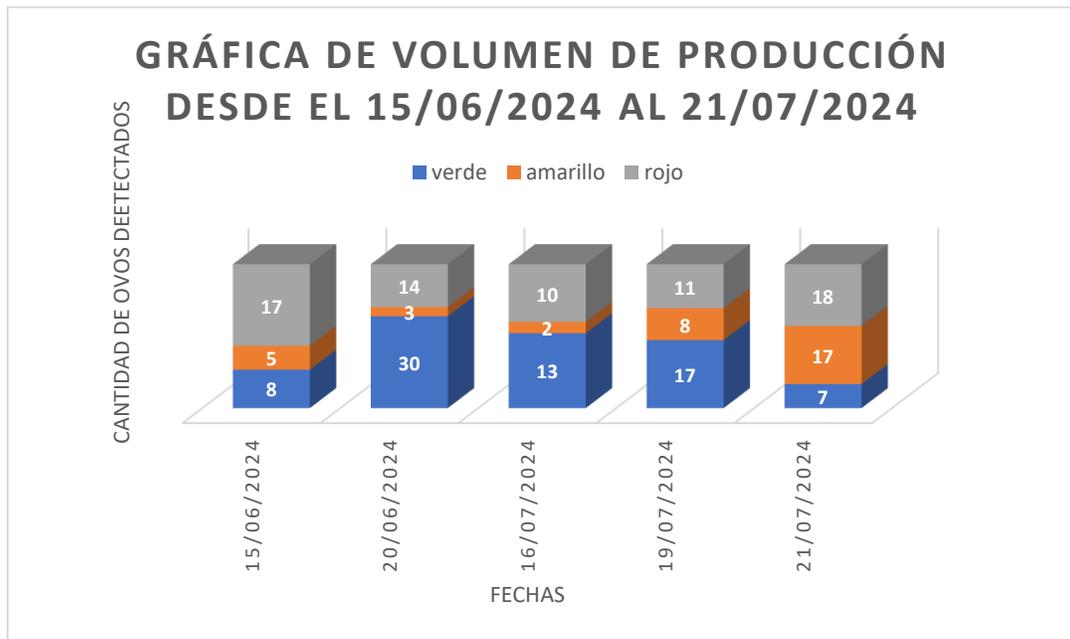


Figura 29 *Volumen de producción*

La Figura 29 muestra la proporción de cada categoría de maduración (rojos, verdes y amarillos) en relación con la cantidad total de ovos producidos. Cada segmento del gráfico

representa el número de ovos en cada categoría como un porcentaje del total general. Esta representación permite identificar rápidamente la participación relativa de cada categoría en la producción total, facilitando la evaluación del estado del inventario de manera eficiente.

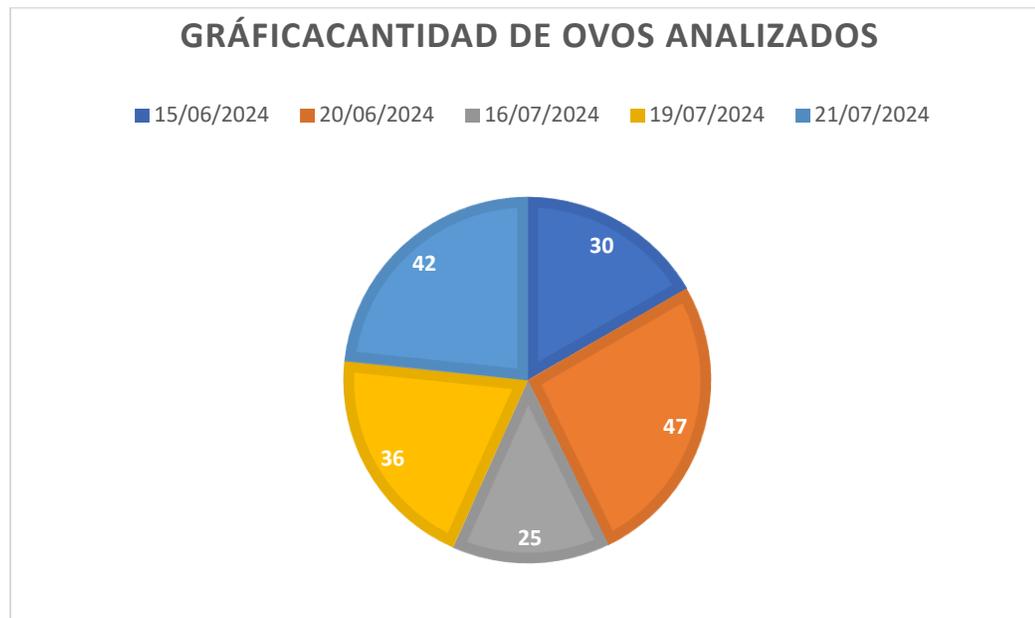


Figura 30 Cantidad de ovos analizados

La Figura 30 ilustra la distribución porcentual de los ovos clasificados en diferentes etapas de maduración: rojos, verdes y amarillos. Cada segmento del gráfico representa la proporción de ovos en cada categoría respecto al total de la producción. En día 16 de Junio se analizaron 30 ovos(azul), el 20 de Junio 47 ovos(naranja), 16 de Julio 25 ovos(gris), 19 de Julio 36 ovos(amarillo), 21 de Julio 42 ovos(celeste). Este gráfico proporciona una perspectiva clara de cómo se distribuye la producción entre las diferentes etapas de maduración, facilitando la comprensión de la composición del inventario de ovos en un formato visualmente intuitivo.

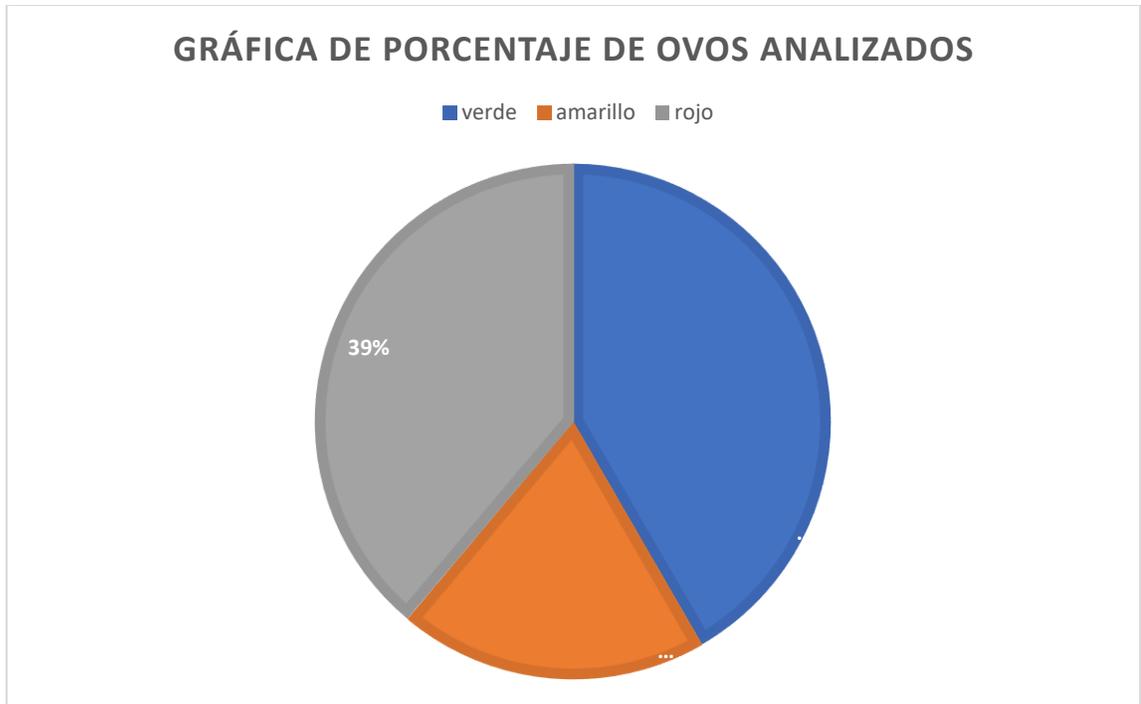


Figura 31 *Porcentaje de ovos analizados*

La Figura 31 expone la distribución porcentual de los ovos clasificados en distintas etapas de maduración: 39 % rojos, 42 % verdes y 15 % amarillos. Cada segmento del gráfico indica la proporción de ovos en cada categoría en relación con el total de la producción. Este gráfico ofrece una visión clara de cómo se distribuye la producción entre las diferentes etapas de maduración, lo que facilita la comprensión de la composición del inventario de ovos de manera visualmente intuitiva.

4.16 Limitaciones del Sistema

El proyecto enfrenta varias limitaciones que deben ser abordadas para mejorar su Efectividad y precisión:

- Escaneo de una imagen a la vez: Actualmente, solo se puede escanear una imagen a la vez, lo cual puede ser una limitación en términos de eficiencia.

- Requisitos de enfoque de imagen: La imagen debe estar bien enfocada, y el sistema no maneja adecuadamente los objetos desenfocados, lo que puede limitar su aplicabilidad en condiciones no ideales.
- Dependencia de computadoras y Python: El sistema solo funciona en computadoras y requiere Python para ejecutarse, lo cual puede restringir su uso en dispositivos sin esta configuración.
- Requisitos de Hardware: Puede depender de que el computador tenga una tarjeta gráfica buena para un análisis más rápido, lo que podría ser un obstáculo para su implementación en entornos con recursos limitados.
- Tamaño de la imagen: El tamaño de la imagen puede ser una limitación, ya que el sistema podría tener dificultades para procesar imágenes demasiado grandes o pequeñas de manera eficiente.
- Formato de la imagen: El formato de la imagen debe ser en JPG, lo cual puede ser restrictivo si se tienen imágenes en otros formatos que no se pueden convertir fácilmente.

4.17 Recomendaciones y líneas de trabajo futuro:

- Diversificación del Dataset: continuar ampliando el dataset con fotografías tomadas en una variedad de condiciones de iluminación, incluyendo días nublados, enriquecerá aún más el conjunto de datos. Esto permitirá entrenar un modelo aún más robusto y preciso, capaz de adaptarse eficazmente a diferentes escenarios de iluminación y ofrecer resultados consistentemente confiables.

- Optimización del Código y Algoritmos: mejorar aún más la eficiencia del código y los algoritmos a través de técnicas avanzadas de procesamiento, como la paralelización y vectorización, potenciará la capacidad del sistema para manejar grandes volúmenes de datos e imágenes en tiempo real. Implementar métodos de compresión y almacenamiento eficiente de datos también contribuirá a esta optimización.
- Adopción de Procesamiento en la Nube y Computación Distribuida: incorporar técnicas de procesamiento en la nube y computación distribuida permitirá al sistema escalar sin comprometer el rendimiento. Esto garantizará que el sistema pueda gestionar cargas de trabajo más grandes y complejas de manera efectiva y eficiente.
- Mantenimiento de Alta Calidad de Imágenes: continuar asegurando que las imágenes capturadas sean de alta calidad y estén bien enfocadas maximizará el rendimiento del modelo YOLOv8. Imágenes claras y libres de ruido visual permitirán una detección y clasificación de objetos más precisa, optimizando el rendimiento general del sistema.

CONCLUSIONES

A través del proceso de investigación y fundamentación teórica, se obtuvo información adecuada sobre visión artificial, redes neuronales, aprendizaje profundo y los diferentes estados de maduración. Esto permitió establecer los requisitos necesarios para alcanzar el objetivo principal del sistema de supervisión. La investigación también abarcó aspectos más específicos, como los hiperparámetros y los indicadores de maduración, que afectan directamente el rendimiento del sistema.

Cabe destacar que las redes neuronales iniciales resultaban demasiado exigentes en términos de recursos computacionales para ser aplicadas en la detección de objetos. Sin embargo, la aparición de las redes neuronales convolucionales marcó un hito importante en este campo, contribuyendo significativamente al éxito actual en la detección de objetos.

La interfaz gráfica de usuario creada, basada en algoritmos de procesamiento de imágenes y diseñada durante este trabajo de investigación, permite cargar imágenes del fruto en las ramas y monitorear la maduración de la *Spondias purpurea*.

Además de supervisar la maduración de la *Spondias purpurea*, el sistema también gestiona el volumen de producción. Esta funcionalidad facilita un seguimiento detallado de la cantidad de fruto producido, lo que garantiza una planificación más concreta y una mejor optimización del rendimiento del cultivo.

RECOMENDACIONES

Es de suma importancia tomar fotografías tanto en condiciones de buena luz como en días nublados. Esta práctica contribuye a aumentar la diversidad en el dataset. La inclusión de una variedad de condiciones de iluminación ayuda a que el modelo sea más adaptable y efectivo al enfrentarse a diferentes escenarios de iluminación en situaciones reales, mejorando su capacidad para generalizar y ofrecer resultados confiables en diversas circunstancias.

Para asegurar que el sistema basado en redes neuronales y YOLO pueda manejar grandes volúmenes de datos e imágenes en tiempo real, resulta crucial optimizar tanto el código como los algoritmos. Esto incluye mejorar la eficiencia del procesamiento mediante la paralelización y vectorización en la implementación de YOLO, y emplear métodos de compresión y almacenamiento eficiente de datos. Además, se pueden implementar técnicas de procesamiento en la nube y computación distribuida para aumentar la capacidad del sistema para escalar sin comprometer el rendimiento, garantizando que el sistema pueda manejar la carga de trabajo de manera efectiva.

Para garantizar un rendimiento óptimo en la detección de objetos utilizando YOLOv8, es esencial que las imágenes capturadas sean de alta calidad y estén bien enfocadas. Las imágenes de alta calidad permiten al modelo extraer características detalladas y precisas, mejorando la precisión en la detección y clasificación de objetos. Asegurarse de que las imágenes estén bien enfocadas y libres de ruido visual contribuirá significativamente a la efectividad del modelo, reduciendo la tasa de falsos positivos (FP) y falsos negativos (FN) y optimizando el rendimiento general del sistema de detección de objetos.

Finalmente, para mantener y mejorar la precisión del modelo, resulta indispensable realizar un monitoreo continuo del rendimiento y actualizar el modelo periódicamente con nuevos datos. Incorporar retroalimentación constante y ajustes basados en datos recientes garantizará que el modelo se mantenga efectivo y preciso con el tiempo, mejorando su capacidad para adaptarse a nuevas condiciones y escenarios.

REFERENCIAS

- [1] R. Monroy, A. Flores, C. Monroy-Ortiz, H. Bahena y R. Monroy-Ortiz, «Ciruela *Spondias purpurea* L. en los huertos frutícolas tradicionales de Morelos, México,» Universidad Autónoma del Estado de Morelos, México, 2020.
- [2] G. Vargas, «Ciruela / Mexican Plum — *Spondias Purpurea* L.,» de *Exotic Fruits*, Academic Press, 2018, p. 141–152.
- [3] L. Chamorro, B. Hernández, J. Mina y F. Torres, «Clasificación del ovo (*Spondias purpurea* L.) de Ambuquí – Ecuador en diferentes estados de maduración mediante determinación del color,» de *III Congreso Internacional de Ingenierías*, UPEC, 2017, pp. 14-20.
- [4] C. Valdivia, «Diseño de un sistema de visión artificial para la clasificación de chirimoyas basado en medidas,» Pontificia Universidad Católica del Perú, Lima, 2016.
- [5] M. Nixon y A. Aguado, *Feature extraction and image processing for computer vision*, Elsevier/Academic Press, 2020.
- [6] E. Tapia, «Sistema de alerta del estado de maduración de alimentos frescos dentro de un refrigerador utilizando inteligencia artificial,» Universidad Autónoma de Querétaro, Querétaro, 2023.
- [7] A. Aguilar y M. Campoverde, «Clasificación de frutas basadas en redes neuronales convolucionales,» *Polo del Conocimiento*, vol. 5, nº 1, 2020.
- [8] G. Ciaparrone, F. Luque, S. Tabik, L. Troiano, R. Tagliaferri y F. Herrera, «Deep learning in video multi-object tracking: a survey,» *Neurocomputing*, vol. 381, p. 61–88, 2020.

- [9] J. González y J. Parra, «Diseño e implementación de un sistema de reconocimiento de naranjas para el robot GIO 1 usando visión asistida por computador,» Universidad Católica de Ecuador, Quito, 2015.
- [10] A. Aguilar, «Diseño de un sistema clasificador de manzanas por color mediante visión artificial para la empresa Fresh & Natural C.I.,» Universidad Tecnológica Equinoccial, Quito, 2017.
- [11] B. Ortega, «Detección automática de malezas en plantaciones de maíz y/o papa utilizando imágenes adquiridas por dron y red neuronal Convolutiva Yolo,» Universidad Técnica del Norte, Ibarra, Ecuador, 2024.
- [12] C. Gershenson, «Artificial Neural Networks for Beginners,» 2012. [En línea]. Available: <https://www.uv.mx/mia/files/2012/10/Artificial-Neural-Networks-for-Beginners.pdf>.
- [13] A. Rodríguez, «El proyecto para crear una IA que supere al ser humano y nos salve de la muerte,» El Confidencial, 2021. [En línea]. Available: https://www.elconfidencial.com/tecnologia/2021-05-02/inteligencia-artificial-singularitynet-ben-goertzel_3059347/. [Último acceso: 26 julio 2024].
- [14] Y. Lecun, Y. Bengio y G. Hinton, «Deep learning,» *Nature*, vol. 521, n° 7553, p. 436–444, 2015.
- [15] Stanford University, « Ramas de la IA.,» 2017. [En línea]. Available: <https://mse238blog.stanford.edu/2017/08/jgokani/the-evolution-of-banking-ai/> Blog de MS&E 238.
- [16] P. Atencio, «Desarrollo de un sistema de visión artificial para el control de calidad del mango de azúcar de exportación,» Universidad del Magdalena, Santa Marta, 2009.

- [17] L. Luce, «Computer Vision and Smart Mirrors,» de *Artificial Intelligence for Fashion*, Berkeley, CA, Apress, 2019, p. 39–51.
- [18] C. De Jeu Boronat, «Estudio y comparación de las versiones de YOLO para la aplicación al tenis en tiempo real,» Universidad de Sevilla, Sevilla, 2022.
- [19] A. Krizhevsky, I. Sutskever y E. Hinton, «ImageNet classification with deep convolutional neural networks,» *Commun. ACM*, vol. 60, n° 6, p. 84–90, 2017.
- [20] I. Mohamed, «Detection and Tracking of Pallets using a Laser Rangefinder and Machine Learning Techniques.,» 2017. [En línea]. Available: https://www.researchgate.net/publication/324165524_Detection_and_Tracking_of_Pallets_using_a_Laser_Rangefinder_and_Machine_Learning_Techniqu.
- [21] X. Xiao y X. Feng, «Multi-Object Pedestrian Tracking Using Improved YOLOv8 and OC-SORT,» *Sensors*, vol. 23, n° 20, p. 8439, 2023.
- [22] X. Wang, L. Kong, Z. Zhang, H. Wang y X. Lu, «Keypoint regression strategy and angle loss based YOLO for object detection,» *Scientific Reports*, vol. 13, n° 1, p. 1–10, 2023 .
- [23] M. Escámez, «Clasificación automática de frutas a partir de imágenes con phyton. Técnicas de transferencia de aprendizaje a partir de conjuntos de datos pequeños,» Universidad Miguel Hernández de Elche, 2022.
- [24] R. Steyer, *Programmierung in Python: Ein kompakter Einstieg für die Praxis*, Springer Vieweg, 2018.
- [25] OpenCV, «OpenCV modules,» 2024. [En línea]. Available: <https://docs.opencv.org/4.x/>. [Último acceso: 24 julio 2024].

- [26] R. Steyer, *Programmierung in Python : Ein kompakter Einstieg für die Praxis*, Springer Vieweg, 2018.
- [27] P. Felzenszwalb, R. Girshick, D. McAllester y D. Ramanan, «Object detection with discriminatively trained part-based models,» *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, n° 9, p. 1627–1645, 2010.
- [28] F. Rodrigues, P. Giffony, S. dos Santos, J. Guedes, M. Ribeiro, T. de Araújo, L. da Silva y G. Zocolo, «Spondias purpurea L. Stem Bark Extract: Antioxidant and in vitro Photoprotective Activities,» *Journal of the Brazilian Chemical Society*, vol. 32, n° 10, p. 1918–1930, 2021.
- [29] E. Pozio, G. La Rosa, K. Murrell y J. Lichtenfels, «Taxonomic revision of the genus *Chaethippus* Duda, 1930 (Diptera: Chloropidae) with the description of four species,» *J. Parasitol.*, vol. 78, n° 4, p. 654–659, 2020.
- [30] A. Romero, A. Marín y J. Jiménez, «Sistema de clasificación por visión artificial de mangos tipo Tommy,» *Revista UIS Ingenierías*, vol. 14, n° 1, pp. 21-31, 2015.
- [31] D. Margüelles, «Detección en tiempo real de comida: Un enfoque innovador para la clasificación automatizada de alimentos con CNN,» Universidad Politécnica de Madrid, Madrid, 2023.
- [32] G. Figueredo-Ávila y A. Ballesteros-Ricaurte, «Identificación del estado de madurez de las frutas con redes neuronales artificiales, una revisión,» *Ciencia y Agricultura*, vol. 13, n° 1, pp. 117-132, 2016.

- [33] A. Balamurugan, S. Srinivasan, S. Mathivanan y M. Shah, «Robust brain tumor classification by fusion of deep learning and channel-wise attention mode approach,» *BMC Medical Imaging*, vol. 24, n° 1, p. 147, 2024.
- [34] D. Salem y M. Waleed, «Drowsiness detection in real-time via convolutional neural networks and transfer learning,» *J. Eng. Appl. Sci.* , vol. 71, n° 122, 2024.
- [35] Ultralytics, «YOLOv8,» 2024. [En línea]. Available: <https://docs.ultralytics.com/models/yolov8/#performance-metrics>. [Último acceso: 24 julio 2024].
- [36] J. Howse y J. Minichino, *Learning OpenCV 4 Computer Vision with Python 3*, Packt Publishing, 2020.
- [37] C. Widodo, K. Adi y R. Gernowo, «Medical image processing using python and OpenCV,» *Journal of Physics: Conference Series*, vol. 1524, n° 1, p. 8–12, 2020.
- [38] A. Krizhevsky, I. Sutskever y G. Hinton, «ImageNet classification with deep convolutional neural networks,» *Commun ACM*, vol. 60, n° 6, p. 84–90, 2017.
- [39] J. Marron, J. Ramsay, L. Sangalli y A. Srivastava, «Functional data analysis of amplitude and phase variation,» *Statistical Science*, vol. 30, n° 4, p. 468–484, 2015.
- [40] V. Maslej-Krešňáková, A. Kandrát, Š. Mackovjak, P. Butka, S. Jaščur, I. Kolmašová y O. Santolík, «Automatic Detection of Atmospheric and Tweek Atmospherics in Radio Spectrograms Based on a Deep Learning Approach,» *Earth and Space Science*, vol. 8, n° 11, p. e2021EA002007, 2021.

- [41] A. Bochkovskiy, C. Wang y H. Liao, «YOLOv4: Optimal Speed and Accuracy of Object Detection,» 2020. [En línea]. Available: <https://arxiv.org/abs/2004.10934v1>. [Último acceso: 24 julio 2024].
- [42] A. Nanduri, R. Neelapu, B. D. V. Maddumala y K. Gowrisankar, «Image Recolorization With Convolutional Neural Network Using Opencv,» *Journal of Natural Remedies*, vol. 21, n° 4, p. 81–87, 2020.
- [43] Ultralytics, «Hyperparameter Tuning,» 2024. [En línea]. Available: <https://docs.ultralytics.com/guides/hyperparameter-tuning/#steps-involved>. [Último acceso: 24 julio 2024].
- [44] V. Prajwal, D. Venkatesh y Nagaraj, «Image processing system for automatic segmentation for strawberries and other fruits using OpenCV,» *International Research Journal of Modernization in Engineering Technology and Science*, vol. 4, n° 1, p. 285–291, 2022.
- [45] K. Brown, J. Smith, M. Wilson y R. Johnson, «Improving construction project management through artificial intelligence,» *Journal of Construction Management and Economics*, 2020.
- [46] “YOLOv8 - Ultralytics YOLO Docs.” Accessed: Jul. 25, 2024. [Online]. Available: <https://docs.ultralytics.com/models/yolov8/#performance-metrics>
- [47] “OpenCV: OpenCV modules.” Accessed: Jul. 25, 2024. [Online]. Available: <https://docs.opencv.org/4.x/>
- [48] “ISO 25010.” Accessed: Jul. 29, 2024. [Online]. Available: <https://iso25000.com/index.php/normas-iso-25000/iso-25010>

ANEXOS

El código de programación desarrollado para realizar este proyecto está guardado en la siguiente carpeta de Google Drive: <https://goo.su/veyA9>

NOMBRE DEL TRABAJO

completo_Tesis_AlejandraChuquin 26-07-2024.docx

AUTOR

Alejandra Chuquín

RECUENTO DE PALABRAS

13742 Words

RECUENTO DE CARACTERES

77018 Characters

RECUENTO DE PÁGINAS

88 Pages

TAMAÑO DEL ARCHIVO

5.0MB

FECHA DE ENTREGA

Jul 26, 2024 4:03 PM GMT-5

FECHA DEL INFORME

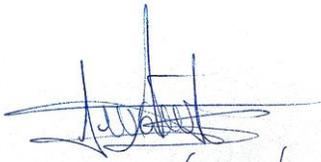
Jul 26, 2024 4:04 PM GMT-5**● 8% de similitud general**

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para cada base de datos.

- 5% Base de datos de Internet
- Base de datos de Crossref
- 6% Base de datos de trabajos entregados
- 1% Base de datos de publicaciones
- Base de datos de contenido publicado de Crossref

● Excluir del Reporte de Similitud

- Material bibliográfico
- Material citado
- Fuentes excluidas manualmente
- Material citado
- Coincidencia baja (menos de 10 palabras)



Ing. Juan Gerardo.