



Universidad  
Nacional  
de Loja

## **Universidad Nacional de Loja**

### **Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables**

#### **Carrera de Ingeniería en Sistemas**

**Modelo predictivo basado en deep learning para identificar la Mancha Angular (*Xanthomonas fragariae*) en la hoja de Fresa para el Municipio del cantón Saraguro**

**Predictive model based on deep learning to identify the Angular Spot (*Xanthomonas fragariae*) in the Strawberry leaf for the Municipality of Saraguro canton.**

Trabajo de Titulación, previo a la obtención del título de Ingeniera en Sistemas.

#### **AUTORA:**

Diana Gabriela González Chillogalli

#### **DIRECTORA:**

Ing. Genoveva Jackelinne Suing Albitto, Mg. Sc.

**Loja – Ecuador**

**2025**

## **Certificación**

**Loja, 29 de enero del 2025**

Ing. Genoveva Jackelinne Suing Albitto, Mgs.

**DIRECTORA DEL TRABAJO DE TITULACIÓN**

### **CERTIFICO:**

Que he revisado y orientado todo el proceso de la elaboración del Trabajo de Titulación denominado: **Modelo predictivo basado en deep learning para identificar la Mancha Angular (*Xanthomonas fragariae*) en la hoja de Fresa para el Municipio del cantón Saraguro**, previo a la obtención del título de **Ingeniera en Sistemas**, de autoría del estudiante: **Diana Gabriela González Chillogalli** con cédula de identidad Nro. **1150036984**, una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja, para el efecto, autorizo la presentación del mismo para su respectiva sustentación y defensa.

Ing. Genoveva Jackelinne Suing Albitto. Mgs.

**DIRECTORA DEL TRABAJO DE TITULACIÓN**

## **Autoría**

Yo, **Diana Gabriela González Chillogalli**, declaro ser autora del presente Trabajo de Titulación y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos y acciones legales, por el contenido del mismo. Adicionalmente acepto y autorizo a la Universidad Nacional de Loja la publicación de mi Trabajo de Titulación en el Repositorio Digital Institucional – Biblioteca Virtual.

**Firma:**

**Cédula de Identidad:** 1150036984

**Fecha:** 29/01/2025

**Correo electrónico:** dggonzalesch@unl.edu.ec

**Teléfono:** 0999652485

**Carta de autorización por parte de la autora, para la consulta, reproducción parcial y/o total, publicación electrónica de texto completo del Trabajo de Titulación**

Yo, **Diana Gabriela González Chillogalli**, declaro ser autora del Trabajo de Titulación denominado: **Modelo predictivo basado en deep learning para identificar la Mancha Angular(*Xanthomonas fragariae*) en la hoja de Fresa para el Municipio del cantón Saraguro**; como requisito para optar el título de **Ingeniera en Sistemas**; autorizo al sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos muestre la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Institucional, en las redes de información del país y del exterior, con los cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del Trabajo de Titulación que realice un tercero.

Para constancia de esta autorización en la ciudad de Loja, a los veintinueve días del mes de enero del dos mil veinticinco.

**Firma:**

**Autor:** Diana Gabriela González Chillogalli

**Cédula:** 1150036984

**Dirección:** Saraguro

**Correo electrónico:** dggonzalesch@unl.edu.ec

**Celular:** 0 999 652 485

**DATOS COMPLEMENTARIOS:**

**Directora del Trabajo de Titulación:** Ing. Genoveva Jackelinne Suing Albito, Mg. Sc

## **Dedicatoria**

Este trabajo de titulación es dedicado con todo mi cariño y gratitud a las personas que han sido mi soporte, inspiración y razón para alcanzar esta meta

A mi amado esposo Álvaro, por su sacrificio, esfuerzo y por creer en mí incluso en los momentos más difíciles. Gracias por tu amor incondicional, tu comprensión y por ser mi compañero en cada paso de este camino.

A mis preciosos hijos, Samuel y Jhonas, quienes han sido mi mayor motivación e inspiración para superarme y demostrarles que con perseverancia se pueden alcanzar los sueños.

A mi linda mamita Paula, quien siempre me brindó su apoyo incondicional y con sus oraciones y palabras de aliento, no me dejó quedar a mitad del camino, sino que siempre me animaba a ser perseverante.

A mi querido hermano César y a su esposa, por su cariño, paciencia y por el enorme esfuerzo que hicieron para ayudarme a construir mi carrera. Sus consejos y enseñanzas me han hecho valorar cada logro y superar cada obstáculo. Sin ustedes, esto no hubiera sido posible.

A mis hermanos, Cristhian, Daniela y Roberth, por su apoyo, por estar presentes en cada etapa de mi vida, por su confianza y por cada esfuerzo que han hecho para que yo termine esta carrera.

A mi papá, quien, aunque ya no está físicamente conmigo, me dejó un legado invaluable: el ejemplo de responsabilidad, esfuerzo y, sobre todo, un profundo amor por la familia

A toda mi familia, quienes de una u otra manera pusieron un granito de arena para que pudiera alcanzar este objetivo académico.

***Diana Gabriela González Chillogalli***

## **Agradecimiento**

Agradezco primeramente a mi Dios, pues por su plenitud he recibido gracia sobre gracia, ha sido mi escudo y fortaleza durante toda mi vida.

Expreso mi gratitud a toda mi familia quienes me han acompañado durante mi formación académica con sus oraciones, consejos, palabras de ánimo y apoyo, en especial a mi madre, mi esposo, mis hijos y mis hermanos.

De la misma manera, extendo mi agradecimiento a mi tutora de Trabajo de Titulación, la Ing. Genoveva Jackeline Suing Albito, Mg. Sc., por su apoyo incondicional, su asesoría y enseñanzas, que me permitieron culminar esta investigación .

***Diana Gabriela González Chillogalli***

## Índice de contenidos

<b>Portada</b> .....	<b>i</b>
<b>Certificación</b> .....	<b>ii</b>
<b>Autoría</b> .....	<b>iii</b>
<b>Dedicatoria</b> .....	<b>v</b>
<b>Agradecimiento</b> .....	<b>vi</b>
<b>Índice de contenidos</b> .....	<b>vii</b>
Índice de tablas .....	x
Índice de figuras .....	xii
Índice de anexos .....	xv
<b>1. Título</b> .....	<b>1</b>
<b>2. Resumen</b> .....	<b>2</b>
<b>Abstract</b> .....	<b>3</b>
<b>3. Introducción</b> .....	<b>4</b>
<b>4. Marco teórico</b> .....	<b>6</b>
4.1 Antecedentes .....	6
4.1.1 Escuela Agroecológica de Saraguro .....	6
4.1.2 Mancha angular .....	6
4.1.3 Síntomas y daños asociados a la mancha angular en la fresa .....	7
4.2 Fundamentos teóricos .....	8
4.2.1 Inteligencia Artificial (IA) .....	8
4.2.2 Machine Learning .....	8
4.2.3 Deep Learning .....	8
4.2.4 Visión por computadora .....	9
4.2.5 Redes Neuronales Convolucionales .....	9
4.2.6 YOLO (You Only Look Once) .....	10
4.2.7 Yolov5 .....	11
4.2.8 Yolov7 .....	12
4.2.9 Yolov8 .....	13
4.2.10 Faster R-CNN .....	15

4.2.11	Metodología CRISP-DM.....	17
4.2.11.1	Fases de la metodología CRISP-DM.....	18
4.2.12	Colab.....	19
4.2.13	Zero-Shot Learning.....	19
4.2.14	Matriz de Confusión.....	20
4.2.15	Métricas de la matriz de confusión.....	21
4.2.16	Dataset strawberry diseases.....	22
4.3	Herramientas de etiquetado de imágenes.....	23
4.3.1	LabelImg.....	23
4.3.2	LabelMe The Open Annotation Tool.....	23
4.3.3	Roboflow.....	23
4.3.4	Comparativa de las herramientas de etiquetado.....	24
4.4	Aplicación Móvil.....	24
4.4.1	Metodología XP (Programación Extrema).....	24
4.4.2	Historias de Usuario.....	25
4.4.3	Roles de la Metodología XP.....	26
4.4.4	Fases de la Programación Extrema.....	27
4.5	Flutter.....	28
4.6	Trabajos Relacionados.....	29
<b>5.</b>	<b>Metodología.....</b>	<b>35</b>
5.1	Área de estudio.....	35
5.2	Procedimiento.....	36
5.2.1	Implementar un modelo de red neuronal convolucional y el conjunto de datos necesarios para la detección de la Mancha Angular en la fresa.....	36
5.2.2	Construir un prototipo de software e integrar el modelo de red neuronal utilizando la metodología XP.....	37
5.3	Recursos.....	37
5.3.1	Recursos científicos.....	37
5.3.2	Recursos técnicos.....	38
5.4	Participantes.....	39
<b>6.</b>	<b>Resultados.....</b>	<b>40</b>



6.1	Objetivo 1: Implementar un modelo de red neuronal convolucional y el conjunto de datos necesarios para la detección de la Mancha Angular en la fresa. ....	40
6.1.2	Fase 1: Entendimiento de los datos .....	43
6.1.3	Fase 2: Preparación de datos .....	46
6.1.4	Fase 3: Modelado .....	52
6.1.5	Fase 4: Evaluación de los modelos.....	67
6.2	Objetivo 2: Construir un prototipo de software e integrar el modelo de red neuronal utilizando la metodología XP. ....	78
6.2.1	Fase 5: Implementación.....	78
6.2.2	Metodología de Desarrollo del prototipo de aplicación móvil.....	79
6.2.3	Actores.....	79
6.2.4	Planificación.....	80
6.2.5	Diseño.....	83
6.2.6	Codificación .....	85
6.2.7	Pruebas .....	91
<b>7.</b>	<b>Discusión.....</b>	<b>94</b>
<b>8.</b>	<b>Conclusiones.....</b>	<b>97</b>
<b>9.</b>	<b>Recomendaciones.....</b>	<b>98</b>
<b>10.</b>	<b>Bibliografía.....</b>	<b>99</b>
<b>11.</b>	<b>Anexos.....</b>	<b>104</b>

## Índice de tablas

<b>Tabla 1.</b> Estructura de la Matriz de Confusión .....	20
<b>Tabla 2.</b> Resumen conjunto de Datos de Kaggle .....	22
<b>Tabla 3.</b> Tabla Comparativa de las herramientas de etiquetado .....	24
<b>Tabla 4.</b> Plantilla de Historia de Usuario .....	26
<b>Tabla 5.</b> Trabajos Relacionados.....	29
<b>Tabla 6.</b> Fases que comprenden el proceso de la presente RSL .....	40
<b>Tabla 7.</b> Bases de datos científicas utilizadas para la búsqueda de estudios primarios .....	41
<b>Tabla 8.</b> Análisis establecidos para la elección de investigaciones definitivas en la etapa 2 de la revisión sistemática de la literatura (RSL) .....	41
<b>Tabla 9.</b> Recursos empleados en el proceso de adquisición de datos .....	44
<b>Tabla 10.</b> Parámetros para el aumento de datos.....	47
<b>Tabla 11.</b> Conjunto de datos .....	49
<b>Tabla 12.</b> Formato de etiqueta para los modelos Faster RCNN, Yolo v5, v7 y v8.....	50
<b>Tabla 13.</b> Métricas establecidas por cada modelo identificado en los trabajos analizados (TA) de la RSL ....	53
<b>Tabla 14.</b> Parámetros de entrenamiento.....	55
<b>Tabla 15.</b> Resultados de entrenamiento de los modelos de detección .....	70
<b>Tabla 16.</b> Comparación de los resultados de predicción de los modelos implementados en el conjunto de datos de prueba .....	72
<b>Tabla 17.</b> Conjunto de datos con dos clases (DATASET-2) .....	74
<b>Tabla 18.</b> Comparación de métricas entre el modelo Yolov5 con una clase y Yolov5 con dos clases .....	77
<b>Tabla 19.</b> Descripción de los roles que son parte del desarrollo del TT .....	80
<b>Tabla 20.</b> Historia de Usuario N°01 – Integrar modelo entrenado .....	80
<b>Tabla 21.</b> Historia de Usuario N°02 – Información principal.....	81
<b>Tabla 22.</b> Historia de Usuario N°03 – Seleccionar foto de Galería .....	81
<b>Tabla 23.</b> Historia de Usuario N° 04 – Capturar foto con la cámara .....	81
<b>Tabla 24.</b> Historia de Usuario N° 05 – Mostrar Resultado .....	82
<b>Tabla 25.</b> Valores de priorización de las Historias de Usuario .....	82
<b>Tabla 26.</b> Estimación e iteración de HU .....	83
<b>Tabla 27.</b> Recursos software.....	85
<b>Tabla 28.</b> Pantalla de Inicio .....	89
<b>Tabla 29.</b> Información de enfermedad.....	89
<b>Tabla 30.</b> Pantalla con selección de cámara .....	90
<b>Tabla 31.</b> Pantalla con selección de Galería .....	90
<b>Tabla 32.</b> Pantalla con resultados de detección .....	91
<b>Tabla 33.</b> Resultados de métricas de evaluación final del modelo integrado en el prototipo móvil .....	93
<b>Tabla 34.</b> Fases que comprenden el proceso de RSL basado en la metodología de Bárbara Kitchenham ...	105

<b>Tabla 35.</b> Preguntas de Investigación .....	105
<b>Tabla 36.</b> Palabras claves en el método PICOC .....	106
<b>Tabla 37.</b> Fuentes de búsqueda.....	107
<b>Tabla 38.</b> Cadenas de búsqueda por cada base de datos científica .....	107
<b>Tabla 39.</b> Criterios de Inclusión.....	108
<b>Tabla 40.</b> Criterios de exclusión .....	108
<b>Tabla 41.</b> Resultados de Búsqueda .....	109
<b>Tabla 42.</b> Número de artículos aceptados, descartados y duplicados resultantes del análisis uno.....	110
<b>Tabla 43.</b> Número de artículos aceptados, descartados y duplicados resultantes del análisis dos .....	110
<b>Tabla 44.</b> Número de artículos aceptados, descartados y duplicados resultantes del análisis tres.....	111
<b>Tabla 45.</b> Trabajos aprobados (TA) obtenidos durante la fase de selección de estudios .....	111
<b>Tabla 46.</b> Identificación de Modelos CNN en los TA .....	114
<b>Tabla 47.</b> Métricas para la evaluación de modelos CNN.....	118
<b>Tabla 48.</b> Número de imágenes de la hoja de fresa con mancha angular capturadas en áreas de estudio .....	127
<b>Tabla 49.</b> Comparativa de resultados de prueba del modeloYolov5 integrado en la App móvil y el experto	137

## Índice de figuras

<b>Figura 1.</b> Enfermedad de la mancha angular en la hoja de fresa .....	7
<b>Figura 2.</b> Proceso del funcionamiento de una Red Neuronal Convolutacional .....	9
<b>Figura 3.</b> Arquitectura Yolo .....	11
<b>Figura 4.</b> Arquitectura Yolov5 .....	11
<b>Figura 5.</b> Arquitectura Yolov7 .....	13
<b>Figura 6.</b> Arquitectura Yolov8 .....	14
<b>Figura 7.</b> Arquitectura de Faster R-CNN.....	15
<b>Figura 8.</b> Mapa de características de salida de la imagen después de pasarlas a través de la columna vertebral (Red base).....	16
<b>Figura 9.</b> Cuadros de anclaje en el espacio de la imagen .....	16
<b>Figura 10.</b> Etiquetado objetivo de un ROI.....	17
<b>Figura 11.</b> Área experimental 1 y 2 – Mapa de la Comunidad La Matara y Barrio Puente Chico .....	35
<b>Figura 12.</b> Área experimental 3 – Mapa Parroquia El Tablón.....	36
<b>Figura 13.</b> Número de artículos encontrados en el proceso de selección de estudios primarios .....	41
<b>Figura 14.</b> Gráfica que muestra la cantidad de artículos admitidos, rechazados y duplicados como resultado de los análisis realizados.....	42
<b>Figura 15.</b> Resultado de los modelos CNN encontrados en los estudios primarios para la identificación de enfermedades foliares .....	43
<b>Figura 16.</b> Áreas experimentales de cultivo de fresa en Saraguro .....	44
<b>Figura 17.</b> Conjunto de imágenes de mancha angular en la hoja de fresa propias .....	45
<b>Figura 18.</b> Información de imágenes propias .....	45
<b>Figura 19.</b> Conjunto de imágenes del repositorio de Kaggle.....	46
<b>Figura 20.</b> Información imágenes de Kaggle.....	46
<b>Figura 21.</b> Sección del código para aplicar técnicas de aumento de datos .....	48
<b>Figura 22.</b> Etiquetado de la enfermedad en Roboflow .....	48
<b>Figura 23.</b> Conjunto de datos etiquetado .....	49
<b>Figura 24.</b> Etiquetas de las imágenes en el formato Yolo .....	50
<b>Figura 25.</b> Estructura del conjunto de datos en Yolo.....	51
<b>Figura 26.</b> Etiquetas de las imágenes en formato Faster RCNN .....	51
<b>Figura 27.</b> Estructura del conjunto de datos en Faster RCNN ( <i>Formato COCO</i> ) .....	52
<b>Figura 28.</b> División del conjunto de datos DATASET-1 .....	52
<b>Figura 29.</b> Modelos de detección con mejores resultados de precisión según la RSL.....	54
<b>Figura 32.</b> Arquitectura de Faster R-CNN [38] .....	56
<b>Figura 33.</b> Instalación de la biblioteca Detectron2 y sus dependencias .....	56
<b>Figura 34.</b> Adquisición del dataset con Roboflow en el formato COCO .....	57
<b>Figura 35.</b> Código de registro de conjunto de datos en Detectron2.....	57

<b>Figura 36.</b> Código de preparación de datos para ser utilizado por Detectron2 .....	57
<b>Figura 37.</b> Subclase para ajustar la evaluación al modelo Faster R-CNN .....	58
<b>Figura 38.</b> Modelo entrenado Faster R-CNN .....	58
<b>Figura 39.</b> Gráfico de diagrama de capas de Yolov5 entrenado .....	59
<b>Figura 40.</b> Líneas de código ejecutadas para la descarga del modelo pre entrenado Yolov5 .....	59
<b>Figura 41.</b> Código para importar el dataset desde Roboflow .....	60
<b>Figura 42.</b> Estructura del archivo <i>data.yaml</i> .....	60
<b>Figura 43.</b> Configuración del repositorio de Yolov5.....	61
<b>Figura 44.</b> Código para entrenar el modelo de Yolov5 .....	61
<b>Figura 45.</b> Archivos de resultados del modelo Yolov5 .....	62
<b>Figura 46.</b> Gráfico de capas del modelo Yolov7 .....	63
<b>Figura 47.</b> Repositorio de yolov7 en Github .....	63
<b>Figura 48.</b> Instalación de requerimientos para yolov7.....	63
<b>Figura 49.</b> Importación de Datos desde Roboflow en formato yolov7 .....	64
<b>Figura 50.</b> Código de descarga de los pesos pre entrenados del modelo Yolov7(yolov7_training.pt) .....	64
<b>Figura 51</b> Línea de código para el entrenamiento del modelo Yolov7 .....	64
<b>Figura 52.</b> Archivos resultantes del entrenamiento del modelo Yolov7.....	65
<b>Figura 53.</b> Diagrama de capas de Yolov8.....	65
<b>Figura 54.</b> Código de instalación de la librería Ultralytics .....	66
<b>Figura 55.</b> Código de importación de Dataset de roboflow en formato Yolov8.....	66
<b>Figura 56.</b> Código de entrenamiento del modelo Yolov8.....	66
<b>Figura 57.</b> Archivos resultantes del entrenamiento del modelo Yolov8.....	67
<b>Figura 58.</b> Matriz de confusión del modelo Yolov5.....	68
<b>Figura 59.</b> Matriz de confusión del modelo Yolov7.....	69
<b>Figura 60.</b> Matriz de confusión del modelo Yolov8.....	69
<b>Figura 61.</b> Predicciones Yolov5 .....	71
<b>Figura 62.</b> Predicciones Yolov7 .....	71
<b>Figura 63.</b> Predicciones YOLOv8 .....	72
<b>Figura 64.</b> Predicciones con Faster R-CNN .....	72
<b>Figura 65.</b> Curva de precisión de los resultados en la clase “sana” y “m_angular” .....	74
<b>Figura 66.</b> Curva de confianza de la sensibilidad de los resultados en la clase “sana” y “m_angular” .....	75
<b>Figura 67.</b> Curva de precisión- sensibilidad de los resultados en la clase “sana” y “m_angular” .....	75
<b>Figura 68.</b> Matriz de confusión con la clase "sana" y "m_angular". .....	76
<b>Figura 69.</b> Predicciones realizadas por el modelo entrenado con dos clases. a) Etiquetas originales b) Predicciones realizadas por el modelo.....	77
<b>Figura 70.</b> Estructura de modelo Yolov5 entrenado.....	78
<b>Figura 71.</b> Fases de la metodología XP aplicadas en la investigación.....	79
<b>Figura 72.</b> Roles de la metodología XP que son parte del TT .....	79

<b>Figura 73.</b> Diseño del Prototipo móvil .....	83
<b>Figura 74.</b> Interfaces de ingreso de imágenes.....	84
<b>Figura 75.</b> Interfaz de vista de resultados .....	84
<b>Figura 76.</b> Conversión del modelo al formato torchscript.....	85
<b>Figura 77.</b> Pesos del modelo entrenado y convertido a torchscript .....	85
<b>Figura 78.</b> Integración del modelo entrenado a la App móvil .....	86
<b>Figura 79.</b> Integración del modelo en la app .....	86
<b>Figura 80.</b> Módulo de interfaz gráfica pantalla principal .....	87
<b>Figura 81.</b> Método de importación de imagen para la detección.....	88
<b>Figura 82.</b> Código para la detección de la enfermedad .....	88
<b>Figura 83.</b> Matriz de confusión de prueba del modelo en la fase de pruebas .....	92
<b>Figura 84.</b> Detecciones realizadas por la aplicación móvil .....	93
<b>Figura 85.</b> Proceso de la metodología de Bárbara Kitchenham.....	104
<b>Figura 84.</b> Resultado de los modelos CNN aplicados en los estudios primarios .....	117
<b>Figura 85.</b> Modelos con mejores precisiones según la RSL.....	118
<b>Figura 86.</b> Área de cultivo de fresa al aire libre en la comunidad de La Matara .....	128
<b>Figura 87.</b> Conjunto de datos adquirido en el cultivo de fresa de la comunidad La Matara.....	128
<b>Figura 88.</b> Invernadero de cultivo de fresa en la parroquia El Tablón .....	129
<b>Figura 91.</b> Captura de fotografías de las hojas enfermas en el cultivo de fresa de la parroquia El Tablón ...	129
<b>Figura 92.</b> Captura de fotografías de las hojas enfermas en el cultivo de fresa Cañicapac .....	130
<b>Figura 93.</b> Conjunto de datos recopilado en el invernadero de la parroquia El Tablón.....	130
<b>Figura 94.</b> Conjunto de Datos adquirido en la comunidad de Puente Chico .....	131
<b>Figura 95.</b> Conjunto de Datos adquirido en la comunidad de Cañicapac .....	131
<b>Figura 96.</b> Prueba de aceptación Historia de Usuario N° 02 .....	132
<b>Figura 97.</b> Prueba de aceptación Historia de Usuario N° 03 y 04 .....	133
<b>Figura 98.</b> Prueba de aceptación Historia de Usuario N° 05 .....	134
<b>Figura 99.</b> Comparación de resultados con el conjunto de prueba en la fase de validación.....	135
<b>Figura 100.</b> Evidencia de la práctica de prueba de la aplicación móvil en la comunidad de Tuchin.....	139

## Índice de anexos

<b>Anexo 1.</b> Revisión Sistemática de Literatura .....	104
<b>Anexo 2.</b> Ensayo de adquisición del conjunto de datos mediante la captura de fotografía tomadas en huertos del cantón Saraguro. ....	126
<b>Anexo 3.</b> Pruebas de Aceptación de las interfaces del prototipo móvil .....	132
<b>Anexo 4.</b> Resultados de la fase de evaluación de los modelos Yolov5, Yolov7, Yolov8 y Faster R-CNN con el conjunto de datos de pruebas.....	135
<b>Anexo 5.</b> Ensayo de pruebas de la aplicación móvil bajo supervisión del experto en un entorno real.....	136
<b>Anexo 6.</b> Manual de Usuario.....	140
<b>Anexo 7.</b> Certificado de Data Set otorgado por el Técnico de Apoyo Agropecuario del GADMIS .....	150
<b>Anexo 8.</b> Certificado de pruebas de validación de la App Móvil.....	151
<b>Anexo 9.</b> Certificado de traducción del resumen al idioma inglés .....	152

## **1. Título**

**Modelo predictivo basado en deep learning para identificar la Mancha Angular (*Xanthomonas fragariae*) en la hoja de Fresa para el Municipio del cantón Saraguro**

**Predictive model based on deep learning to identify the Angular Spot (*Xanthomonas fragariae*) in the Strawberry leaf for the Municipality of Saraguro canton.**



## 2. Resumen

El cultivo de fresa es una actividad agrícola común en el cantón de Saraguro. Sin embargo, esta plantación se ha visto afectada significativamente por la bacteria *Xanthomonas fragariae*, causante de la Mancha Angular, la cual, puede infectar rápidamente gran parte del cultivo, especialmente aquellas al aire libre, provocando baja calidad de producción y significativas pérdidas económicas. El propósito de este estudio fue implementar un modelo basado en deep learning para detectar la Mancha Angular en la hoja de fresa. La metodología adoptada se basó en el Proceso Estándar de la Industria Transversal para el Aprendizaje Automático con Garantía de Calidad (CRISP-ML(Q)) adaptando cinco fases: entendimiento de datos, preparación de datos, modelado, evaluación e implementación. En la primera y segunda fase se recolectaron imágenes en diferentes sectores del cantón Saraguro como: La Matara, Tablón, Puente Chico y Cañicapac, donde existen huertos e invernaderos con cultivos de fresa, además, se añadieron imágenes disponibles en Kaggle, conformando un conjunto de 2987 imágenes que incluyen hojas sanas y afectadas por Mancha Angular. Durante la tercera y cuarta fase, se implementaron y compararon los modelos de clasificación y detección de objetos basados en deep learning Yolov5, Yolov7, Yolov8 y Faster R-CNN. Estos fueron entrenados únicamente con datos de la clase "mancha angular", obteniendo precisiones de 93%, 89%, 92% y 82%, respectivamente, destacando Yolov5 como el modelo más preciso. Posteriormente al integrar un conjunto de datos complementarios con imágenes de hojas sanas en el entrenamiento, se logró una mejora leve pero significativa en la precisión de Yolov5, alcanzando un 94%. En la fase final, se desarrolló un prototipo de aplicación móvil utilizando la metodología XP y el framework Flutter para integrar el modelo entrenado. Las pruebas finales, realizadas en colaboración con el Técnico de Desarrollo Agrícola del GAD municipal, alcanzaron una precisión del 96% en un entorno real. Estos resultados demuestran que el modelo propuesto es eficaz para detectar la mancha angular en las hojas de fresa, ofreciendo una valiosa herramienta para los pequeños agricultores del cantón de Saraguro.

**Palabras clave:** Aprendizaje profundo, CRISP-ML(Q), Mancha Angular, Yolov5

## Abstract

Strawberry cultivation is a common agricultural activity in the canton of Saraguro. However, this crop has been significantly affected by the bacterium *Xanthomonas fragariae*, the causative agent of Angular Leaf Spot. This disease can rapidly infect large quantities of plants, particularly in open-field cultivation, leading to reduced production quality and significant economic losses. The objective of this study was to implement a deep learning-based model to detect Angular Leaf Spot in strawberry leaves. The methodology adopted was based on the Cross-Industry Standard Process for Machine Learning with Quality Assurance (CRISP-ML(Q)), adapting five phases: data understanding, data preparation, modeling, evaluation, and implementation. During the first and second phases, images were collected from various sectors of the canton of Saraguro, including La Matara, Tablón, Puente Chico, and Cañicapac, where strawberry crops are cultivated in both orchards and greenhouses. Additionally, images available on Kaggle were incorporated, resulting in a dataset of 2987 images, including both healthy leaves and those affected by Angular Leaf Spot. In the third and fourth phases, deep learning-based classification and object detection models, namely YOLOv5, YOLOv7, YOLOv8, and Faster R-CNN, were implemented and compared. These models were trained exclusively on data from the "Angular Leaf Spot" class, achieving accuracies of 93%, 89%, 92%, and 82%, respectively, with YOLOv5 emerging as the most accurate model. Subsequently, by integrating a complementary dataset containing images of healthy leaves into the training process, a slight but significant improvement in YOLOv5's accuracy was achieved, reaching 94%. In the final phase, a mobile application prototype was developed using the XP methodology and the Flutter framework to integrate the trained model. Final testing, conducted in collaboration with the Agricultural Development Technician of the municipal GAD, achieved an accuracy of 96% in a real-world environment. These results demonstrate that the proposed model is effective in detecting Angular Leaf Spot in strawberry leaves, providing a valuable tool for small-scale farmers in the canton of Saraguro

**Keywords:** *Angular leaf spot, CRISP-ML(Q), Deep learning, Yolov5*

### 3. Introducción

La producción de fresas es una parte fundamental de la industria agrícola en muchas regiones del mundo debido a su alta demanda en el mercado nacional e internacional [1], gracias a su fácil adaptación a diferentes climas y tipos de suelo es cultivada en casi todo el mundo. Sin embargo, una amenaza constante para este cultivo es la presencia de enfermedades, especialmente la mancha angular, la cual es causada por la bacteria *Xanthomonas fragariae* [2] que afecta directamente a las hojas, transportándose rápidamente con el salpique de agua y contagiando a todo el cultivo [3], para los agricultores esto significa, disminución de calidad tanto, del producto como de la cantidad de cosecha. En Ecuador, este cultivo se realiza mayormente en la región sierra, debido al clima y tipo de suelo; sin embargo, se registra que la mancha angular a afectado hasta un 48% del cultivo en ciudades como Imbabura y Azuay. En el cantón Saraguro, por medio de una entrevista realizada al Técnico de Apoyo Agropecuario y Gestión Ambiental del GAD municipal, se conoce que esta enfermedad ha afectado a muchos de los pequeños agricultores que son parte de la Escuela Agroecológica impulsada por esta institución, quienes, debido a la falta de conocimiento para identificarla, recurren a la aplicación descontrolada de agroquímicos que no son correctos para el tratamiento de su cultivo. Por lo tanto, la detección apropiada de esta enfermedad permite la elección correcta del tratamiento preventivo, evitando gastos innecesarios y mejorando la calidad de producción.

En este contexto, la aplicación de técnicas de inteligencia artificial se ha convertido en una herramienta valiosa para la identificación de enfermedades foliares. En particular, las Redes Neuronales Convolucionales (CNN) han demostrado ser altamente eficaces al reconocer patrones y características específicas en imágenes [5], ideales para detectar enfermedades en hojas. Con el fin de realizar la detección de la mancha angular utilizando visión por computadora en los cultivos, se ha planteado la siguiente pregunta de investigación: ¿En qué medida podrá el modelo predictivo basado en deep learning identificar la enfermedad de la Mancha Angular en las hojas de fresa del Cantón Saraguro?

El principal objetivo para dar respuesta a la pregunta de investigación, consiste en implementar un modelo predictivo basado en deep learning para identificar la Mancha Angular (*Xanthomonas fragariae*) en la hoja de la Fresa para el Municipio del cantón Saraguro, planteando como objetivos específicos implementar un modelo de red neuronal convolucional y el conjunto

de datos necesarios para la detección de la Mancha Angular en la fresa, además, construir un prototipo de software e integrar el modelo de red neuronal.

Este proyecto en relación con otros trabajos afines, se distingue por la creación de un dataset local mediante una práctica experimental en cuatro huertos del cantón Saraguro, abarcando tanto cultivos al aire libre, como invernaderos. Este conjunto de datos se complementó con imágenes del repositorio de Kaggle (ver sección 4.2.16), ampliando la variedad de ejemplos de la enfermedad en contraste con los trabajos relacionados (ver sección 4.6) que se limitan al uso de datasets disponibles en la web. También se añadió una nueva clase con hojas sanas al modelo final con el propósito de mejorar su precisión en la detección. La implementación se llevó a cabo en Google Colab, aprovechando su capacidad de procesamiento en la nube. Además, en este trabajo se integra el modelo de aprendizaje profundo en un prototipo de software para facilitar su uso y evaluación.

La estructura de este trabajo se exhibe de la siguiente manera: En primer lugar, se encuentra una introducción donde se contextualiza el tema y se expone los objetivos de la investigación. Luego, se presenta un marco teórico que aborda los fundamentos teóricos y conceptuales relacionados con la mancha angular en la hoja de fresa, los modelos de deep learning utilizados en la detección de enfermedades foliares y las métricas de evaluación para medir el desempeño de los mismos. Posteriormente, se describe la metodología empleada para la implementación del modelo predictivo. Subsecuentemente, se presentan los resultados obtenidos a partir de la aplicación de las técnicas propuestas. Seguidamente, se realiza la sección de discusión donde se analizan y se interpretan los resultados obtenidos. Finalmente se exponen las conclusiones procedentes de la investigación, junto a un conjunto de recomendaciones para futuros estudios en este campo de la Inteligencia Artificial.

## **4. Marco teórico**

Esta sección provee la información vital para comprender el desarrollo de Trabajo de Titulación. Los subtemas presentes en esta sección comprenden: Antecedentes, donde se da a conocer brevemente el contexto de la problemática relacionada con el cultivo de fresa en el cantón Saraguro, el Fundamento Teórico, contiene los conceptos relacionados con el campo de la inteligencia artificial y las técnicas aplicadas para llevar a cabo la detección de objetos, entre los cuales están las herramientas que serán utilizados para la realización del presente proyecto y porque fueron seleccionados, se describen las métricas de evaluación que serán utilizadas para dar respuesta a la pregunta de investigación, además de las metodologías aplicadas para el cumplimiento de cada objetivo, finalmente la parte de trabajos relacionados que aportarán las diferentes soluciones aplicadas para resolver problemas similares y como este trabajo se diferencia de los mismos.

### **4.1 Antecedentes**

#### **4.1.1 Escuela Agroecológica de Saraguro**

La escuela Agroecológica del cantón Saraguro está conformada por un grupo de pequeños agricultores que desde el año 2008 iniciaron a cultivar desde sus huertos productos agroecológicos, con el objetivo de impulsar el consumo de alimentos saludables sin químicos. Actualmente este grupo ha tomado el nombre de Escuela Agroecológica “Sumak Tarpuna”, y está bajo la cobertura del Gobierno Autónomo Descentralizado Intercultural de Saraguro (GADMIS), quienes brindan capacitaciones y productos como semillas, abono, plantas, etc., a todos quienes son parte de esta entidad. En el año 2018, el GADMIS inauguró el primer Mercado Agroecológico, que fue un espacio de comercialización de productos sanos, liderado por todos los productores de esta escuela.

Actualmente, se han sumado más agricultores de diferentes sectores del cantón, cultivando diferentes tipos de productos, entre ellos la fresa, siendo el de mayor demanda dentro del mercado local.

#### **4.1.2 Mancha Angular**

La Mancha Angular, también conocida como Angular Leaf Spot en inglés, es una enfermedad común en el cultivo de la fresa, en especial en huertos al aire libre, aunque también puede presentarse en invernaderos, esta es causada por el patógeno *Xanthomonas Fragariae* [1].

Se caracteriza principalmente por la aparición de manchas angulares necróticas primeramente en las hojas, migrando si no se controla a tallos y frutos de las plantas afectadas. Estas lesiones pueden variar en tamaño y color, y están rodeadas por un halo amarillo, se debe principalmente a los cambios en los factores climáticos y la humedad no controlada en el espacio de siembra.

La presencia de la Mancha Angular en la fresa puede tener consecuencias negativas tanto en la calidad de los frutos como en la productividad del cultivo, debido a que las lesiones en las hojas pueden reducir la capacidad fotosintética de la planta, afectando su crecimiento y desarrollo y por consecuencia la producción [2].



**Figura 1.** Enfermedad de la mancha angular en la hoja de fresa

#### **4.1.3 Síntomas y daños asociados a la mancha angular en la fresa**

Los principales síntomas que se presentan en el cultivo de fresa a causa de la enfermedad de mancha angular son:

- *Aparición de manchas angulares:* Las lesiones comunes de la mancha angular en las hojas de fresa son de forma angular y de color amarillo, marrón oscuro a negro. Estas manchas suelen tener una forma definida, con bordes bien delimitados, y pueden variar en tamaño dependiendo del avance de la enfermedad.
- *Halo amarillo:* Las lesiones angulares están rodeadas por un halo amarillo, que se observan fácilmente en las hojas jóvenes y llegan a extenderse más allá de los bordes de las manchas necróticas.
- *Decoloración y marchitamiento:* Con el avance de la enfermedad, las hojas infectadas pueden presentar decoloración completa, marchitamiento e incluso necrosis en toda la hoja.

## **4.2 Fundamentos teóricos**

### **4.2.1 Inteligencia Artificial (IA)**

Esta ciencia intenta diseñar y construir máquinas capaces de imitar el comportamiento e inteligencia del ser humano. La IA es una rama de la computación que se enfoca en el desarrollo de algoritmos y sistemas que permiten a las máquinas simular la inteligencia humana y realizar tareas que requieren de la capacidad cognitiva de un ser humano, como la percepción visual, el reconocimiento de patrones, el procesamiento del lenguaje natural y la toma de decisiones. La IA se desenvuelve en áreas clave, incluyendo el aprendizaje automático, la visión por computadora, el procesamiento del lenguaje natural y la robótica, que hoy se consideran ramas importantes ya que pueden transformar cómo operan las empresas y las industrias, mejorando la eficiencia y la toma de decisiones para resolver algunos de los desafíos más urgentes de la sociedad, como la atención médica, la energía y el medio ambiente [3].

### **4.2.2 Machine Learning**

El aprendizaje automático o machine learning es una rama de la inteligencia artificial propuesta por Arthur Samuel en 1959, que se enfoca en el desarrollo de algoritmos y modelos estadísticos capaces de aprender patrones y relaciones a partir de datos, sin necesidad de ser programados explícitamente para ello. Estos modelos pueden entrenarse con datos para mejorar su capacidad de generalización y predicción, lo que los hace útiles para muchas aplicaciones en campos como la ciencia de datos, visión por computador y el procesamiento de lenguaje natural. Una de las técnicas más comunes en el Machine Learning es el aprendizaje supervisado, en el cual se proporcionan ejemplos etiquetados para enseñar al modelo a hacer predicciones precisas. Otras técnicas incluyen el aprendizaje no supervisado y el aprendizaje por refuerzo [4][5].

### **4.2.3 Deep Learning**

El Aprendizaje automático o en inglés llamado Deep Learning es una que utiliza redes neuronales artificiales de múltiples capas para aprender representaciones de alto nivel a partir de datos. Estas redes neuronales pueden aprender automáticamente características abstractas y complejas de los datos de entrada, lo que las hace muy útiles para tareas como el reconocimiento de imágenes y el procesamiento del lenguaje natural. Una de las principales ventajas del Deep Learning es su capacidad para manejar grandes volúmenes de datos y aprender características útiles

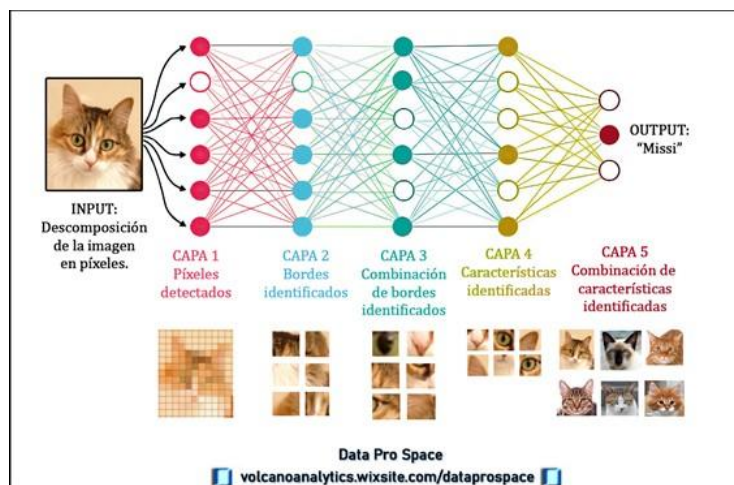
de manera no supervisada lo cual ha significado importantes avances en el área de visión por computadora y la robótica [6].

#### 4.2.4 Visión por computadora

La visión por computador es una parte de la IA que se enfoca en crear algoritmos y técnicas para que las máquinas puedan interpretar y comprender imágenes y videos, con la finalidad de extraer información útil de estos [7]. Esta disciplina se enfoca en el desarrollo de algoritmos y técnicas de procesamiento de imágenes, aprendizaje automático y reconocimiento de patrones, estos algoritmos se aplican a problemas específicos ajustándose a las características de los datos y objetivos de la aplicación.

#### 4.2.5 Redes Neuronales Convolucionales

Las redes neuronales convolucionales (CNN, por sus siglas en inglés) son un tipo de arquitectura de redes neuronales profundas conformadas de múltiples capas convolucionales, de agrupamiento y completamente conectadas, que aprenden a identificar y extraer características relevantes de las imágenes mediante el uso de filtros convolucionales, que se aplican a cada parte de la imagen para detectar patrones específicos, tales como bordes o texturas, posteriormente pasan a través de capas de agrupamiento, que reducen la dimensionalidad de la salida de las capas convolucionales, permitiendo que la red pueda procesar la información más eficientemente. Finalmente, las capas completamente conectadas se encargan de la clasificación de la imagen en una o varias categorías. En la Figura 2 tomada de [8], se indica el proceso que sigue una CNN desde la entrada de una imagen hasta su salida para obtener la clasificación.



**Figura 2.** Proceso del funcionamiento de una Red Neuronal Convolutiva

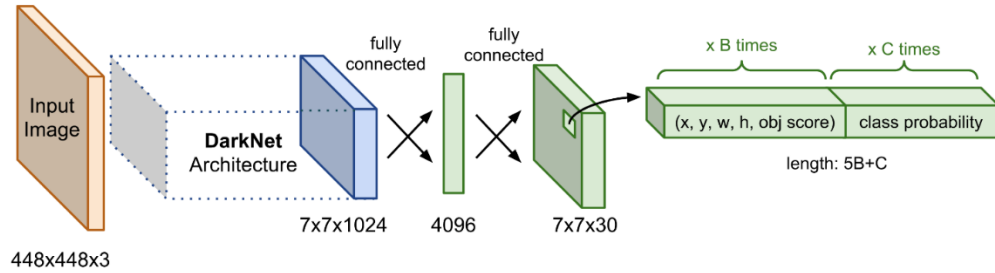


Las redes neuronales están conformadas por algunos elementos que le permiten llevar a cabo un funcionamiento eficiente para reconocer patrones y características en las imágenes [9] estos elementos son:

- **Capas convolucionales (Convolutional layer):** estas capas aplican filtros convolucionales a la entrada para extraer características relevantes de la imagen, es decir aplica una función de ventana deslizante a una matriz de píxeles que representa una imagen. Cada filtro convolucional se desplaza sobre la imagen de entrada y realiza operaciones matemáticas para producir una representación enriquecida de la imagen. Las capas convolucionales se componen de múltiples filtros, cada uno aprendiendo a detectar un patrón específico en la imagen
- **Capas de agrupamiento (Pooling layer):** estas capas reducen la dimensionalidad de la salida de las capas convolucionales mediante la combinación de varias características en una sola característica. El objetivo de las capas de agrupamiento es reducir el número de parámetros en la red y evitar el sobreajuste
- **Capas completamente conectadas:** estas capas toman las características extraídas por las capas convolucionales y de agrupamiento y las utilizan para clasificar la imagen. En esta capa, cada nodo de la capa de salida está conectado directamente a uno de la anterior, por lo que, con base en las características extraídas de las capas anteriores junto con los filtros, se realiza la tarea de clasificación
- **Funciones de activación:** estas funciones se aplican después de cada capa y permiten a la red aprender relaciones no lineales entre las características extraídas [10].

#### 4.2.6 YOLO (You Only Look Once)

Es un algoritmo de detección de objetos en imágenes y videos en tiempo real creado en el año 2016 por José Redmon. A diferencia de otros métodos que dividen la imagen en regiones y las analizan por separado, YOLO utiliza una única pasada de la red neuronal convolucional para predecir simultáneamente las clases y las ubicaciones de múltiples objetos en la imagen completa, por lo que es mucho más rápido y preciso al ser comparado con otros modelos de detección de objetos [11].

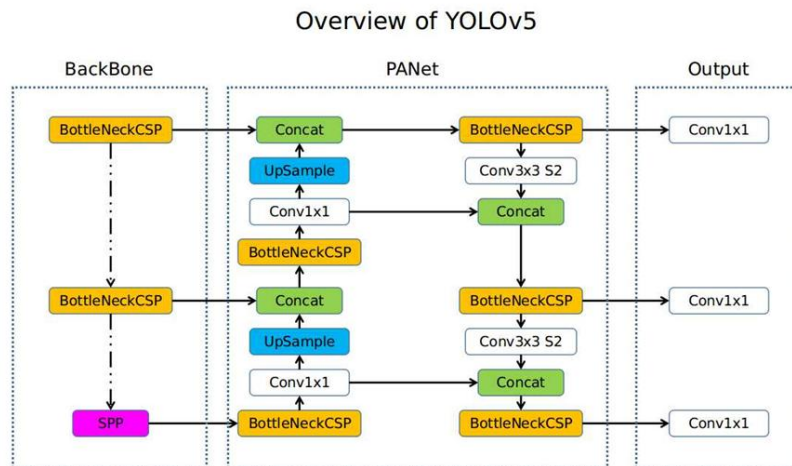


**Figura 3.** Arquitectura Yolo

La arquitectura de este modelo se caracteriza por procesar todas las características de las imágenes, a esto se lo conoce como Darknet. Seguido se le aplican dos capas totalmente conectadas para predecir los cuadros delimitadores de los objetos como se muestra en la Figura 3. Fue evaluado con el conjunto de datos Pascal VOC con parámetros específicos que determinan el tamaño de los mapas de características finales y la salida del modelo.

#### 4.2.7 Yolov5

Con los años se han lanzado varias versiones mejoradas de YOLO y en el año 2020 Ultralytics LLC creó la edición de la serie Yolo v5, esta arquitectura como se visualiza en la Figura 4, se basa en una red neuronal convolucional (CNN) profunda y eficiente, utilizando un enfoque de detección de objetos de extremo a extremo. A diferencia de los enfoques basados en regiones, que dividen la imagen en múltiples regiones de interés para la detección de objetos, Yolov5 realiza la detección y clasificación de objetos directamente a partir de la imagen completa [12].



**Figura 4.** Arquitectura Yolov5

Este modelo está formado por tres partes principales que son:

- **Backbone (Espina dorsal):** El backbone, o espina dorsal, es la parte principal de la red neuronal convolucional (CNN) que se encarga de extraer características de la imagen de entrada. En el contexto de detección de objetos, el backbone procesa la imagen a través de múltiples capas convolucionales para capturar características de diferentes niveles de abstracción. Estas características se utilizan posteriormente para identificar objetos en la imagen.
- **PANet (Feature Pyramid Network):** PANet, o Feature Pyramid Network (FPN), es una arquitectura diseñada para capturar características a diferentes escalas en una imagen. Consiste en una red piramidal de características que fusiona las características de diferentes niveles de resolución para mejorar la detección de objetos a diferentes escalas. Esto permite detectar objetos de diferentes tamaños y escalas en una sola pasada de red.
- **Output (Salida):** La salida de la red neuronal es la predicción final del modelo para una imagen de entrada dada. En el contexto de detección de objetos, la salida generalmente consiste en una serie de cuadros delimitadores (bounding boxes) que representan la ubicación y el tamaño de los objetos detectados, junto con las probabilidades de clase que indican la confianza del modelo en la presencia de cada objeto detectado. La salida se produce después de que la imagen de entrada ha sido procesada por el backbone y PANet, y ha pasado por capas de detección y supresión de no máximos.

#### 4.2.8 Yolov7

Esta arquitectura se construye teniendo como base modelos anteriores como Yolov4, Yolo-R y Scaled Yolov4, fusionando los aprendizajes obtenidos de estos modelos para mejorar la precisión general.

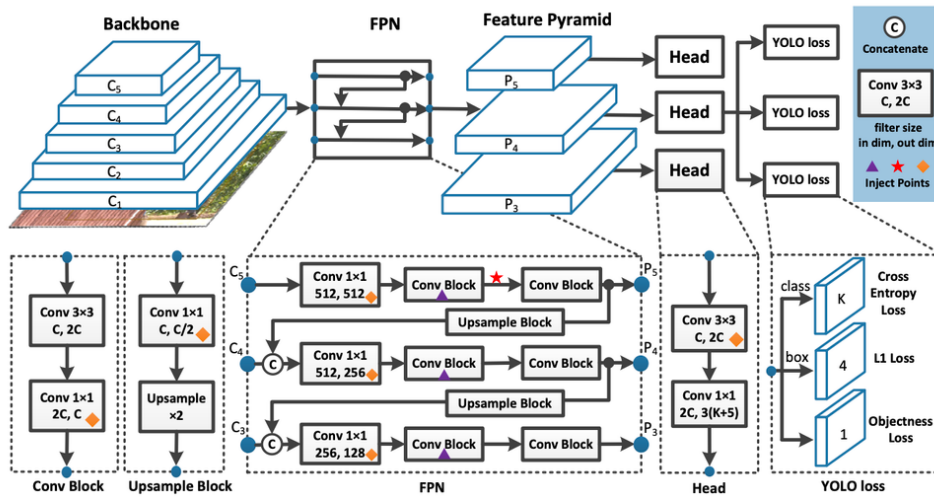


Figura 5. Arquitectura YOLOv7

- **Red extendida de agregación de capas eficientes (E-ELAN)**

En la arquitectura de YOLOv7 se utiliza un bloque llamado E-ELAN, que está diseñado teniendo en cuenta aspectos como la cantidad de parámetros, la carga computacional y factores importantes como el rendimiento de la red. Además, incorpora estrategias de re-parametrización para mejorar la robustez del modelo frente a los patrones que intenta modelar. Estas estrategias se basan en la planificación de rutas de propagación de flujo de gradiente para determinar qué partes de la red deben re-parametrizarse y cuáles no [13].

- **Escalado de modelos para modelos basados en concatenación**

Finalmente, introduce un cabezal auxiliar de detección de objetos, que se ubica en un punto intermedio de la red y permite supervisar el proceso de entrenamiento con diferentes niveles de granularidad. Esto ayuda a mejorar la eficiencia del entrenamiento y la precisión de las predicciones finales [14].

#### 4.2.9 YOLOv8

Este algoritmo está caracterizado por el manejo de una única red neuronal para la predicción simultánea de cajas delimitadoras y probabilidades de clase asociadas a objetos presentes en una imagen [16]. La versión contiene mejoras como módulos de atención espacial, fusión de características y agregación de contexto, que detectan objetos más eficientes y precisas.

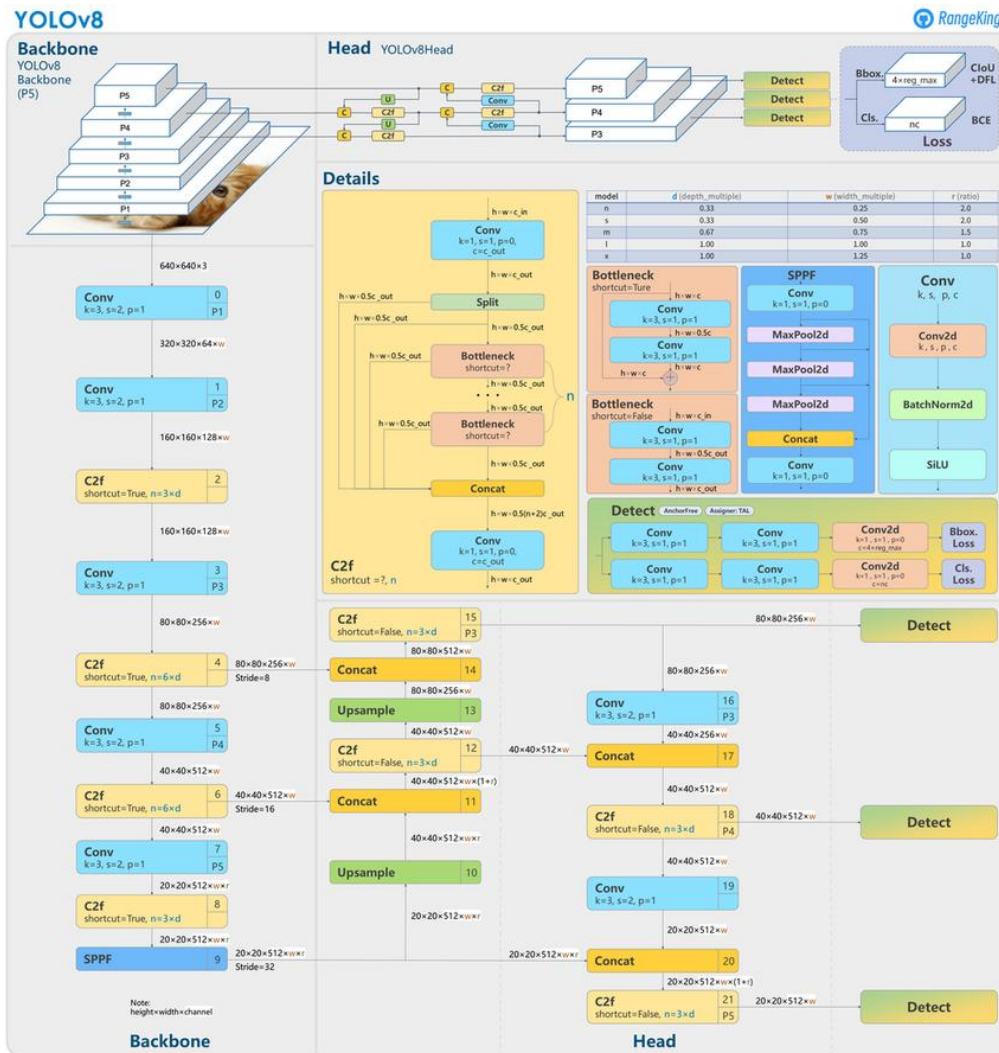


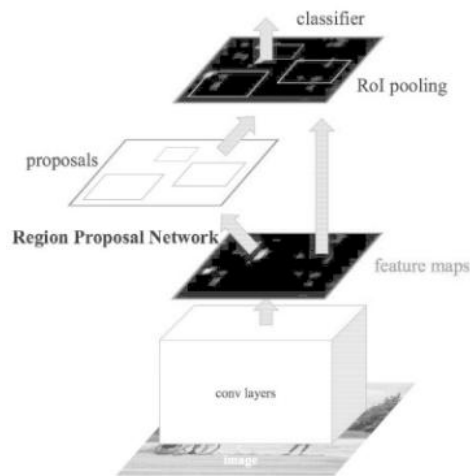
Figura 6. Arquitectura YOLOv8

La arquitectura subyacente de YOLOv8 se fundamenta en versiones previas de los algoritmos YOLO, presentando una estructura compuesta por dos componentes principales: la columna vertebral y la cabeza. La columna vertebral de YOLOv8 se construye a partir de una variante modificada de la arquitectura CSPDarknet53, la cual consta de 53 capas convolucionales e incorpora conexiones parciales entre etapas cruzadas para mejorar la propagación de información entre capas. En lo que respecta a la cabeza de la red, esta se compone de múltiples capas convolucionales seguidas por capas completamente conectadas, siendo estas últimas responsables de la generación de predicciones concernientes a cajas delimitadoras, puntuaciones de objetividad y probabilidades de clase asociadas a los objetos detectados en la imagen [17].

Yolov8 se caracteriza porque integra un mecanismo de auto atención en la cabeza de la red, la cual habilita al modelo para enfocarse en distintas regiones de la imagen, ajustando que características son más importantes dependiendo la tarea que se esté realizando.

#### 4.2.10 Faster R-CNN

Faster R-CNN es un algoritmo detección de objetos basado en aprendizaje profundo, esta arquitectura contiene una Red de Propuestas de Región(RPN), cuyo objetivo es eliminar el requisito de ejecutar la búsqueda selectiva antes de la inferencia, y en su lugar integrar la propuesta de región de manera directa en la arquitectura R-CNN. La Figura 7 proporciona una vista general de la arquitectura actual de este modelo, observándose en primer lugar una imagen de entrada a la red, cuyas características se extraen mediante CNN previamente entrenada, es decir la red base, estas características en paralelo se envían a dos componentes diferentes de la arquitectura Faster-RCNN.

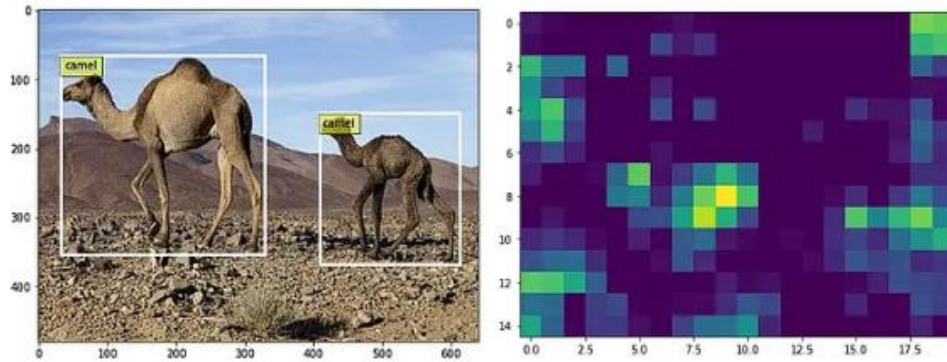


**Figura 7.** Arquitectura de Faster R-CNN

Esta arquitectura es un sistema de detección de objetos que consta de varias etapas interconectadas, mismas que se detallan a continuación:

- **Red Convolucional Base (Base Network)**

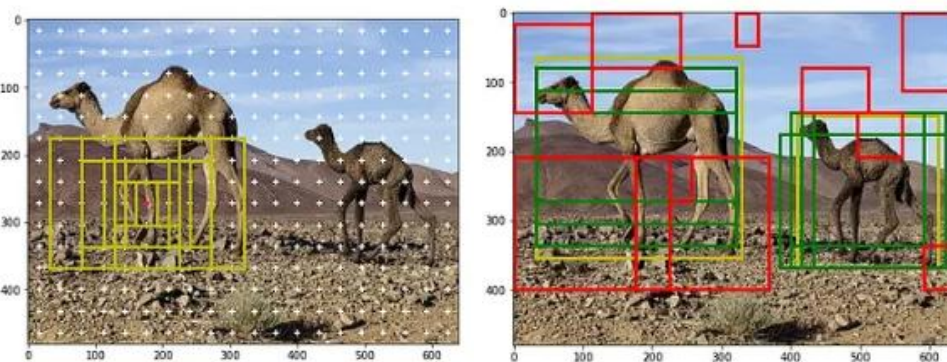
Después de ingresar la imagen en la arquitectura, el primer componente que encontramos es la red base. La red base suele ser una CNN previamente entrenada para una tarea de clasificación particular, como VGG16 o Resnet, para la extracción de características convolucionales que posteriormente son utilizadas por el resto del sistema para realizar la detección de objetos [18].



**Figura 8.** Mapa de características de salida de la imagen después de pasarlas a través de la columna vertebral (Red base)

- **Red de Propuestas Regionales (RPN)**

El RPN opera sobre el mapa de características convolucionales, generando propuestas de regiones que podrían contener objetos mediante el uso de anclajes, que son regiones predefinidas de diversos tamaños y relaciones de aspecto, en este punto no se sabe que es el objeto, solo que potencialmente hay un objeto en cierta ubicación de la imagen [19].



**Figura 9.** Cuadros de anclaje en el espacio de la imagen

- **RoI Pooling Layer (Región of Interest Pooling)**

Las regiones propuestas en la RPN son procesadas por una capa de ROI Pooling, donde se extraen características de tamaño fijo de cada región. Esta capa permite evaluar rápidamente la "objetividad" de una región en la imagen, clasificándola como primer plano o fondo. Posteriormente, las regiones de anclaje etiquetadas como fondo son descartadas, mientras que las regiones de primer plano se transmiten al módulo de ROI Pooling. En la Figura 10 se visualiza el etiquetado de "objetividad" de una región de interés de entrada.





**Figura 10.** Etiquetado objetivo de un ROI

- **Capas de Clasificación y Regresión**

Las características extraídas de cada región propuesta se procesan mediante dos rutas paralelas. En la ruta de clasificación, se usan capas conectadas para predecir las probabilidades de cada clase de objeto dentro de la región propuesta. Esto permite determinar qué objetos están en la región propuesta. Por otro lado, en la ruta de regresión, se emplean capas totalmente conectadas para predecir ajustes precisos a las coordenadas del cuadro delimitador de cada objeto dentro de la región propuesta. Esta etapa fina de ajuste ayuda a refinar la localización de los objetos detectados, mejorando así la precisión del modelo.

- **Non-Maximum Suppression (NMS) y Post-Procesamiento**

Se utiliza la supresión no máxima para eliminar las detecciones redundantes y seleccionar las detecciones más confiables. En esta técnica, primero consideramos el cuadro delimitador con la puntuación de clasificación más alta. Luego calculamos el IoU de todos los demás cuadros con este cuadro y eliminamos los que tienen una puntuación de IoU alta. Estos son los cuadros delimitadores duplicados que se superponen con los "originales".

#### **4.2.11 Metodología CRISP-DM**

CRISP-DM (Cross Industry Standard Process for data Mining) se trata de un marco metodológico orientado a la analítica y obtención de modelos basados en datos. Esta metodología es muy utilizada en la Minería de Datos teniendo ciertas adaptaciones dependiendo si se emplea para machine learning, deep learning o con el uso de AutoML [20] [21].



#### 4.2.11.1 Fases de la metodología CRISP-DM

- **Entendimiento del Negocio:** Esta fase tiene como finalidad la comprensión de los objetivos y requerimientos del proyecto desde una perspectiva de negocio, para luego convertirlos en un Plan de Minería de Datos.
  - Determinar los Objetivos del Negocio
  - Traducir los objetivos empresariales en objetivos de ML
  - Evaluar la viabilidad del proyecto
  - Realizar el Plan de Proyecto
- **Entendimiento de Datos:** Esta fase consiste en la recolección y exploración inicial de los datos para la comprensión de estos, por lo cual se llevan a cabo las siguientes tareas:
  - Recolectar datos iniciales
  - Descripción de los datos
  - Exploración de los datos
  - Verificar la calidad de los datos
- **Preparación de los datos:** En esta fase se construye, a partir de los datos recolectados, el conjunto de datos(Dataset) que alimentará finalmente al modelo. Para ello se lleva a cabo las siguientes tareas:
  - Limpieza de datos
  - Construir los datos
  - Integrar los datos
- **Modelado:** Esta fase consiste en la aplicación de diferentes modelos machine learning/deep learning/data mining, por ejemplo, se pueden crear modelos de clasificación o detección, para esto se realizan las siguientes tareas:
  - Escoger técnica de modelado
  - Generar plan de prueba
  - Construir el modelo
  - Evaluar el modelo
- **Evaluación:** Esta fase busca determinar el modelo más adecuado para solucionar el problema y aproximarse a los objetivos de la investigación, a través de la evaluación de los resultados obtenidos y por tanto se puede pasar al despliegue. En esta fase se realizan las siguientes tareas:

- Evaluar Resultados
- Revisar el proceso
- Determinar los próximos pasos
- **Implementación:** en esta última fase se concretan las estrategias para implementar, monitorear y mantener el modelo, garantizando un servicio al cliente sin interrupciones. Se realiza una revisión exhaustiva del proceso del proyecto para identificar lecciones aprendidas y mejorar en futuros proyectos. Las tareas por realizarse en esta fase son:
  - Evaluar el modelo en condiciones de producción
  - Garantizar la aceptación del usuario y su facilidad de uso

#### 4.2.12 Colab

Colab es una plataforma de colaboración en línea que permite escribir y ejecutar código de Python en Notebooks, permitiendo almacenar dichos proyectos en la nube. Fue creado por Google Research y otorga acceso gratuito a GPU (Unidad de procesamiento gráfico) y TPU (Unidad de Procesamiento Tensorial). Está diseñada para el desarrollo de aplicaciones de IA y Análisis de datos. Las características más importantes de esta plataforma son [22]:

- Ejecutar comandos de terminal del pc.
- Importar conjuntos de datos de repositorios externos como Kaggle y Roboflow.
- Guardar cuadernos de Google Colab en el repositorio de Google Drive.
- Importar cuadernos desde Google Drive.
- Servicio gratuito pero limitado en la nube para el uso de GPU y TPU.
- Integración con PyTorch, Tensorflow, Open CV para el desarrollo de análisis de datos e IA.
- Importación o publicación directa en o desde GitHub.

#### 4.2.13 Zero-Shot Learning

Se refiere a la capacidad que tiene el modelo para realizar tareas para las cuales no ha sido entrenado, es decir que el modelo puede generalizar y realizar estas tareas sin haber sido expuesto a ejemplos específicos durante su entrenamiento inicial [23]. Esta técnica es útil en el reconocimiento de imágenes donde el modelo puede identificar objetos no presentes en el conjunto de datos de entrenamiento inicial basándose en los atributos visuales aprendidos.

#### 4.2.14 Matriz de Confusión

En el contexto del aprendizaje automático una matriz de confusión es una tabla que muestra el número de veces que un modelo a predicho correctamente una clase y cuantas veces se equivocó, siendo una herramienta fundamental para evaluar el funcionamiento del mismo. Esta matriz organiza la información en columnas que representan el número de predicciones realizadas para cada clase y filas que representan las instancias clasificadas según su clase real. La matriz de confusión brinda información sobre los aciertos y errores que un modelo ha cometido durante el proceso de aprendizaje con los datos [24].

En la Tabla 1 se indican los valores que representan una clase o categoría predicha por el modelo, y cada fila representa la clase real de las instancias. Cada celda de la matriz muestra cuántas instancias han sido clasificadas correcta o incorrectamente para una combinación específica de clase real y clase predicha [24] [25].

**Tabla 1.** Estructura de la Matriz de Confusión

		Predicción	
		Positivos	Negativos
Observaciones	Positivos	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

Al analizar la matriz de confusión, podemos identificar:

- **Verdaderos positivos (VP):** Representa las instancias que fueron clasificadas correctamente como pertenecientes a una clase específica.
- **Falsos positivos (FP):** Indica las instancias que fueron incorrectamente clasificadas como pertenecientes a una clase específica cuando en realidad no lo son.

- **Verdaderos negativos (VN):** Son las instancias que fueron correctamente clasificadas como no pertenecientes a una clase específica.
- **Falsos negativos (FN):** Representa las instancias que fueron incorrectamente clasificadas como no pertenecientes a una clase específica cuando en realidad sí lo son.

#### 4.2.15 Métricas de la matriz de confusión

- **Exactitud:** Esta métrica mide la proporción de predicciones correctas realizadas por el modelo en relación con el total de casos. Se calcula dividiendo la suma los casos correctamente clasificados para la suma de todos los casos.

$$\frac{VP + VN}{VP + FP + VN} \quad (1)$$

- **Precisión:** Mide el porcentaje de predicciones de VP correctamente detectadas por el modelo en relación con el número total de predicciones positivas. Es decir, nos dice qué tan precisas son las predicciones positivas del modelo [26]. El resultado se obtiene mediante la división del total de predicciones correctas (verdaderos positivos) entre el número total de predicciones positivas.

$$\frac{VP}{VP + FP} \quad (2)$$

- **Sensibilidad/Recall:** Esta métrica calcula la proporción de instancias positivas correctamente identificadas por el modelo en relación con el número total de instancias positivas en el conjunto de datos, es decir, proporciona el porcentaje de casos verdaderos positivos que el modelo a identificado correctamente. El resultado de esta métrica se obtiene dividiendo el número de verdaderos positivos entre la suma de verdaderos positivos y falsos negativos.

$$\frac{VP}{VP + FN} \quad (3)$$

- **Especificidad:** La especificidad es una métrica que indica qué tan bien el modelo es capaz de identificar correctamente las instancias que no pertenecen a una clase específica.

Esta métrica calcula la proporción de instancias negativas que el modelo clasifica correctamente con respecto al número total de instancias negativas en el conjunto de datos. Se calcula dividiendo el número de verdaderos negativos (VN) entre la suma de verdaderos negativos y falsos positivos (FP).

$$\frac{VN}{VN + FP} \quad (4)$$

- **Map:** Es una medida que combina la precisión y el rendimiento promedio para evaluar la capacidad de un modelo para detectar objetos con un alta precision en diferentes situaciones. La métrica se calcula utilizando curvas de precision y recuperación obtenida al variar el umbral de confianza del modelo en la detección de objetos.

#### 4.2.16 Dataset strawberry diseases

El conjunto de datos tiene 2500 imágenes de enfermedades de la fresa adquiridas en campos/invernaderos en Corea del Sur y cuenta con los correspondientes archivos de anotaciones de segmentación para siete tipos de enfermedades de la fresa, con imágenes que abarcan desde las etapas inicial, media y final de cada enfermedad. Fue recopilado por los miembros del laboratorio de IA, departamento de Ingeniería y Ciencias de la Computación, JBNU, para el entrenamiento de un modelo de segmentación de instancias para enfermedades de la fresa [27].

Este conjunto de datos se encuentra de manera libre en el repositorio de Kaggle [28] y contiene los permisos para modificarlo y usarlo tal como está o aumentar la variedad de imágenes si es necesario según la investigación.

**Tabla 2.** Resumen conjunto de Datos de Kaggle

Enfermedad	Entrenamiento	Validación	Pruebas	TOTAL
Mancha foliar angular	245	43	147	435
Antracnosis fruto	52	12	33	97
Tizón de las flores	117	29	62	208
Moho gris	255	77	145	477
Mancha foliar	382	71	162	615

Mildiú polvoroso en la fruta	80	12	43	135
Mildu polvoriento en la hoja	319	63	151	533
<b>Total</b>	1450	307	743	2500

### 4.3 Herramientas de etiquetado de imágenes

#### 4.3.1 LabelImg

LabelImg es una herramienta de software gratuita y ligera para la anotación de imágenes, diseñada para etiquetar cuadros delimitadores de objetos en imágenes de manera fácil y eficiente. Fue lanzada por Tzutalin en 2015 y está escrita en Python. Utiliza QT para su interfaz gráfica, lo que proporciona una experiencia de usuario intuitiva y amigable. Este software permite guardar las anotaciones como archivos XML en formato PASCAL VOC, lo que las hace compatibles con una variedad de otros programas y sistemas de reconocimiento de objetos. [29]

#### 4.3.2 LabelMe The Open Annotation Tool

LabelMe es una herramienta de anotación web que permite de manera sencilla delimitar los objetos de la imagen creando polígonos punto a punto, líneas, rectángulos y círculos añadiéndoles un nombre. Russel et al. creó esta herramienta en 2005 con el objetivo de dar facilidad al anotar y compartir estos datos como contribuciones para otras investigaciones [30].

#### 4.3.3 Roboflow

Roboflow es una herramienta de anotación en línea que incorpora inteligencia artificial, utilizado en la etiquetación de imágenes para la tarea de detección, clasificación y segmentación de objetos. Ofrece varias capacidades de preprocesamiento de datos diseñadas para mejorar conjuntos de datos para entrenar modelos de visión por computadora, con transformaciones, que incluyen redimensionamiento, recorte, rotación y aplicación de filtros de imagen, vitales para preparar los datos para un entrenamiento efectivo. Además, técnicas de aumento de datos como, volteo aleatorio, rotación y ajustes de brillo contribuyen a mejorar la diversidad y robustez del conjunto de datos, aspectos importantes para el rendimiento del modelo [31]. Esta herramienta cuenta con una versión gratuita y otra de pago que contiene más funcionalidades. Permite descargar las etiquetas en formato XML, JSON, COCO, YOLO, PASCAL VOC, CSV y TFRecord.

#### 4.3.4 Comparativa de las herramientas de etiquetado

**Tabla 3.** Tabla Comparativa de las herramientas de etiquetado

Aspecto	LabelImg	Labelme	Roboflow
<b>Precio</b>	Gratuito	Gratuito	Variable (Plan gratuito y planes de suscripción)
<b>Instalación</b>	Descarga e instalación local	Instalación local mediante pip o conda	Plataforma basada en la nube, no requiere instalación
<b>Plataforma</b>	Software de escritorio	Software de escritorio o en línea	En línea, basado en la nube
<b>Gestión de Proyectos</b>	No incluido	Incluido	Incluido
<b>Facilidad de uso</b>	Interfaz sencilla, pero requiere instalación y conocimientos básicos de Python	Interfaz web amigable, no requiere conocimientos de programación	Interfaz intuitiva, diseñada para usuarios con poca experiencia técnica
<b>Tipo de etiquetado</b>	Manual	Manual	Manual y automático (detección de objetos)
<b>Formatos de exportación</b>	XML, CSV	JSON, YAML, PNG	XML, JSON, COCO, YOLO, Pascal VOC, entre otros

De acuerdo al análisis que se muestra en la Tabla 3, la herramienta de etiquetado más pertinente es Roboflow, debido a la variedad de características sobresalientes en comparación con las otras herramientas de etiquetado de imágenes, además que permite un amplia opción de formatos de descarga de las anotaciones.

#### 4.4 Aplicación Móvil

##### 4.4.1 Metodología XP (Programación Extrema)

Esta metodología ágil promueve el trabajo en equipo para llevar a cabo el desarrollo del software, fomentando el crecimiento profesional de los desarrolladores y un ambiente laboral favorable. Esto se debe a la continua retroalimentación entre el cliente y el equipo de desarrollo, lo que asegura una contante comunicación entre los involucrados. Además, se destaca por la implementación de soluciones simples y la valentía para afrontar los cambios de manera efectiva [32] [33].

#### 4.4.2 Historias de Usuario

Las Historias de Usuario son breves narraciones que describen el comportamiento deseado del sistema. Cada una se relaciona con una característica clave del sistema y cumple un rol importante en la estimación temporal y la planificación de lanzamientos. Además, reemplazan la necesidad de un extenso documento de requisitos y desempeñan un papel central en la formulación de las pruebas de aceptación.

Es esencial que cada Historia de Usuario sea clara y bien definida, de manera que los programadores puedan llevar a cabo su implementación en un período de unas pocas semanas. Estas historias encapsulan un entendimiento completo y preciso de las funcionalidades deseadas, permitiendo un enfoque ágil y eficiente en el desarrollo del software [34].

- **Plantilla de una historia de Usuario**

Programación extrema propone una plantilla de historia de usuario, la cual se utilizará en este proyecto para el levantamiento de requerimientos dentro del desarrollo del prototipo de software. Esta plantilla consta de los siguientes campos que se pueden observar en la Tabla 4 tomada de [35]:

- *Número*: Identificación de cada historia de usuario.
- *Nombre Historia de Usuario*: Nombre que identifica a cada historia de usuario
- *Iteración Asignada*: Iteraciones de las historias de usuario
- *Usuario*: Nombre del responsable de la actividad descrita en la historia de usuario.
- *Prioridad*: Este puede ser Alto, Medio, Bajo, de acuerdo a la descripción e importancia de la actividad.
- *Riesgo en Desarrollo*: Puede ser Alto, Medio, Bajo, de acuerdo a la complejidad de desarrollo de la actividad.
- *Programador Responsable*: Persona responsable de llevar a cabo el desarrollo de la actividad.
- *Descripción*: Detalle de la actividad a desarrollar.
- *Observaciones*: Aspectos importantes referentes a la historia.



**Tabla 4.** Plantilla de Historia de Usuario

<b>Historia de Usuario</b>	
<b>Número:</b>	Nombre Historia de Usuario:
<b>Usuario:</b>	Iteración Asignada:
<b>Prioridad:</b>	
<b>Riesgo en Desarrollo:</b>	Programador Responsable:
<b>Descripción:</b>	
<b>Observaciones:</b>	

#### 4.4.3 Roles de la Metodología XP

- **Programador:** Es la persona que lleva a cabo la implementación de las historias de usuario según las necesidades del cliente. Además, estima el tiempo requerido para completar cada historia de usuario y mediante esta información el cliente puede determinar el nivel de prioridad de cada iteración. Esta persona también se encarga de diseñar y ejecutar las pruebas de unidad para el código desarrollado o modificado.
- **Cliente:** Es la persona que decide que funcionalidades se llevara a cabo en cada iteración, estableciendo las prioridades de implementación de acuerdo al valor de negocio que aporta cada historia, además, es responsable del diseño y ejecución de las pruebas de aceptación.
- **Encargado de pruebas (Tester):** Es quien se encarga de realizar pruebas de manera regular y comparte los resultados con el equipo.
- **Encargado de seguimiento (Tracker):** Es el responsable de inspeccionar como se desarrollan las estimaciones ejecutadas por los programadores las compara con el tiempo real de desarrollo, de tal manera que se pueda tener un informe de la calidad de las estimaciones y se identifique áreas en las que mejorar.
- **Entrenador (Coach):** Es quien se encarga de liderar e instruir al equipo en la implementación de las prácticas de la metodología XP, guiando al grupo a la aplicación efectiva de cada una de las fases para lograr un desarrollo ágil y colaborativo.

- **Consultor:** Es un miembro externo del equipo con conocimientos específicos sobre cierta área que resulta importante dentro del proyecto, tiene como finalidad brindar información y ayudar a solucionar problemas específicos.
- **Gestor (Big Boss):** Es el enlace entre el clientes y programadores. Se encarga de obtener los recursos necesarios y dar solución a los problemas que surgen en el desarrollo, administrando reuniones, planes de iteración y la coordinación segura para lograr excelentes resultados.

#### 4.4.4 Fases de la Programación Extrema

- **Exploración:** Esta fase comprende el levantamiento de requisitos del proyecto y su comprensión, a través de la creación de Historias de Usuario priorizándolos en función del valor que tenga para el cliente.
- **Planificación:** En esta fase se seleccionan un conjunto de historias de usuario de acuerdo al nivel de prioridad para desarrollarlos en la iteración actual. Conjuntamente el grupo estima el esfuerzo necesario para completar cada historia y se planifica la asignación de tareas.
- **Diseño:** Se diseña la arquitectura y la estructura del software necesario para implementar las historias de usuario seleccionadas. En esta fase, la finalidad es adaptarse a cada usuario, por lo cual se busca simplicidad y claridad en el diseño.
- **Codificación:** Los desarrolladores escriben el código para implementar las historias de usuario. Se enfoca en la calidad y se utilizan pruebas unitarias para verificar la funcionalidad.
- **Pruebas:** Se realizan pruebas exhaustivas, incluyendo pruebas unitarias, de integración y funcionales, para asegurarse de que el software cumpla con los requisitos establecidos.

- **Integración:** Se integran regularmente los cambios en un repositorio compartido, lo que permite la detección temprana de problemas de integración y facilita la colaboración.
- **Entrega:** Al final de cada iteración, se entrega software funcional y potencialmente desplegable al cliente para su evaluación y retroalimentación.
- **Retroalimentación del Cliente:** El cliente revisa y evalúa el software entregado, proporcionando comentarios que guían las iteraciones futuras y permiten ajustes en los requisitos.
- **Refactorización:** Se mejora el diseño y la estructura del código según sea necesario, manteniendo la base de código limpia y adaptable.
- **Pruebas de Aceptación:** Las pruebas finales se realizan para verificar que las historias de usuario se hayan implementado correctamente y cumplan con los criterios de aceptación.
- **Despliegue:** Una vez que se ha alcanzado un conjunto de características deseado, el software se despliega en producción y está disponible para su uso.
- **Mantenimiento:** Después del despliegue, el equipo sigue manteniendo y mejorando el software en función de la retroalimentación del cliente y las nuevas necesidades que surjan.

#### 4.5 Flutter

Este framework, creado por Google, proporciona un conjunto de herramientas para la construcción de interfaces de usuario. Sus inicios fueron en el año 2015, sin embargo, su producción fue hasta el año 2018. Este Framework fue desarrollado para el desarrollo aplicaciones híbridas con rendimiento nativo [36].

Existen diversas características sobre el framework de Flutter que se darán a conocer:

- **Desarrollo rápido**

Permite un desarrollo ágil debido a su función de recarga en caliente con estado (Stateful Hot Reload), actualiza las aplicaciones en cuestión de milisegundos, pone a disposición un conjunto de widgets personalizados que facilitan la creación de aplicaciones nativas en poco tiempo.

- **Rendimiento Nativo**

Los widgets están configurados para una alta integración con la plataforma, las cuales incluyen animaciones, navegación, iconos y fuente lo que garantiza el rendimiento nativo en dispositivos como Android e IOS.

#### 4.6 Trabajos Relacionados

En base a la revisión bibliográfica realizada para la fundamentación teórica del presente Trabajo de Titulación (TT) y al desarrollo de la Revisión Sistemática de Literatura (RSL) para la determinación de modelos de redes neuronales convolucionales empleados en la identificación de enfermedades en las hojas de plantas vegetales, en la Tabla 5 se presentan algunos de los trabajos relacionados (TR) obtenidos durante estos análisis, mismos que respaldan la presente investigación:

**Tabla 5.** Trabajos Relacionados

<b>Código</b>	<b>Título</b>	<b>Descripción</b>
<b>TA02</b>	Automatic Recognition of Soybean Leaf Diseases Using UAV Images and Deep Convolutional Neural Networks	En este estudio, se comparan cuatro modelos de aprendizaje profundo para reconocer y clasificar enfermedades en la hoja de soja: Inception-v3, Resnet-50, VGG-19 y Xception. Las imágenes se capturaron a 2 metros por encima de la plantación, formando un dataset de 3000 imágenes distribuidas para 6 enfermedades. Los resultados se miden en métrica de precisión, donde, el modelo Inception-v3 obtuvo el valor más alto de precisión (99,04%).
<b>TR03</b>	Beans Leaf Diseases Classification Using MobileNet Models.	Este estudio, se implementa el modelo MobileNet para identificar enfermedades en la hoja de frejol, se utilizó un conjunto de datos público que contiene 1296 imágenes de hojas de frejol divididas 2 clases de enfermedades y 1 sana, el set se dividió en un 80% para entrenamiento, 10% de validación y 10% de pruebas. El modelo se entrenó con 100 épocas y los resultados se midieron en métrica de precisión que alcanzó un 0.95% para la clase sanas, 90% para mancha angular y 93% para la roya.

<b>TR04</b>	<p>Modelo YOLO (You Only Look Once) para el diagnóstico preliminar de la enfermedad de Huanglongbing en la planta de limón en la Región Piura</p>	<p>En este estudio, se utilizó el modelo Yolov5 para el diagnóstico de la enfermedad de huanglongbing en la hoja de limón. Se establecieron 3 conjuntos de datos, un dataset propio con 752 imágenes, un dataset de acceso público en Roboflow con 1354 imágenes y un conjunto de datos en la misma plataforma con 3369 imágenes de la hoja de limón, formando un conjunto de 5475 imágenes de hojas sanas y enfermas, el muestreo se dividió en un 70% para entrenamiento, 15% para evaluación y 15% pruebas, la herramienta para el etiquetado fue Yolo Mark RS. Los resultados son: precisión 67%, recuperación 33.3%, sensibilidad la media fue de 52.86% y la especificidad la media fue de 63.57%. El modelo se entrenó con 200 y 300 épocas,</p>
<b>TR05</b>	<p>A Novel Enhanced VGG16 Model to Tackle Grapevine Leaves Diseases With Automatic Method</p>	<p>En este estudio se realiza la implementación y comparación de modelos de aprendizaje profundo de para identificar y clasificar 3 enfermedades en la hoja de uva. Para la creación del set de datos se utilizó un cuadrilóptero para capturar imágenes a 30 y 40 cm de altura del cultivo, se recolectó 8000 imágenes para entrenamiento y 2000 para pruebas, se aplicó la técnica de aumento de datos utilizando Matlab aplicando rotación y brillo. Los resultados obtenidos demuestran que el modelo VGG16 mejorado es más eficiente con una precisión media de 99.40% para moho polvoriento, 99.80% para antracnosis, 99.60% para oídio y 99.70% para sanas.</p>
<b>TR07</b>	<p>Fast anther dehiscence status recognition system established by deep learning to screen heat tolerant cotton</p>	<p>En este estudio se proponen dos modelos de aprendizaje profundo Faster R-CNN y Yolov5 para detectar el número y estado de anteras de algodón. El modelo Yolov5 muestra mayor velocidad de rendimiento y capacidad para implementarse en un extremo móvil. El conjunto de datos está formado por 2845 imágenes de anteras, distribuidas en un 70% para entrenamiento y 30% para validación, para las anotaciones se utilizó el software</p>

	<p>LabelImg. Se aplicó técnicas de aumento de datos como rotación, volteo, recorte, mixup, cutmix y gridmask.</p> <p>Los resultados muestran que el mAP@0.5:0.95 de Yolov5 fue de 0.485, mientras que el mAP@0.5:0.95 de Faster R-CNN fue de 0.478. En mAP@0.5:0.95, Yolov5 fue 0.007 más alto que Faster R-CNN</p>
<p><b>TR08</b></p> <p>A robust deep learning approach for tomato plant leaf disease localization and classification</p>	<p>Este estudio realiza la implementación del modelo deep learning Faster R-CNN con ResNet-34 como red troncal para clasificar enfermedades en la hoja de tomate. Se utilizó el conjunto de datos de PlantVillage con 10 clases de enfermedades. Las anotaciones se realizaron con el software LabelImg. Como resultado Faster-RCNN con ResNet-34 muestra un alto rendimiento con los valores promedio de precisión, recuperación, puntuación F1 y exactitud de 99,48 %, 99,32 %, 99,42 % y 99,97 %, respectivamente.</p>
<p><b>TR09</b></p> <p>Classification of physiological disorders in apples using deep convolutional neural network under different lighting conditions</p>	<p>Este estudio propone el uso de la red neuronal convolucional Xception para clasificar imágenes de hojas de manzana con trastornos fisiológico. La exactitud, precisión, recuperación y puntuación F1 medias del modelo Xception fueron de 0,996, 0,994, 0,996 y 0,998, respectivamente.</p>
<p><b>TR10</b></p> <p>Detection of powdery mildew on strawberry leaves based on DAC-YOLOv4 model</p>	<p>En este estudio se implementa el modelo Yolov4 para identificar la enfermedad del oídio en las hojas de fresa. Para la creación del set de datos se capturo 1023 imágenes con un teléfono móvil, Para el etiquetado se utilizó el software LabelImg, y se distribuyó la muestra en un 75 % para entrenamiento, el 10 % para validación y el 15 % para prueba. Se utilizó siete técnicas de mejora de datos que incluyen rotación de 90°, 180°, volteo, brillo y atenuación, ruido gaussiano y desenfoque gaussiano, ampliando el conjunto de datos a 6112 imágenes. Como resultado se obtuvo valores de 75,5 %, 72,7 % y 0,716, respectivamente, para las puntuaciones, mAP@0,5 y F1.</p>

<b>TR11</b>	Detecting the Simultaneous Occurrence of Strawberry Fungal Leaf Diseases with a Deep Normalized CNN	<p>Este estudio presenta un modelo de Red Neuronal Convolutacional para identificar las enfermedades del tizón de la hoja y la quemadura de la hoja en la fresa, Se utilizó un conjunto de datos formado por un set primario que está formado por 75 imágenes tomadas en explotaciones de fresas en Kinangop, Kenia central y un set secundario disponible en Kaggle con 1.134 imágenes con 5 clases, el conjunto de datos se dividió en un 80% para entrenamiento y el 20% para validación. Las técnicas de aumento de imagen utilizadas en este estudio incluyen volteo, rotación, recorte, zoom y cizallamiento. Los parámetros de entrenamiento consisten en número de épocas: 30, batch: 32, learning rate: 1e-3. El modelo alcanzó una exactitud del 98%, una precisión del 98,9%, una recuperación del 93,3% y una puntuación f1 del 95,9%. En cuanto a las enfermedades tizón de la hoja y la quemadura de la hoja se obtuvo una precisión del 86,67%.</p>
<b>TR13</b>	Deep Learning Application for Plant Diseases Detection	<p>Este estudio presenta un modelo basado en la arquitectura VGGnet16 para el reconocimiento y clasificación de hojas enfermas y sanas. El conjunto de datos contiene 13692 imágenes de hojas de plantas recopiladas en diferentes recursos (sitio web, conjunto de datos gratuito, imágenes capturadas con la cámara del smartphone), se distribuye en el 80% para entrenamiento y el 20% para pruebas. Según los resultados experimentales, el modelo VGGnet16 con el optimizador SGD alcanzó una precisión del 90% en la prueba de validación.</p>
<b>TR14</b>	Deep Learning Based Classification for Paddy Pests & Diseases Recognition	<p>En este estudio se utilizó el modelo CaffeNet para identificar plagas y enfermedades en las hojas de arroz. El conjunto de datos está formado por 4.511 imágenes recopiladas utilizando motores de búsqueda y sometidos a técnicas de aumento de datos. El resultado del experimento en el modelo logró una precisión del 87%</p>

<b>TR19</b>	Design of Efficient Methods for the Detection of Tomato Leaf Disease Utilizing Proposed Ensemble CNN Model	Este estudio compara los modelos CNN MobileNetV3Small, EfficientNetV2L, InceptionV3 y MobileNetV2 y dos modelos propios para identificar enfermedades en la hoja de tomate. El conjunto de datos se tomó de PlantVillage y está formado por 18.160 imágenes divididos en 10 clases. El modelo EfficientNetV2L demostró un mejor rendimiento en su precisión promedio con un 99%.
<b>TR20</b>	DCNet: DenseNet-77-based CornerNet model for the tomato plant leaf disease detection and classification	Este estudio presenta el modelo CornerNet basado en DenseNet-77, para la localización y clasificación de las anomalías de las hojas de la planta de tomate. El conjunto de datos es tomado del set de Plant Village, formado por diez enfermedades, se utilizó LabelImg para el etiquetado. Se obtiene los resultados en las medidas de precisión, recuperación, puntuación F1 y exactitud, con un recuento numérico de 99.62%, 99.53%, 99.57% y 99.98%, respectivamente.
<b>TR21</b>	Deep Learning Utilization in Agriculture: Detection of Rice Plant Diseases Using an Improved CNN Model	Este estudio propone un método de aprendizaje por transferencia basado en VGG19, para detectar y clasificar seis clases distintas de la hoja de arroz: sana, mancha marrón estrecha, escaldado foliar, estallido foliar, mancha marrón y tizón foliar bacteriano. El conjunto de datos contiene 2167 imágenes disponibles en línea, y está distribuido en el 80% para entrenamiento y 20% para pruebas. La precisión, la recuperación, la especificidad y la puntuación F1 correspondientes fueron de 0,9620, 0,9617, 0,9921 y 0,9616, respectivamente.
<b>TR22</b>	Deep Learning Based Automatic Grape Downy Mildew Detection	Este estudio propone un modelo de aprendizaje profundo denominado Yolov5-CA, ya que al modelo Yolov5 se le integra un mecanismo de atención coordinada (CA), para identificar la enfermedad del mildiú vellosa en la hoja de uva. Como parámetros de entrenamiento se fijó la época de entrenamiento en 1000, el tamaño del lote en 16 y la tasa de aprendizaje en 0,0013. Los resultados



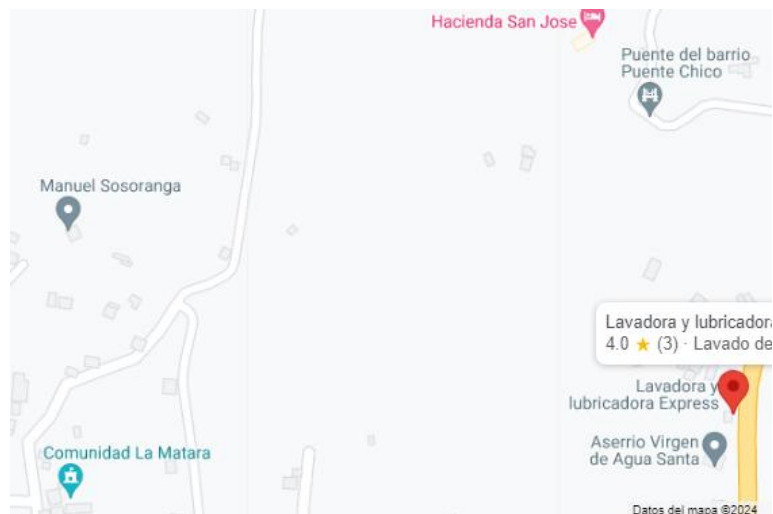
	<p>experimentales muestran que el modelo propuesto alcanza una precisión de detección del 85,59%, un recall del 83,70% y un mAP@0.5 del 89,55%, destacando que la medida de precision es superior Faster R-CNN(79.97%), YOLOv4(82.69%) y YOLOv5(85.35%).</p>
<p><b>TR25</b> Tomato Diseases and Pests Detection Based on Improved YOLO V3 Convolutional Neural Network</p>	<p>Este estudio realiza la implementación del modelo YOLOv3 optimizando la capa de características, para la detección de enfermedades en la hoja de tomate. El conjunto de datos está formado por 15 000 muestras con 10 enfermedades. Los parámetros de entrenamiento del modelo son: batch size: 64, learning rate: 0.01, épocas: 13 000, nms: 0.03. Los resultados se compararon con otros modelos CNN para evaluar el mejor rendimiento, se obtuvo una precisión de 84.32% para SSD, 90.67% para Faster R-CNN, 88.31% para YOLOv3 y 92.39% para el modelo propuesto YOLOv3 mejorado.</p>
<p><b>TR28</b> Tea leaf disease detection and identification based on YOLOv7 (YOLO-T)</p>	<p>Este estudio presenta el entrenamiento del YOLOv7 como una solución para la detección de enfermedades de las hojas de té. El conjunto de datos está formado de 4000 imágenes de cinco tipos de enfermedades de las hojas de estas plantaciones de té. Las muestras se dividieron en el 70% para entrenamiento, 10% para validación y 20% para pruebas. Se utilizó el software LabelImg para el etiquetado. Para el entrenamiento del modelo se establecieron los siguientes parámetros: tamaño de imagen: 640 píxeles, batch size: 10, y número de épocas: 205. En medida de precision se obtuvo los siguientes resultados: Podredumbre negra 97%, Tizón marrón 97%, Roya de la hoja 97%, Araña roja 98% y Mosquito del té 97%.</p>

## 5. Metodología

### 5.1 Área de estudio

El presente TT fue realizado en huertos ecológicos de pequeños agricultores que están bajo la cobertura del departamento de producción del municipio del cantón Saraguro.

En este territorio existen varios sectores de cultivo, por lo cual se ha tomado como casos de estudio a tres de ellos. El primero se encuentra ubicado en la comunidad de la Matara, cuya representante es la Sra. Rosa Zhingre Medina, el segundo, se encuentra ubicado en el barrio Puente Chico, cuyo dueño es el señor Roberto Cango y el tercero en la parroquia del Tablón, en este caso el cultivo es en invernadero, y su dueña es la Sra. Margarita Armijos. Estas personas se dedican a la agricultura y comercio de diferentes tipos de hortalizas y frutas, entre ellos la fresa.



**Figura 11.** Área experimental 1 y 2 – Mapa de la Comunidad La Matara y Barrio Puente Chico



**Figura 12.** Área experimental 3 – Mapa Parroquia El Tablón

## 5.2 Procedimiento

Esta investigación se basó en el enfoque del marco metodológico CRISP-MD adaptando cinco fases como el entendimiento de datos, preparación de datos, modelado, evaluación e implementación.

### 5.2.1 Implementar un modelo de red neuronal convolucional y el conjunto de datos necesarios para la detección de la Mancha Angular en la fresa.

- Fase 1: **Entendimiento de datos**

Recolección de imágenes mediante fotografías de la hoja de fresa que visualice la enfermedad de la mancha angular en distintos huertos del centro de Saraguro.

- Fase 2: **Preparación de datos**

Selección, limpieza y preparación de las imágenes recopiladas para la creación del conjunto de datos que alimentará al modelo de detección.

- Fase 3: **Desarrollo y entrenamiento del modelo**

Búsqueda de un modelo basado en Redes Neuronales Convolucionales prediseñado para su adaptación y entrenamiento en la detección de objetos.

- Fase 4: **Evaluación**

Evaluación de los resultados obtenidos por el modelo en cuanto a la detección de la mancha angular mediante métricas de la Matriz de Confusión.

Contraste de los resultados obtenidos por el modelo de predicción y las etiquetas reales sobre el conjunto de datos de prueba.

### **5.2.2 Construir un prototipo de software e integrar el modelo de red neuronal utilizando la metodología XP**

- **Fase 5: Implementación**

Creación de un prototipo de aplicación móvil para identificar la Mancha Angular a través de una imagen de la hoja de fresa con base en la metodología XP.

## **5.3 Recursos**

Con el propósito de cumplir con los objetivos planteados, se utilizaron los siguientes recursos:

### **5.3.1 Recursos científicos**

- **Método científico:** Este es un enfoque sistemático y riguroso para el estudio de los fenómenos naturales y la adquisición de información objetiva y verificable, ya que, tiende a reunir una serie de características que permiten la obtención de nuevo conocimiento científico. Se basa en la observación, la formulación de preguntas, la elaboración de hipótesis, la realización de experimentos controlados, el análisis de datos y la interpretación de resultados [37]. Este método se utilizó en el presente TT para la selección de estudios relacionados durante la revisión bibliográfica.
- **Método analítico:** El presente método implica la subdivisión de un objeto en componentes o elementos consecutivos [38]. En el caso de este TT, se utilizó dicho enfoque para descomponer el objeto de estudio en diversas etapas. Estas etapas se establecieron como objetivos específicos, acompañados de actividades secuenciales a llevar a cabo.
- **Metodología de Bárbara Kitchenman:** El enfoque de esta metodología radica en el desarrollo de una Revisión Sistemática de Literatura, la cual tiene como objetivo brindar a los investigadores las directrices necesarias para realizar una evaluación e interpretación exhaustiva de toda la investigación pertinente disponible,

relacionada con una pregunta de investigación específica, un área temática determinada o un fenómeno de interés. Su propósito es proporcionar un marco sólido para el análisis y la síntesis de la literatura existente, con el fin de obtener conclusiones fundamentadas y contribuir al avance del conocimiento en el campo correspondiente [39]. La metodología se aplicó al recopilar información relacionada con este TT. Inicialmente se identificaron 138 estudios, de los que luego de un minucioso análisis se seleccionaron 28. La evaluación y extracción de información de cada uno de estos estudios no solo contribuyó a respaldar científicamente el trabajo, sino que también permitió identificar las técnicas y métodos empleados para identificar enfermedades en las hojas de las plantas aplicando técnicas deep learning.

- **Técnica de Observación:** Consiste en registros tomados para observar el desarrollo, comportamiento o características de un fenómeno en un entorno específico, para ser analizado posteriormente. Esta técnica fue aplicada durante la fase de adquisición de imágenes para la elaboración del conjunto de datos.
- **Zero-Shot Learning:** Esta técnica fue aplicada durante la fase de pruebas del modelo deep learning integrado en el software, con la finalidad de verificar la flexibilidad y capacidad del modelo para adaptarse a nuevos escenarios en cuanto a característica de la enfermedad en entornos de campo reales.

### 5.3.2 Recursos técnicos

- **Herramientas colaborativas:** Se hizo uso de herramientas colaborativas disponibles en la web tales como: Google Drive como área de trabajo para la gestión de documentos, imágenes y otros recursos. Zoom como principal herramienta de comunicación entre los principales participantes del proyecto de investigación.
- **Entornos virtuales de desarrollo:** Se utilizaron herramientas colaborativas como Google Colaboratory como editor de código y entorno de aprendizaje de los modelos de detección
- **Roboflow:** Permitted la carga y el procesamiento de conjuntos de datos de imágenes sin etiquetar, donde realiza un análisis exhaustivo de cada imagen para identificar los elementos relevantes. Utilizando su capacidad de aprendizaje automático, asigna automáticamente etiquetas descriptivas a las imágenes, lo que facilita su clasificación y

organización. Esta herramienta también se utilizó para la clasificación de las imágenes en datos de validación, entrenamiento, y pruebas.

- **Flutter:** Este framework facilitó un toolkit para crear las distintas interfaces de usuario en la App, además de que soporta la integración con Tensorflow, lo que permitió y facilitó la implementación del modelo deep learning.

#### 5.4 Participantes

El presente TT enfocado en la línea de investigación de Inteligencia Artificial fue desarrollado por los siguientes participantes:

- La estudiante Diana Gabriela González Chillogalli como autora de la presente investigación, sus actividades iniciaron desde el planteamiento del TT, hasta el desarrollo y finalización de los objetivos planteados en el mismo.
- La Ing. Genoveva Jackeline Suing Albito, Mg. Sc., como directora del TT, quien supervisó los avances académicos y técnicos desarrollados por la autora durante el desarrollo.
- El Ing. Miguel Condolo, como experto, contribuyó con la información y movilización a los lugares de cultivo de fresa, certificando que las enfermedades en el conjunto de datos estuvieran correctamente identificadas, además, participó activamente en las pruebas finales del modelo en un entorno real, facilitando la comparación de resultados para garantizar la efectividad del mismo.
- Agricultores propietarios de los cultivos de fresa de los huertos ecológicos que se encuentran bajo cobertura y apoyo del municipio del cantón Saraguro, en donde se realizó el levantamiento del conjunto de imágenes y las pruebas finales del modelo predictivo.

## 6. Resultados

### 6.1 Objetivo 1: Implementar un modelo de red neuronal convolucional y el conjunto de datos necesarios para la detección de la Mancha Angular en la fresa.

#### 6.1.1 Selección del modelo Deep Learning

Se llevó a cabo la búsqueda de un modelo base, fundamentado en Redes Neuronales Convolucionales, a través de una revisión sistemática de la literatura (RSL), con el afán de establecer modelos o arquitecturas utilizados en el ámbito de la inteligencia artificial, específicamente para identificar enfermedades presentes en las hojas de las plantas, especialmente en la fresa.

Las fases y tareas llevadas a cabo dentro de la RSL se basaron en la metodología propuesta por Bárbara Kitchenham. En la Tabla 6 se detallan las tareas realizadas en cada una de las tres fases: 1) Planificar, 2) Realizar y 3) Documentar la revisión sistemática.

**Tabla 6.** Fases que comprenden el proceso de la presente RSL

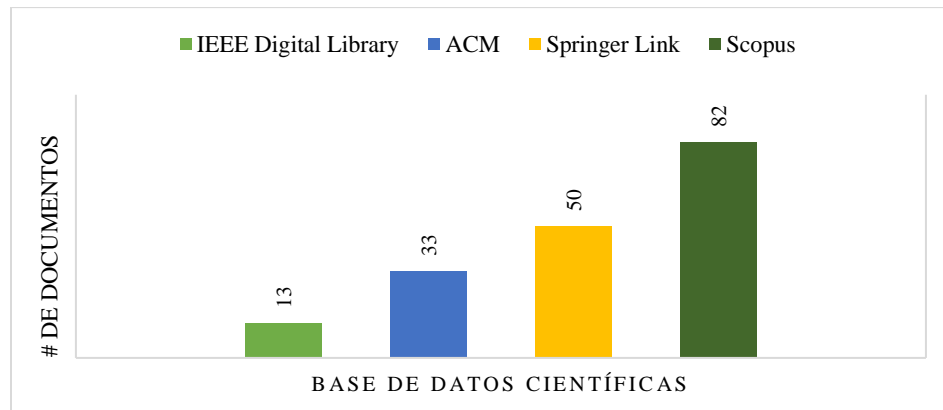
Fases	Tarea
<b>Planificar Revisión</b>	<ul style="list-style-type: none"><li>• Identificar la necesidad de una revisión</li><li>• Especificar preguntas de investigación</li><li>• Desarrollar protocolo de revisión</li></ul>
<b>Realizar Revisión</b>	<ul style="list-style-type: none"><li>• Identificar fuentes/estudios relevantes.</li><li>• Seleccionar estudios primarios</li><li>• Sintetizar datos</li></ul>
<b>Documentar Revisión</b>	<ul style="list-style-type: none"><li>• Escribir informe de revisión.</li></ul>

Las bases de datos científicas seleccionadas para realizar las búsquedas correspondientes se representan en la Tabla 7.

**Tabla 7.** Bases de datos científicas utilizadas para la búsqueda de estudios primarios

<b>Bases de datos Cientificas</b>			
IEEE Digital Library	ACM	Springer Link	Scopus

Tras completar el proceso de selección de estudios primarios, se obtuvieron 178 artículos. La Figura 13 ilustra la distribución de los mismos conforme a cada base de datos científica.



**Figura 13.** Número de artículos encontrados en el proceso de selección de estudios primarios

Los estudios resultantes fueron objeto de tres análisis distintos, mediante los cuales se identificaron y seleccionaron los artículos que cumplían con los criterios de inclusión y exclusión. La principal consideración durante este proceso fue elegir los artículos que contenían información relevante para abordar de manera efectiva las preguntas de investigación planteadas.

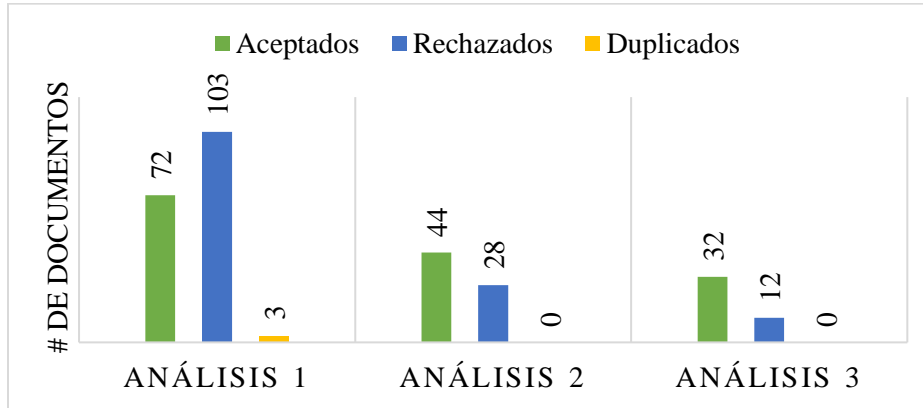
**Tabla 8.** Análisis establecidos para la elección de investigaciones definitivas en la etapa 2 de la revisión sistemática de la literatura (RSL)

<b>Secciones analizadas</b>	
<b>Análisis uno</b>	Título y Resumen
<b>Análisis dos</b>	Introducción, Métodos y Conclusiones
<b>Análisis tres</b>	Lectura crítica de artículos

En la Figura 14 se ilustra la cantidad total de estudios que fueron aceptados, rechazados y duplicados en cada fase de análisis. Los resultados de los artículos identificados en el tercer



análisis, se encuentran documentados en el **Anexo 1**. Cada documento fue exhaustivamente examinado, con el objetivo de discernir si su contenido aportaba de manera significativa en cuanto a modelos de CNN utilizados para la identificación de enfermedades en la hoja de fresa.



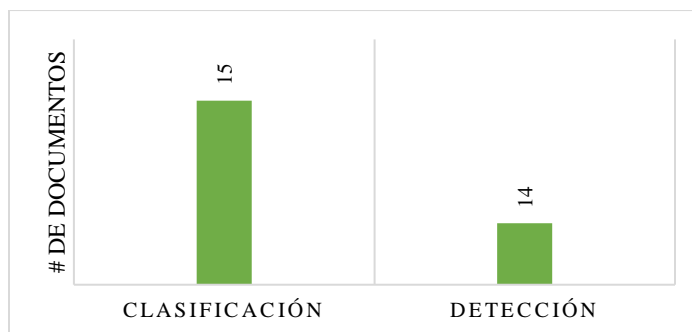
**Figura 14.** Gráfica que muestra la cantidad de artículos admitidos, rechazados y duplicados como resultado de los análisis realizados

Cabe mencionar que dentro de la presente revisión literaria se identificaron algunas de las métricas más utilizadas en el campo de la detección de objetos, y también las características del conjunto de datos empleado para llevar a cabo la tarea de identificar enfermedades en las hojas.

- **PI\_1: Modelos basados en CNN con mayor precisión, empleados hasta la fecha, para identificar enfermedades en las hojas**

En virtud de la RSL realizada, tras un análisis exhaustivo de la documentación pertinente, se ha constatado la existencia de dos enfoques predominantes en la identificación de enfermedades en las hojas de las plantas. Estos, se basan en la utilización de modelos de redes neuronales convolucionales (CNN) tanto para clasificar, como para identificar objetos.

En la Figura 15 se visualiza los resultados obtenidos, donde se indican ambos enfoques en la identificación de enfermedades foliares, observándose que los modelos orientados a la clasificación son los más comunes al momento de llevar a cabo esta tarea.



**Figura 15.** Resultado de los modelos CNN encontrados en los estudios primarios para la identificación de enfermedades foliares

No obstante, en el contexto del presente TT, se ha determinado que la aplicación de modelos de detección de objetos resulta más funcional y adecuada para cumplir con eficiencia los objetivos planteados, por lo cual fueron éstos los seleccionados para su posterior análisis, lo que permitió también tener las bases para el correcto tratamiento de los datos.

La RSL completa se detalla en el **Anexo 1**, donde se pueden revisar a profundidad los resultados obtenidos durante el proceso investigativo.

## **6.1.2 Fase 1: Entendimiento de los datos**

### **6.1.2.1 Recolectar datos iniciales**

El conjunto de datos con imágenes de hojas de fresa con Mancha Angular utilizado para el presente estudio, fue adquirido de marzo a junio del 2023 en el cantón Saraguro, perteneciente a la provincia de Loja, los lugares de investigación fueron una muestra de tres huertos ecológicos pertenecientes a pequeños agricultores de la zona, se recopilaron imágenes en condiciones de clima soleado, nublado, lluvioso, en huertos al aire libre e invernaderos. Todas las imágenes se obtuvieron mediante la técnica de observación, basándose en las características de la enfermedad y bajo supervisión del experto.



**Figura 16.** Áreas experimentales de cultivo de fresa en Saraguro

El recurso utilizado en este proceso de adquisición de imágenes fue la cámara trasera de un dispositivo móvil de marca Samsung A23 cuya descripción se ilustra en la Tabla 9.

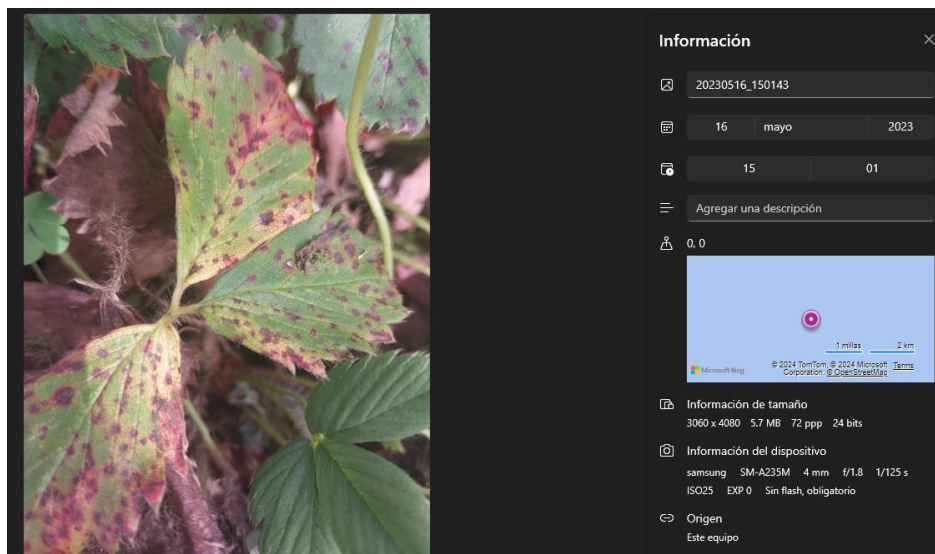
**Tabla 9.** Recursos empleados en el proceso de adquisición de datos

<b>Recurso para la adquisición de imágenes</b>			
	<b>Cámara trasera</b>	<b>Procesador</b>	<b>Memoria RAM</b>
<b>Cámara de teléfono móvil Samsung A23</b>	50 MP f/1.8, OIS Gran angular 5 MP f/2.2 Profundidad 2 MP f/2.4 Macro 2 MP f/2.4 Flash LED	Snapdragon 695 5G GPU Adreno 619	4GB

Como resultado de esta tarea se obtuvo un conjunto de 534 fotografías con una dimensión de 3060 x 4080 pixeles, almacenados en formato JPG. Estas imágenes se tomaron en diferentes condiciones climáticas y de iluminación. En el **Anexo 2**, se puede observar con mayor detalle la información de la práctica realizada para la obtención de imágenes en los diferentes lugares seleccionados como áreas de muestra.



**Figura 17.** Conjunto de imágenes de mancha angular en la hoja de fresa propias



**Figura 18.** Información de imágenes propias

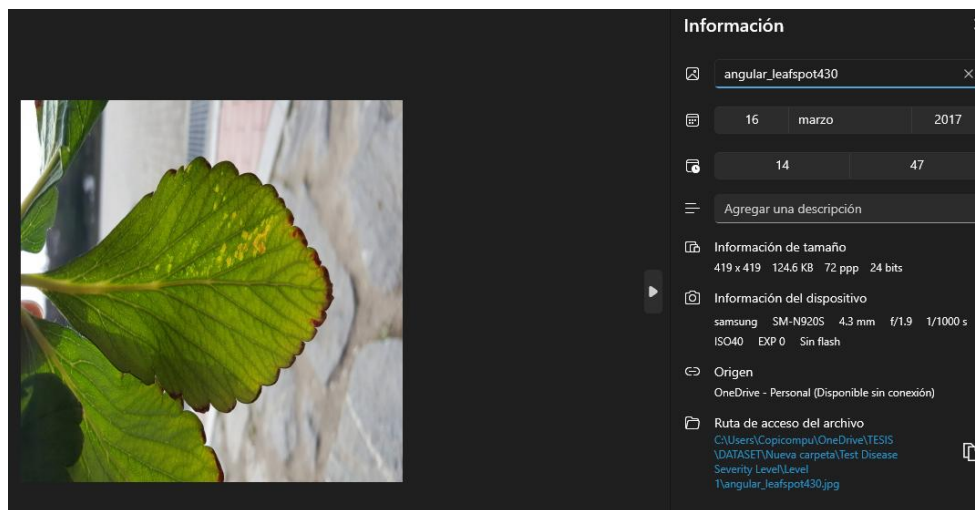
### 6.1.2.1.1 Conjunto de datos de Kaggle

Adicional al conjunto de datos propio, se recopiló imágenes de mancha angular en la hoja de fresa del sitio Kaggle, que es una plataforma en línea que contiene una gran variedad de conjunto de datos y otros recursos relacionados al aprendizaje profundo de manera pública. La base de datos obtenida consta de 2500 imágenes, distribuidas en siete enfermedades de la fresa capturadas en un invernadero, de estas, 435 corresponden a hojas de fresa con mancha angular. Este conjunto de datos contiene etiquetas orientadas a la segmentación de imágenes, por lo cual fue necesario

seleccionarlas y volver a etiquetarlas de acuerdo a los requerimientos de los modelos a implementarse en este proyecto.



**Figura 19.** Conjunto de imágenes del repositorio de Kaggle



**Figura 20.** Información imágenes de Kaggle

### 6.1.3 Fase 2: Preparación de datos

#### 6.1.3.1 Limpieza de datos

Una vez capturadas las imágenes se realizó una inspección minuciosa de cada fotografía con la finalidad de descartar aquellas que no presentaban características claras de la enfermedad. Para llevar a cabo este proceso se tuvo en cuenta los siguientes criterios de inclusión y exclusión:

### Criterios de exclusión de las imágenes

- Imágenes repetidas.
- Imágenes borrosas.
- Imágenes con baja calidad en cuanto a píxeles.

### Criterios de aceptación de las imágenes

- Imágenes claras y de buena calidad.
- Imágenes de hojas de fresa con presencia de mancha angular.

Finalizado este proceso se obtuvo un total de 796 imágenes que cumplieron con el criterio de aceptación, 500 correspondientes al dataset local denominado “*Primario*” y 296 que pertenecen al conjunto de Kaggle denominado “*Secundario*”.

#### 6.1.3.2 Construir los datos (Aumento de Datos (Data Augmentation))

Para esta fase primeramente se apartó el 10% del dataset “*Primario*” que corresponde a 50 imágenes, para conformar el conjunto de pruebas, garantizando una evaluación adecuada del modelo entrenado. A continuación se utilizó técnicas de aumento de datos para generar nuevos datos partiendo de los ya existentes, agrandando el conjunto de datos inicial como lo sugieren los trabajos relacionados descritos en la sección 4.6. Para realizar este proceso se aplicó un script utilizando el lenguaje Python, que se muestra en la Figura 21, el cual permitió incluir diferentes técnicas de aumento de datos con una probabilidad del 50% ( $p=0.5$ ), para evitar que todas las imágenes sufran los mismos aumentos, asegurando así la diversidad de imágenes en el conjunto de datos.

Las técnicas incluidas en este proceso se detallan en la Tabla 10 y fueron aplicadas directamente al conjunto de datos “*Primario*”, dando como resultado un total de 1791 imágenes, Al conjunto de datos “*Secundario*” no se le aplicaron, dado que ya disponía de aumento de datos.

**Tabla 10.** Parámetros para el aumento de datos

<b>Parámetros</b>	<b>Valor</b>
<b>Volteo horizontal</b>	Activado
<b>Brillo</b>	Activado
<b>Rotación</b>	90° y 180°
<b>Saturación</b>	Activado
<b>Zoom</b>	Activado
<b>Ruido gaussiano</b>	Activado



```
augmenter = A.Compose([
    A.HorizontalFlip(p=0.5), # Volteo horizontal
    A.RandomBrightnessContrast(p=0.5), # Brillo y contraste
    A.Rotate(limit=[90, 180], p=0.5), # Rotaciones 90 y 180 grados
    A.HueSaturationValue(hue_shift_limit=0, sat_shift_limit=50, val_shift_limit=0, p=0.5),# Saturación
    A.RandomScale(scale_limit=0.2, p=0.5), # Zoom (escala aleatoria)
    A.Blur(blur_limit=3, p=0.3), # Desenfoque
    A.GaussNoise(var_limit=(10.0, 50.0), p=0.3), # Añadir ruido gaussiano
    A.ShiftScaleRotate(shift_limit=0.1, scale_limit=0.2, rotate_limit=30, p=0.5),
    A.RandomGamma(p=0.5) # Ajuste de gamma
], bbox_params=A.BboxParams(format='pascal_voc', label_fields=['category_ids']))
```

Figura 21 Sección del código para aplicar técnicas de aumento de datos

### 6.1.3.3 Etiquetado de imágenes

Para esta fase se utilizó la plataforma de Roboflow, ya que, como se describió en la sección 4.3, permite un etiquetado más ágil y la exportación del dataset en el formato correspondiente al modelo que se vaya a entrenar. Primeramente se cargaron las 1791 imágenes del dataset “Primario”, posterior a ello se realizó el respectivo etiquetado como se indica en la Figura 22.

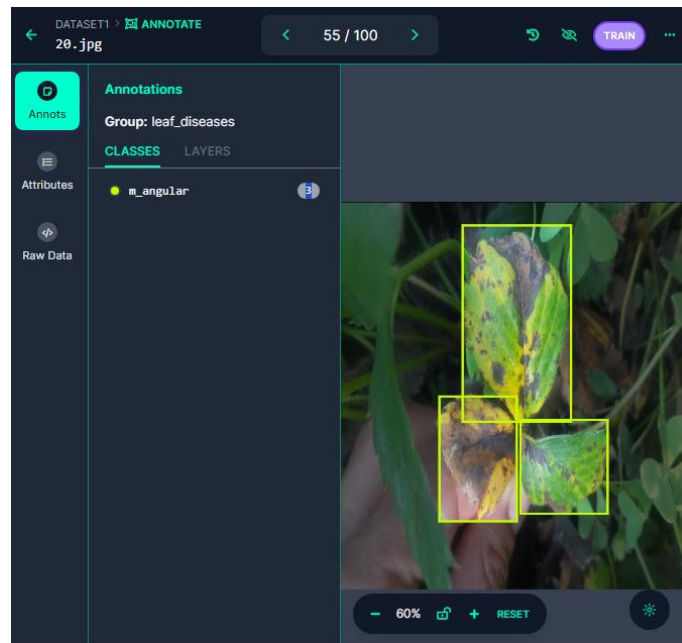
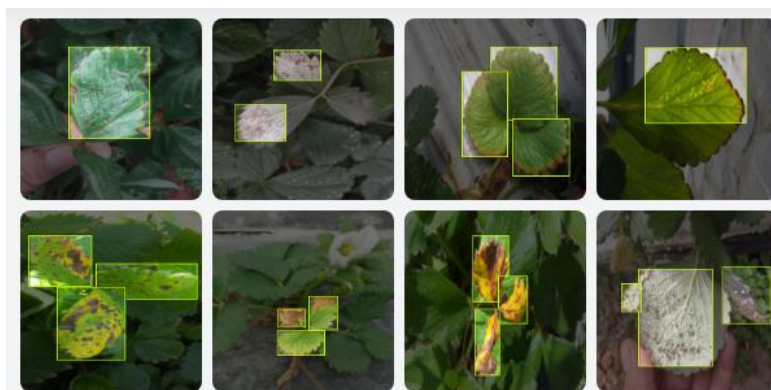


Figura 22. Etiquetado de la enfermedad en Roboflow



**Figura 23.** Conjunto de datos etiquetado

Una vez etiquetado el set de datos “*Primario*”<sup>1</sup>, se pasó a cargar el dataset “*Secundario*” para proceder de la misma manera, estableciendo así un conjunto de datos denominado DATASET-1 con un total de 2087 imágenes y 4345 anotaciones, el cual que fue utilizado para el entrenamiento final del modelo de inteligencia artificial.

**Tabla 11.** Conjunto de datos

<b>Dataset</b>	<b>Enfermedad</b>	<b>Conjunto Inicial</b>	<b>Aumento de datos</b>	<b>Anotaciones</b>
<b>Primario</b>	Mancha Angular	450	1791	
<b>Secundario</b>	Mancha Angular	296	296	
<b>DATASET-1</b>	Mancha Angular	717	2087	4345

#### 6.1.3.4 Formato de etiquetas de imágenes

El uso de Roboflow ha permitido seleccionar de manera directa el formato de exportación del conjunto de datos según sean los requerimientos de cada modelo, en este caso, en el formato “COCO” para Faster RCNN, “YOLO v5 PyTorch” para Yolov5, “YOLO v7 PyTorch” en el caso de Yolov7 y el formato “YOLO v8” para el modelo Yolov8.

<sup>1</sup> Conjunto de datos: [https://drive.google.com/drive/folders/1PJE7En-aO4kdS-aeVT\\_vhqNBpMoRkUws?usp=drive\\_link](https://drive.google.com/drive/folders/1PJE7En-aO4kdS-aeVT_vhqNBpMoRkUws?usp=drive_link)



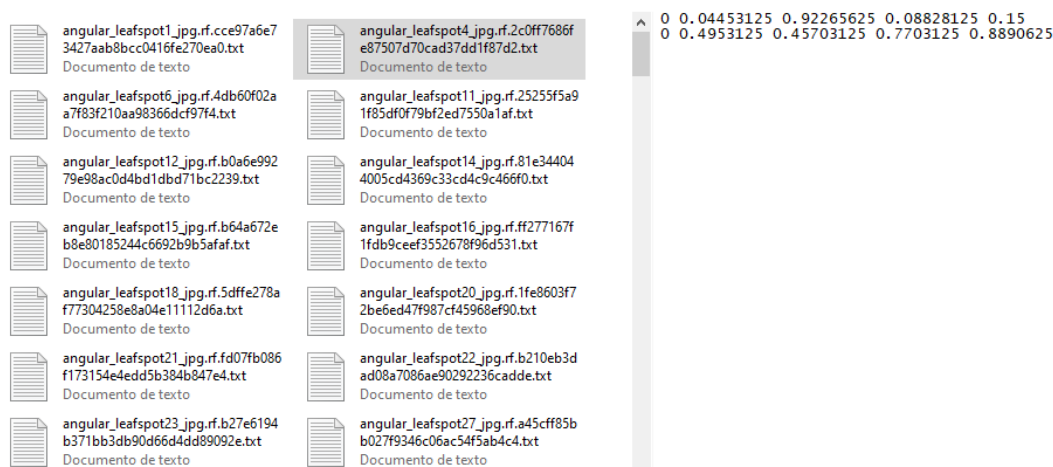
**Tabla 12.** Formato de etiqueta para los modelos Faster RCNN, Yolo v5, v7 y v8

Modelo de detección de Objetos	Formato de etiquetas
FasterRcnn	COCO
Yolov5	YOLO v5 PyTorch
Yolov7	YOLO v7 PyTorch
Yolov8	YOLO v8

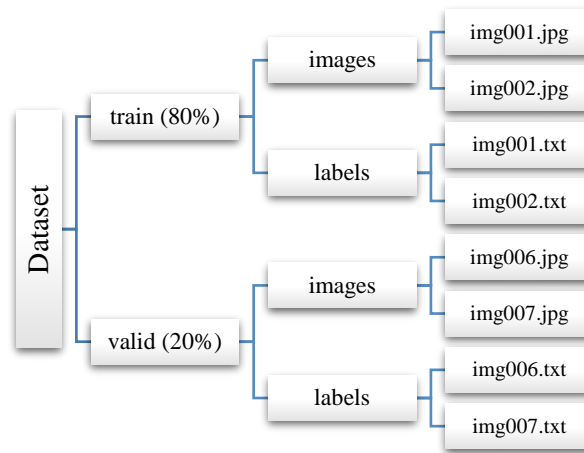
### 6.1.3.5 División del Conjunto de Datos

El conjunto de datos DATASET-1 se organizó en dos subconjuntos: entrenamiento (train) y validación (valid), asignando un porcentaje específico de imágenes a cada uno. De acuerdo con las recomendaciones de trabajos previos (TR11, TR13, TR21), el 80% de las imágenes se destinó al entrenamiento, mientras que el 20% restante se utilizó para validación. Para ello se utilizó la plataforma Roboflow que anteriormente facilitó el etiquetado.

En el formato utilizado por Yolo, se crearon archivos con extensión .txt que almacenaban las etiquetas de cada imagen. Estas etiquetas consistían en cinco parámetros: *id\_clase*, *x\_centro*, *y\_centro*, *ancho (w)* y *alto (h)*, tal como se ilustra en la Figura 24. Debido a que el modelo solo trabajaba con una clase, el *id\_clase* siempre se mantuvo en 0, mientras que las coordenadas variaron según el etiquetado específico de las zonas afectadas por la enfermedad. La estructura del set de datos en este formato se observa en la Figura 25.



**Figura 24** Etiquetas de las imágenes en el formato Yolo

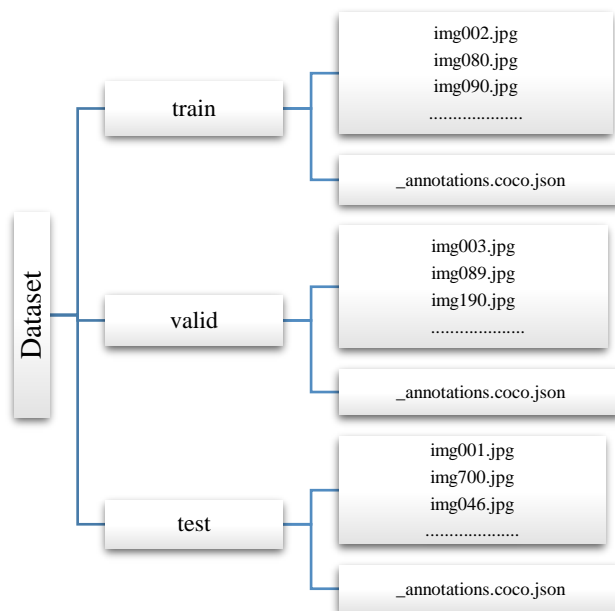


**Figura 25.** Estructura del conjunto de datos en Yolo

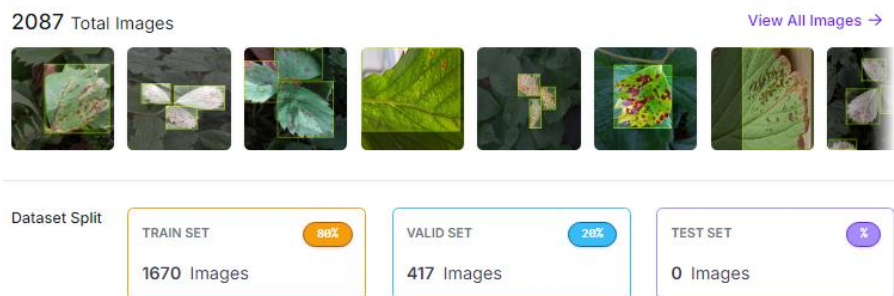
En el formato utilizado por Faster RCNN, las etiquetas de cada imagen se almacenan en un solo archivo *.json*, que incluye información clave como se ilustra en Figura 26. Las categorías están representadas por dos identificadores: *id (0)* para "ManchaAngular", como la categoría principal, el *id (1)* para "m\_angular", una subcategoría de la categoría principal. Además, cada imagen cuenta con atributos como el id único (por ejemplo, *id: 0*), el nombre del archivo (*file\_name*), las dimensiones de la imagen (*height* y *width*), y la fecha de captura(*date\_captured*). La estructura del set de datos en este formato se observa en la Figura 27.



**Figura 26** Etiquetas de las imágenes en formato Faster RCNN



**Figura 27.** Estructura del conjunto de datos en Faster RCNN (*Formato COCO*)



**Figura 28.** División del conjunto de datos DATASET-1

### 6.1.4 Fase 3: Modelado

#### 6.1.4.1 Desarrollo y entrenamiento del modelo

Para llevar a cabo esta fase, se consideró los resultados obtenidos en la RSL, en cuanto a modelos CNN empleados para detectar enfermedades en hojas de las plantas, se identificó que se aplicaron enfoques tanto de clasificación como de identificación de objetos, por lo cual, teniendo en cuenta el contexto del presente Trabajo de Titulación se seleccionó el segundo que corresponde a modelos que permitan detectar que hoja de fresa presenta características de mancha angular.

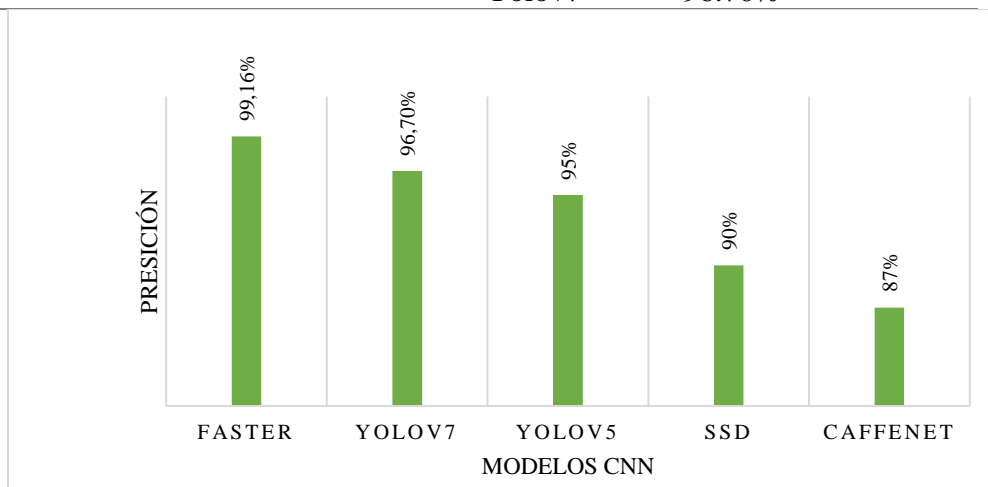
El resultado de este análisis se muestra en la Tabla 13, donde se indican los modelos empleados con el valor de precisión alcanzado. Se eligió tres modelos con los resultados más

sobresalientes, los cuales serán entrenados y evaluados para determinar cuál de ellos es el más adecuado para su posterior implementación, todo esto basado en métricas definidas para la detección de objetos tomadas de la segunda pregunta de investigación en la RSL (Ver Tabla 47. *Métricas para la evaluación de modelos CNN*)

**Tabla 13.** Métricas establecidas por cada modelo identificado en los trabajos analizados (TA) de la RSL

<b>Trabajo Relacionado</b>	<b>Planta</b>	<b># Clases/ Enfermedades</b>	<b>Modelos</b>	<b>Precisión</b>	<b>N° Imágenes</b>
<b>TA04</b>	Hoja de limón	1	Yolov5	67.00%	5475
<b>TA05</b>	Hoja de uva	4	FasterRCNN con:		10 000
			VGG16	98.15%	
			GoogleNet	97.73%	
			ResNet50	99.16%	
			AlexNet	97.76%	
VGG16mejorado	99.60%				
<b>TA07</b>	Anteras de algodón	1	Yolov5 con Darknet	61.88%	38.895
			Faster-RCNN con Resnet 101	87.12%	
<b>TA08</b>	Hojas de tomate	10	Faster-RCNN con:		54,306
			GoogleNet	87.00%	
			Xception	88.00%	
			Resnet-34	99.00%	
<b>TA10</b>	Hojas de fresa	2	Faster-RCNN	55.70%	764
			Yolov3-tiny	69.40%	
			Yolov4-tiny	68.90%	
			Yolov5s	77.50%	
			Yolov4-DAC	75.50%	
<b>TA11</b>	Hojas de fresa	5	Modelo propio	98.00%	1134
<b>TA13</b>	Hojas de café	4	SSD	90.00%	13.692
<b>TA14</b>	Hojas de arroz	13	CaffeNet	87.00%	4,511
<b>TA15</b>	Varias especies de hojas	27	MobileNet	mAP (at 50% iou)=32.8	2,598

			Faster R-CNN- InceptionResnet V2	mAP (at 50% iou)= 38.9	
<b>TA23</b>	Hoja de uva	1	Yolov5-CA	85.59%	820
			FASTER- RCNN	79.97%	
			Yolo v4	82.69%	
			Yolov5	85.35%	
<b>TA25</b>	Hoja de caña	3	Yolov5	95.00%	1680
			Yolor	87.00%	
			DTR	77.00%	
			FasterRCNN	90.00%	
<b>TA26</b>	5 especies de hojas	12	SSD	84.32%	15000
			Faster RCNN	90.67%	
			Yolov3	88.31%	
<b>TA27</b>	Hoja de Té	5	LeNet	90.23%	4000
			Yolov5	95.00%	
			Yolov7	96.70%	



**Figura 29.** Modelos de detección con mejores resultados de precisión según la RSL

Se identificó que los modelos Faster R-CNN, Yolov5 y Yolov7 han presentado resultados con mejor desempeño en medidas de precisión en comparación con SSD y CaffeNet, como se muestra en la Figura 29. Por tal motivo, en el presente proyecto de investigación se decidió entrenar y evaluar estos tres modelos, sumando a estos, la versión 8 de Yolo que es una de sus versiones más recientes, por lo que no existe evidencia de la utilización de este modelo en los estudios analizados.

### 6.1.4.2 Configuración del entorno de entrenamiento

Los modelos seleccionados poseen diferentes arquitecturas, por lo cual su rendimiento fue evaluado mediante métricas basadas en la matriz de confusión. Se entrenó cada modelo con un mismo conjunto de datos y su funcionamiento se validó con el conjunto de prueba, para esto se utilizó la herramienta Google Colab, donde se configuró el acelerador de hardware disponible para aprovechar la potencia de las Unidades de Procesamiento Gráfico (GPU), acelerando el proceso de entrenamiento, lo que permitió un cálculo más rápido y eficiente de las operaciones involucradas durante el entrenamiento de cada modelo.

Los parámetros utilizados en esta etapa incluyen tamaño de la imagen, formato de los datos, número de épocas, tamaño de lote y pesos pre entrenados. Los valores finales se muestran con detalle en la Tabla 14.

**Tabla 14.** Parámetros de entrenamiento

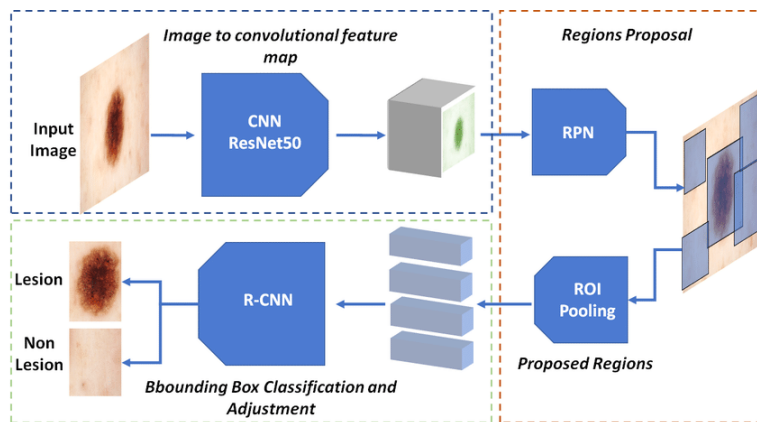
Parámetro	Descripción	Valor
<b>task</b>	Tarea que realiza el modelo	Detect
<b>model</b>	Pesos pre entrenados del modelo	yolov8l.pt yolov5l.pt yolov7_training.pt fasterrcnn_resnet50
<b>data</b>	Formato del conjunto de datos	<i>Yolo</i> : archivo.Yaml .txt <i>Faster RCNN</i> : COCO, .json
<b>epochs:</b>	Épocas de entrenamiento	<i>Yolov5</i> : 200 <i>Yolov7</i> : 200 <i>Yolov8</i> : 200 <i>FasterR-CNN</i> : 200
<b>batch</b>	Tamaño de lote	16
<b>imgsz</b>	Tamaño de la imagen de entrada	640x640 px
<b>optimizer</b>	Optimizador	SGD

### 6.1.4.3 Faster R-CNN

Para la implementación de este modelo se utilizó la guía disponible en Google Colab<sup>2</sup>, mediante el cuaderno denominado “*Faster R-CNN con DETECTRON2 y ResNet50.ipynb*”,

<sup>2</sup> Código de entrenamiento Faster R-CNN con Detectron2: <https://n9.cl/colabfaster>

centrado en el modelo Faster R-CNN implementado en Detectron2., donde se utiliza la Red Residual 50 (ResNet50) como columna vertebral.



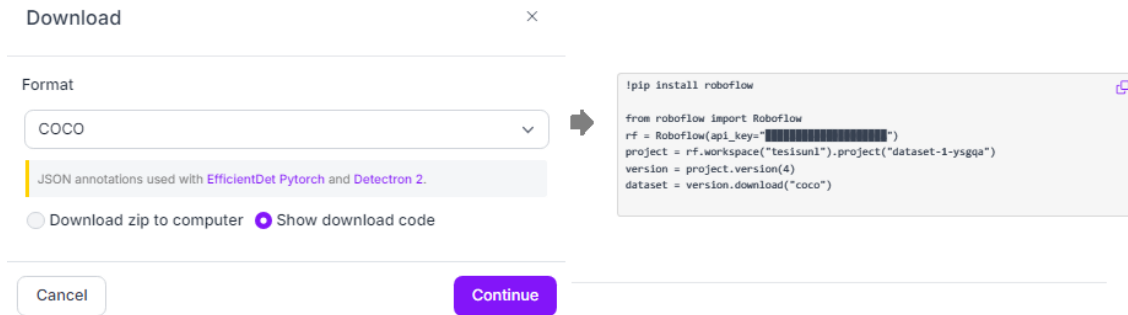
**Figura 30.** Arquitectura de Faster R-CNN

Para llevar a cabo el proceso de entrenamiento se instala en primera instancia la librería Detectron, donde se configura el entorno para el entrenamiento y utilización del modelo, en la Figura 31 se visualiza el código utilizado para este proceso.

```
!python -m pip install pyyaml==5.1
import sys, os, distutils.core
!git clone 'https://github.com/facebookresearch/detectron2'
dist = distutils.core.run_setup("./detectron2/setup.py")
!python -m pip install {' '.join([f'{x}' for x in dist.install_requires])}
sys.path.insert(0, os.path.abspath('./detectron2'))
```

**Figura 31.** Instalación de la biblioteca Detectron2 y sus dependencias

Dado que el modelo Faster R-CNN se basa en conjuntos de datos en formato COCO (Common Objects in Context), se decidió utilizar la herramienta Roboflow para facilitar la descarga del dataset. El snippet de exportación diseñado para Colab, proporcionado por la plataforma, fue posible obtener el conjunto de datos en el formato adecuado. Esto permitió obtener el conjunto de datos en el formato adecuado, facilitando su uso en las etapas de entrenamiento y evaluación del modelo.



**Figura 32.** Adquisición del dataset con Roboflow en el formato COCO

Una vez descargado el conjunto de datos se registran en formato COCO para su uso en Detectron2. Se asoció cada conjunto de imágenes (entrenamiento, validación y prueba) con su respectivo archivo de anotaciones en formato *.json*, esto permitió que Detectron2 los utilice para entrenar el modelo, validar su rendimiento e inferir en los datos de prueba o nuevas imágenes. El código utilizado para este proceso se muestra en la Figura 33.

```

from detectron2.data.datasets import register_coco_instances
register_coco_instances("COCO_train", {}, "/content/DATASET1-2/train/_annotations.coco.json", "/content/DATASET1-2/train")
register_coco_instances("COCO_val", {}, "/content/DATASET1-2/valid/_annotations.coco.json", "/content/DATASET1-2/valid")
register_coco_instances("COCO_test", {}, "/content/prueba--2/test/_annotations.coco.json", "/content/prueba--2/test")

```

**Figura 33.** Código de registro de conjunto de datos en Detectron2

Luego de haber registrado el conjunto de datos se procedió a preparar y estructurar los datos necesarios para el entrenamiento y validación del modelo. El código que se muestra en la Figura 34 permitió a Detectron2 recuperar los metadatos y el diccionario de datos de cada conjunto, accediendo a la información sobre las clases y las anotaciones de cada imagen, lo que permitió al modelo procesar correctamente las imágenes durante la fase de entrenamiento.

```

from detectron2.data import MetadataCatalog
COCO_train_metadata = MetadataCatalog.get("COCO_train")
from detectron2.data import DatasetCatalog
dataset_dicts = DatasetCatalog.get("COCO_train")

```

**Figura 34.** Código de preparación de datos para ser utilizado por Detectron2

A continuación, se realiza el entrenamiento del modelo configurando los respectivos parámetros con los valores de entrada, mismos que se visualizan en la Figura 35.



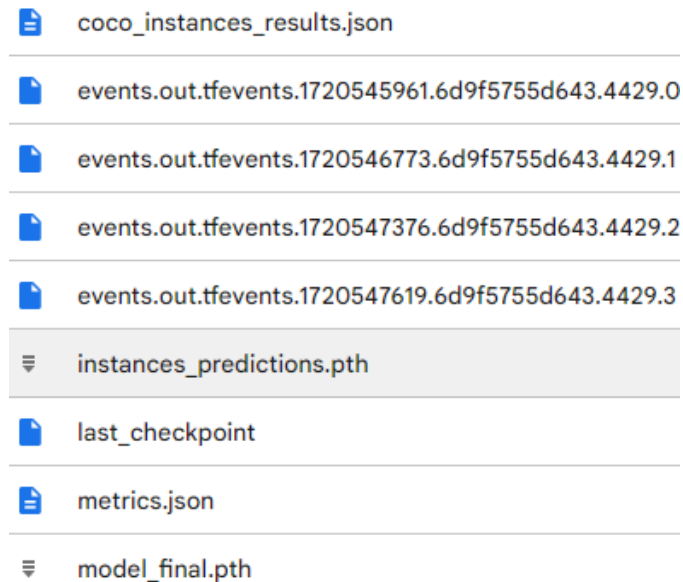
```

11 # Configuración de Faster R-CNN
12 cfg = get_cfg()
13 cfg.merge_from_file("../detectron2_repo/configs/COCO-Detection/faster_rcnn_R_50_FPN_3x.yaml")
14 cfg.DATASETS.TRAIN = ("COCO_train",)
15 cfg.DATASETS.TEST = ("COCO_val",) # Asumimos que este conjunto está definido
16 cfg.DATALOADER.NUM_WORKERS = 2
17 cfg.MODEL.WEIGHTS = "detectron2://COCO-Detection/faster_rcnn_R_50_FPN_3x/137849458/model_final_280758.pkl"
18 cfg.SOLVER.IMS_PER_BATCH = 16
19 cfg.SOLVER.BASE_LR = 0.00025
20 cfg.SOLVER.MAX_ITER = 2000
21 cfg.MODEL.ROI_HEADS.BATCH_SIZE_PER_IMAGE = 128
22 cfg.MODEL.ROI_HEADS.NUM_CLASSES = 2
23
24 # Crear el directorio de salida
25 os.makedirs(cfg.OUTPUT_DIR, exist_ok=True)
26
27 # Entrenamiento del modelo
28 trainer = DefaultTrainer(cfg)
29 trainer.resume_or_load(resume=False)
30 trainer.train()
31

```

**Figura 35.** Subclase para ajustar la evaluación al modelo Faster R-CNN

Mediante la ejecución del código mencionado anteriormente, se realizó el entrenamiento, resultando en la generación de archivos<sup>3</sup> que se muestran en la Figura 36, donde se presentan los resultados de entrenamiento, así como también el archivo “*model\_final.pth*” que es el modelo entrenado, mismo que se utilizó para la las fases de evaluación y pruebas.

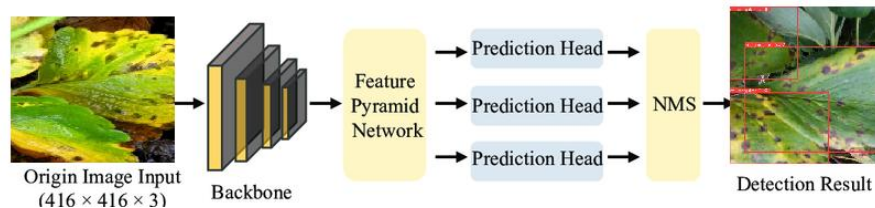


**Figura 36.** Modelo entrenado Faster R-CNN

<sup>3</sup> Resultados de Faster RCNN: <https://n9.cl/resultadosfaster>

#### 6.1.4.4 Yolov5

Esta arquitectura se puede visualizar en la Figura 37. En primer lugar, esta arquitectura toma una imagen como entrada y la procesa a través de un backbone convolucional para extraer características de las hojas con Mancha Angular. Luego, una capa de "cuello" fusiona características de diferentes escalas, y la cabeza de detección emplea múltiples capas convolucionales para predecir las coordenadas de las bounding boxes, las clases de los objetos y las puntuaciones de confianza en tres escalas de detección. Finalmente, la capa de salida proporciona las predicciones finales de detección, lo que permite a Yolov5 detectar si hay presencia de la enfermedad en la hoja.



**Figura 37.** Gráfico de diagrama de capas de Yolov5 entrenado

Para la construcción de este modelo se utilizó el código<sup>4</sup> guía proporcionado por Ultralytics mediante el cuaderno de Google Colab denominado “*Yolov5*”. En primera instancia se procede a descargar el detector de objetos en el cuaderno de trabajo mediante las líneas de código expuestas en la Figura 38.

```
git clone https://github.com/ultralytics/yolov5
```

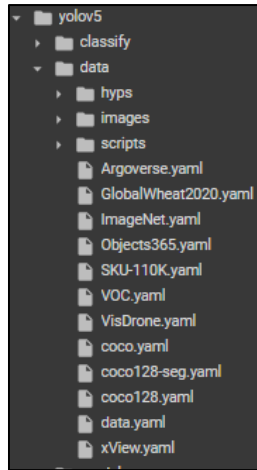
**Figura 38.** Líneas de código ejecutadas para la descarga del modelo pre entrenado Yolov5

El repositorio clonado contiene las configuraciones por defecto del modelo pre entrenado, entre ellos el archivo “*requirements.txt*” donde se ubican los requerimientos en cuanto a versiones de las librerías utilizadas, por lo cual se realiza la instalación de los mismos.

El conjunto de datos que se utilizó para este entrenamiento, de igual manera que con Faster R-CNN, se obtuvo utilizando el código de exportación de Roboflow, en formato “YOLO v5 PyTorch” como se muestra en la Figura 39.

<sup>4</sup>Código de entrenamiento Yolov5: <https://n9.cl/colabyolov5>





**Figura 41.** Configuración del repositorio de YOLOv5

Una vez hecho esto, se procedió a realizar el respectivo entrenamiento, para lo cual se ejecutó la línea de código indicada en la Figura 42, en donde se establecen los valores de entrada para cada parámetro establecidos anteriormente en la Tabla 14.

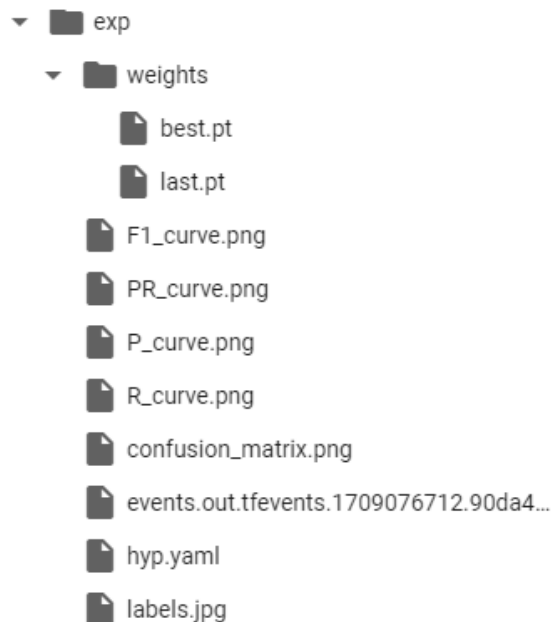
```
!python train.py --img 640 --batch 16 --epochs 200 --data data.yaml --weights yolov5l.pt --cache
```

**Figura 42.** Código para entrenar el modelo de YOLOv5

Una vez terminado este proceso, se obtienen los archivos<sup>5</sup> con los resultados de entrenamiento, así como las métricas de evaluación ya que YOLO durante el entrenamiento realiza la inferencia con los datos de validación para obtener la matriz de confusión y las métricas resultantes de la misma. Los pesos del modelo entrenado se guardaron en la carpeta “weights”, como se muestra en la Figura 43, siendo “best.pt” el que utilizó para realizar la inferencia sobre el set de pruebas pues es el peso con mejor desempeño.

---

<sup>5</sup> Resultados de YOLOv5: <https://n9.cl/yolov5resultados>



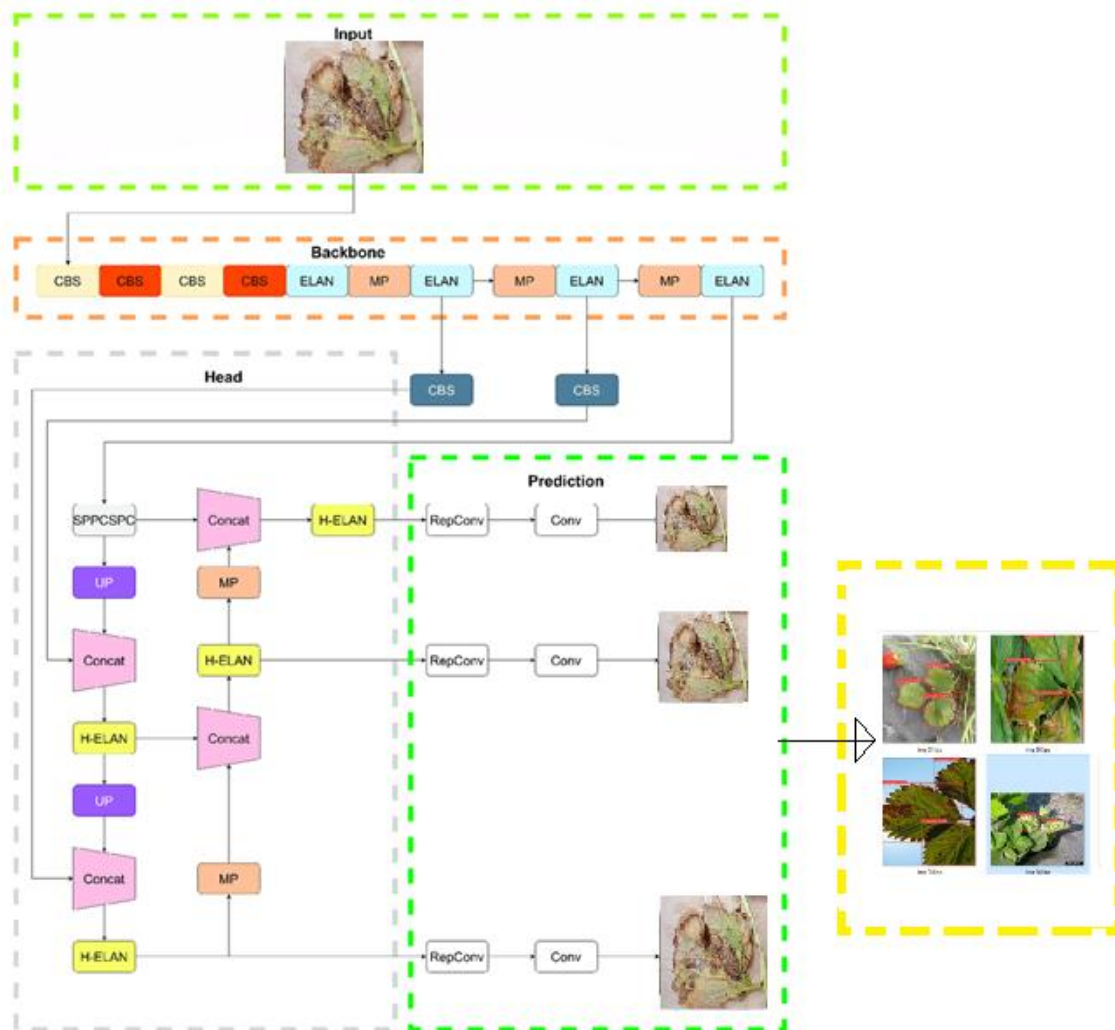
**Figura 43.** Archivos de resultados del modelo YOLOv5

#### 6.1.4.5 YOLOv7

El sitio de Roboflow proporcionó la guía para el entrenamiento mediante el cuaderno con nombre “*Training YOLOv7 on Custom Data*”<sup>6</sup> Este modelo inicia con la entrada de una imagen, pasa por una Espina Dorsal que extrae características, luego estas características se procesan en una Cabeza para generar mapas intermedios, y finalmente, el módulo de predicción produce la salida. La arquitectura es expansible, permitiendo ajustar la profundidad, resolución y ancho del modelo según los requisitos.

---

<sup>6</sup>Código de entrenamiento del modelo YOLOv7: <https://n9.cl/yolov7>



**Figura 44.** Gráfico de capas del modelo Yolov7

Para la implementación de este modelo se siguió la documentación del repositorio oficial. Los archivos se clonaron desde GitHub al cuaderno como lo indica la Figura 45.

```
1 !git clone https://github.com/SkalskiP/yolov7.git
```

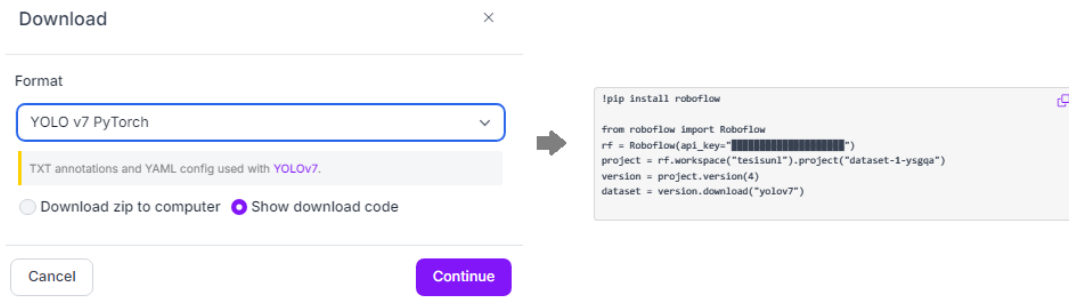
**Figura 45.** Repositorio de yolov7 en Github

A continuación, se instalaron las dependencias en el archivo “requirements.txt” que se encuentra alojado dentro del directorio clonado.

```
!pip install -r requirements.txt
```

**Figura 46.** Instalación de requerimientos para yolov7

Posterior a esto se importa el conjunto de datos desde Roboflow en formato YOLOv7, como se indica en la Figura 47.



**Figura 47.** Importación de Datos desde Roboflow en formato yolov7

A continuación, se descargó los pesos pre entrenados del modelo llamado “*yolov7\_training.pt*” dentro de la carpeta “*yolov7*” (Ver Figura 48) y se procede a ejecutar la línea de código para el respectivo entrenamiento como se muestra en la Figura 49, debido a los recursos limitados de la GPU, se entrenó este modelos con 100 épocas.

```
!wget https://github.com/WongKinYiu/yolov7/releases/download/v0.1/yolov7_training.pt
```

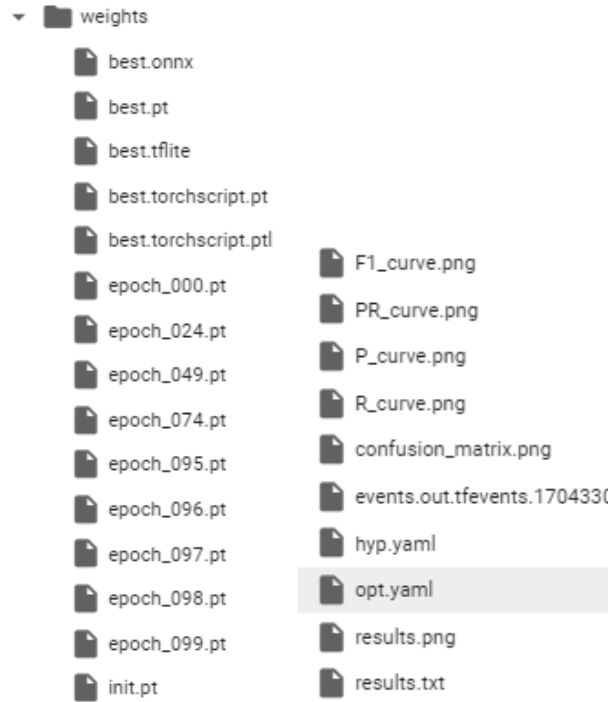
**Figura 48.** Código de descarga de los pesos pre entrenados del modelo YOLOv7(*yolov7\_training.pt*)

```
!python train.py --batch 16 --epochs 100 --data /content/yolov7/DATASET-1-2/data.yaml --weights 'yolov7_training.pt' --device 0
```

**Figura 49** Línea de código para el entrenamiento del modelo YOLOv7

Como resultado de la ejecución del código anteriormente descrito, se obtuvieron los archivos<sup>7</sup> con los valores resultantes de la evaluación en cuanto a métricas de matriz de confusión sobre el rendimiento del modelo, en la carpeta “*weights*” se guardan los pesos entrenados, para la fase de pruebas se utilizó el peso con mejor desempeño denominado “*best.pt*”. Lo archivos resultantes se muestra en la Figura 50.

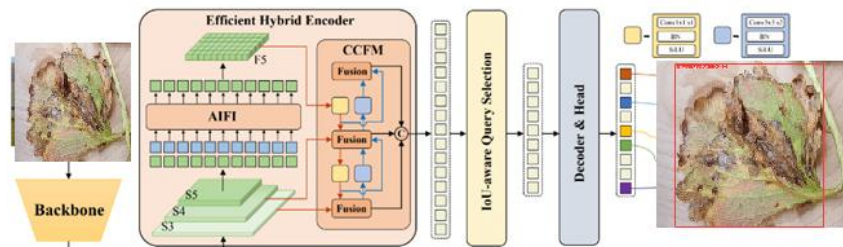
<sup>7</sup> Resultados YOLOv7: <https://n9.cl/archivosyolov7>



**Figura 50.** Archivos resultantes del entrenamiento del modelo Yolov7

### 6.1.4.6 Yolov8

La Figura 51 indica de manera gráfica el diagrama de la arquitectura del modelo Yolov8, mismo que está compuesto por tres partes principales: Entrada, en donde se ajusta la imagen a un tamaño específico, en este caso 640x640 píxeles; Backbone, que son los filtros que pasan sobre la imagen para extraer características básicas como colores y bordes, utiliza redes profundas como CSPDarknet; continua por el siguiente bloque del Neck donde se refinan las características de cada imagen, para lo cual implementa arquitecturas como FPN (Feature Pyramid Network) y PAN (Path Aggregation Network) mejorando la detección de objetos independientemente del tamaño, para finalmente realizar las predicciones finales, encerrando la hoja enferma en cuadros delimitadores y posteriormente asignarlos a la clase correspondiente.



**Figura 51.** Diagrama de capas de Yolov8

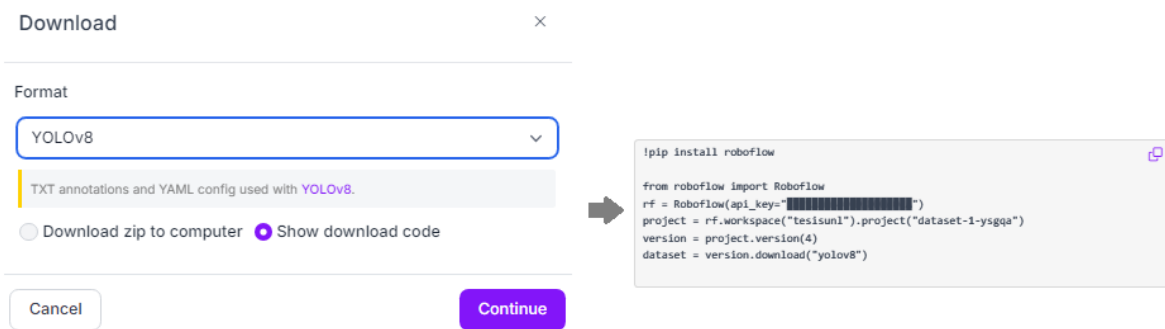


Para llevar a cabo la implementación de esta versión de Yolo, se empleó la guía proporcionada por Roboflow<sup>8</sup>. Inicialmente se llevó a cabo la instalación de la librería de Ultralytics ( Ver Figura 52).

```
1 !pip install ultralytics
```

**Figura 52.** Código de instalación de la librería Ultralytics

A continuación, se realizó la importación directa del conjunto de datos en formato YOLOv8 desde Roboflow.



**Figura 53.** Código de importación de Dataset de roboflow en formato YOLOv8

Posteriormente, se ejecutó el código de entrenamiento configurando los parámetros que se establecieron en un inicio, con el fin de poder establecer una comparativa de resultados basada en parámetros similares. En la Figura 54 se muestra línea de comando aplicado para el respectivo entrenamiento.

```
!yolo task=detect mode=train model=yolov8l.pt data=/content/DATASET-1-2/data.yaml epochs=200 imgsz=640 batch=16 project=/content/ name=propio
```

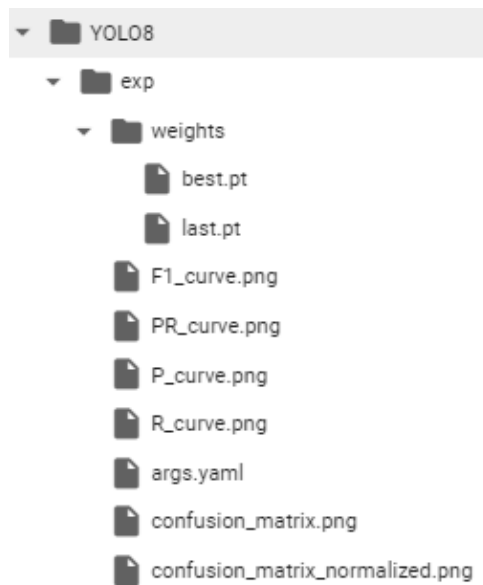
**Figura 54.** Código de entrenamiento del modelo YOLOv8

Finalmente se obtuvo los archivos<sup>9</sup> con los resultados del entrenamiento y las métricas de evaluación como se indica en la Figura 55, los pesos de entrenamiento se guardan en la carpeta

<sup>8</sup>Código de entrenamiento del modelo YOLOv8: <https://n9.cl/colabyolov8>

<sup>9</sup> Resultados YOLOv8: <https://n9.cl/yolov8resultados>

“weights”, para el proceso de pruebas se utilizó como en los casos anteriores el peso con el mejor desempeño “*best.pt*”.



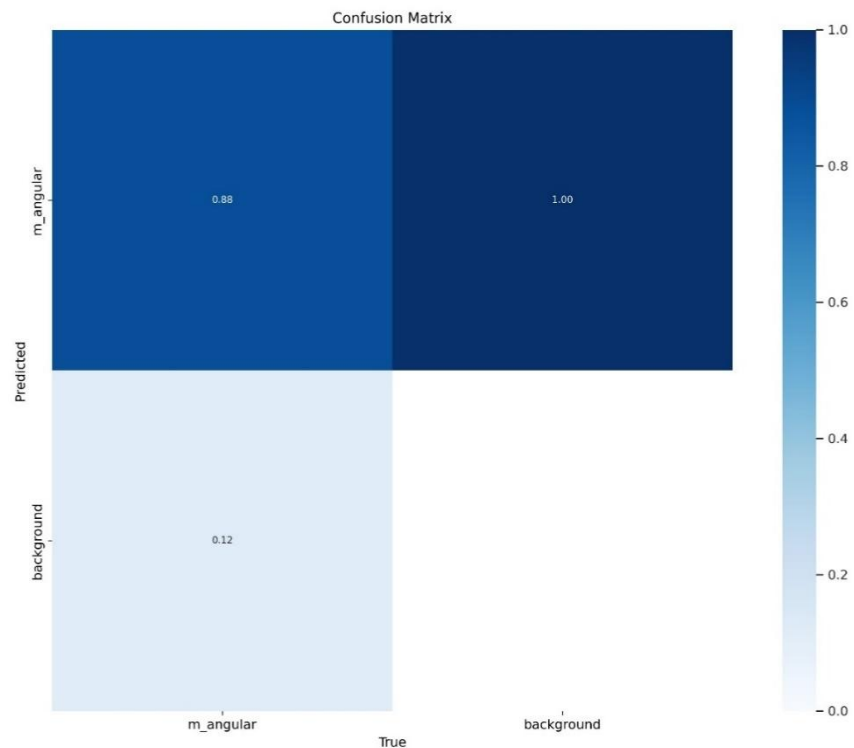
**Figura 55.** Archivos resultantes del entrenamiento del modelo Yolov8

#### 6.1.5 Fase 4: Evaluación de los modelos

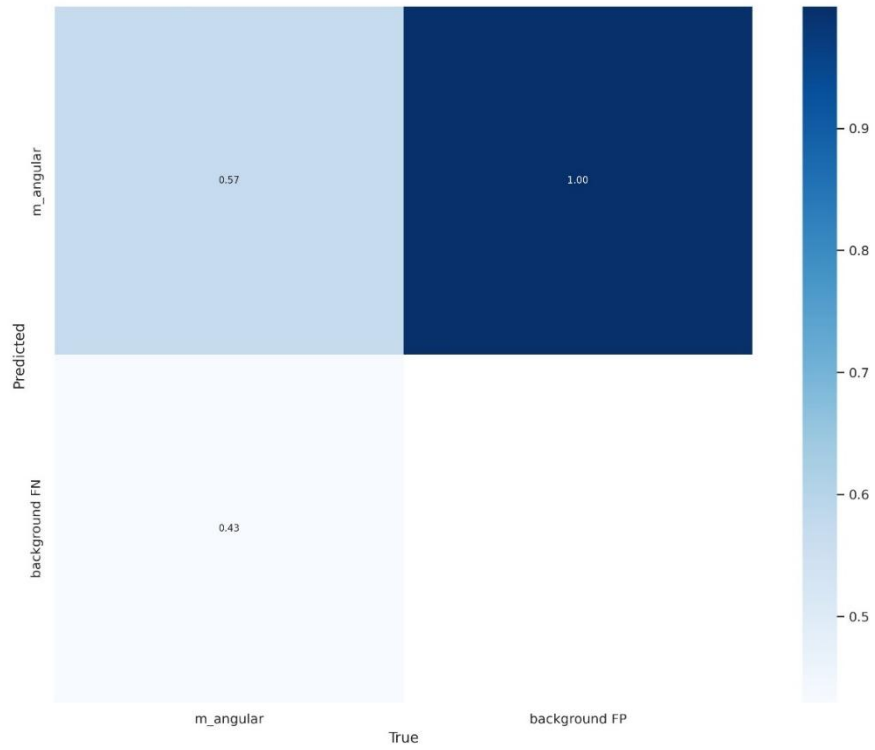
En este apartado, se realizó un análisis comparativo entre los modelos entrenados con los parámetros mencionados en la sección 6.1.4.2, Tabla 14. Parámetros de entrenamiento, evaluando los resultados en métricas de precisión, sensibilidad y puntuación F1.

Se analizó primeramente la matriz de confusión normalizada de los modelos, en donde se recalca que al trabajar con una única clase denominada “m\_angular”, se muestra la precisión correspondiente a cada modelo en la parte superior izquierda de la matriz de confusión, específicamente en la celda m\_angular/m\_angular, estos valores representan los casos reales (Verdaderos Positivos). En la parte inferior izquierda, la celda Background/m\_angular indica el error de predicción, es decir, las instancias de la clase “m\_angular que fueron clasificadas incorrectamente como “Background”. En la parte superior derecha, la celda m\_angular/Background muestra un valor de 1.00, indicando el background, que corresponde a la parte que no está detectando y que no debería ser detectada, representa la porción del entorno que el modelo ignora correctamente porque no forma parte del área que se pretende detectar, si hubiera más clases este valor que es 1.00 varía, pero para esta investigación se consideró una clase como lo es “m\_angular”.

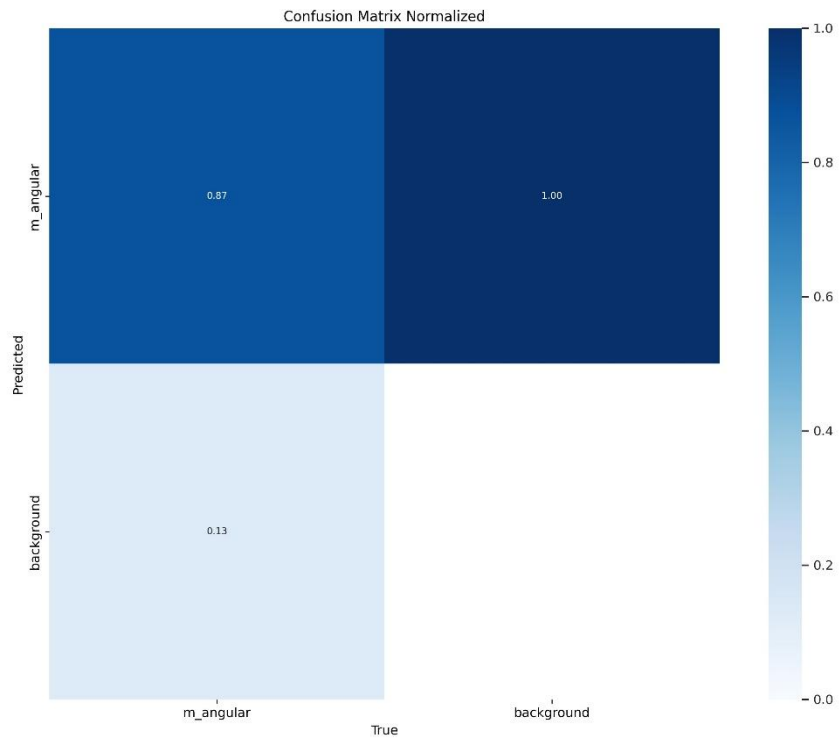
En los resultados obtenidos, el modelo Yolov5 mostró un desempeño bastante sólido, logrando identificar correctamente la clase “m\_angular” en un 88% de los casos. Sin embargo, hubo un 12% de instancias que el modelo confundió con el fondo de la imagen. Por otro lado, Yolov7 tuvo un rendimiento mucho más bajo, acertando solo en el 57% de las detecciones, mientras que en el 43% de las imágenes no logró reconocer ningún objeto. En cuanto a Yolov8, sus resultados fueron bastante cercanos a los de Yolov5, con un 87% de aciertos y un 13% de errores al identificar la clase “m\_angular”.



**Figura 56.** Matriz de confusión del modelo Yolov5



**Figura 57.** Matriz de confusión del modelo YOLOv7



**Figura 58.** Matriz de confusión del modelo YOLOv8

Seguidamente se compararon los resultados obtenidos durante el entrenamiento de cada uno de los modelos, estos valores se miden en términos de precisión, sensibilidad, puntuación F1 y mAP las cuales se detallan en la Tabla 15.

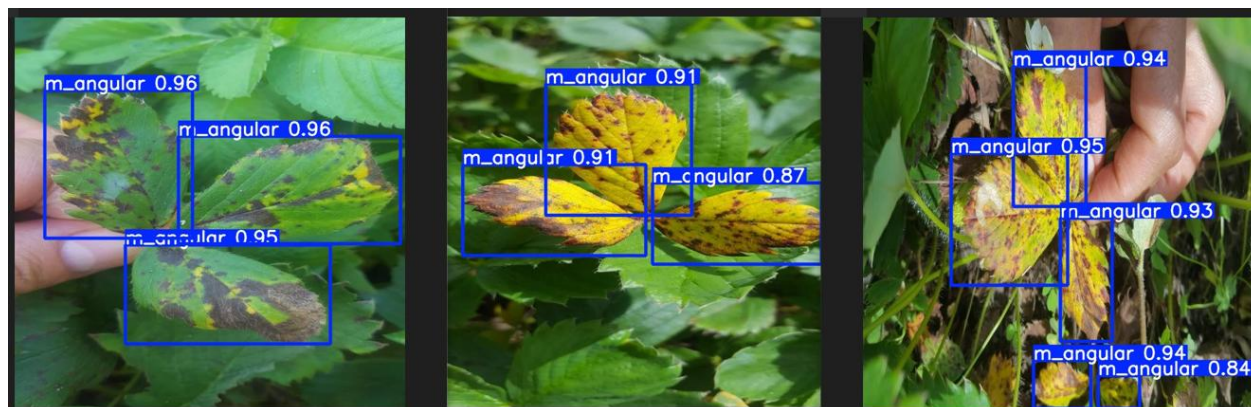
**Tabla 15.** Resultados de entrenamiento de los modelos de detección

<b>Modelo</b>	<b>Precisión</b>	<b>Sensibilidad</b>	<b>Puntuación F1</b>	<b>mAP @0.5</b>
<b>Yolov5</b>	0.93	0.95	0.88	0.92
<b>Yolov7</b>	0.89	0.99	0.70	0.75
<b>Yolov8</b>	0.92	0.94	0.84	0.86
<b>Faster RCNN</b>	0.82	0.81	0.81	0.57

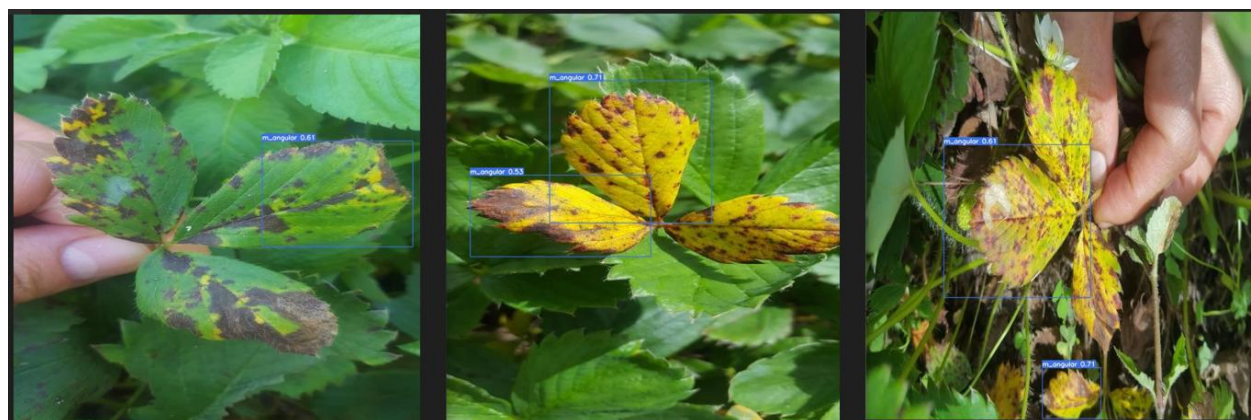
- En cuanto a la *precisión*, que mide cuántas de las predicciones positivas del modelo fueron realmente correctas, Yolov5 sobresalió con un 93%, seguido muy de cerca por Yolov8 con un 92%. Yolov7 obtuvo un 89%, mientras que Faster R-CNN se quedó atrás con un 81%. Esto indica que, en términos de precisión, Yolov5 es el más confiable para detectar la mancha angular en las hojas de fresa.
- La *sensibilidad o recall*, que nos dice cuántos de los casos positivos reales fueron correctamente identificados por el modelo, mostró que Yolov7 fue el más eficiente, con un 99%, aunque Yolov5 y Yolov8 también mostraron muy buen desempeño con un 95% y 94% respectivamente. Faster R-CNN, en cambio, obtuvo solo un 81%, lo que indica una eficiencia aceptable para identificar todos los casos positivos. A pesar de que Yolov7 tiene la mayor sensibilidad, tanto Yolov5 como Yolov8 también se destacan en la identificación de hojas enfermas.
- La *puntuación F1*, que equilibra la precisión y la sensibilidad, muestra que Yolov5 sigue siendo el más fuerte con un 88%, mientras que Yolov8 le sigue con un 84%. Yolov7 obtuvo un 70%, y Faster R-CNN, nuevamente, tuvo el rendimiento más bajo con un 81%, indicando que este último tiene dificultades para equilibrar ambas métricas.
- Por último, el *mAP (@0.5)*, que mide la precisión promedio del modelo en varios niveles de confianza, indicó que Yolov5 es el mejor, con un 92%, seguido por Yolov8 con un 86%. En contraste, Yolov7 alcanzó un 75%, y Faster R-CNN que logró con un 57%, reflejando un rendimiento global limitado.

Los resultados muestran que Yolov5 es el modelo más eficiente y confiable. No solo es preciso en la detección de hojas con mancha angular, sino que también identifica la mayoría de los casos positivos reales, lo que se refleja en su alta sensibilidad. Además, su buen equilibrio entre precisión y sensibilidad, junto con su destacado mAP, lo posicionan como el mejor modelo para esta tarea.

Para constatar que el modelo está generalizando correctamente y es capaz de realizar predicciones precisas en nuevos datos, se realizó la inferencia con un conjunto de 50 datos de pruebas separadas inicialmente del dataset *Primario*, los resultados de manera resumida se visualizan en la Tabla 16. En el **Anexo 4** se muestra con detalle una tabla comparativas<sup>10</sup> de los resultados de cada modelo implementado, con los valores correspondientes al porcentaje de precisión de cada predicción, número de instancias reales y las detectadas.



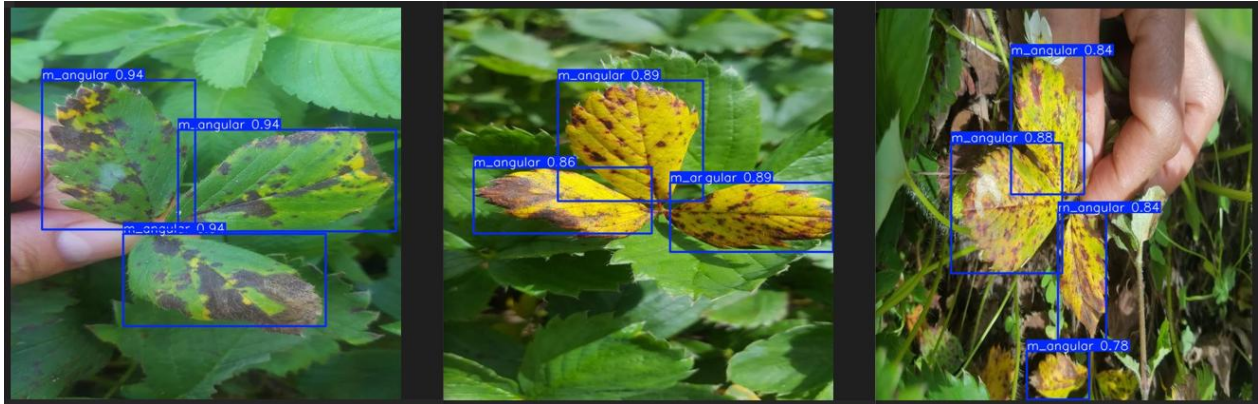
**Figura 59.** Predicciones Yolov5



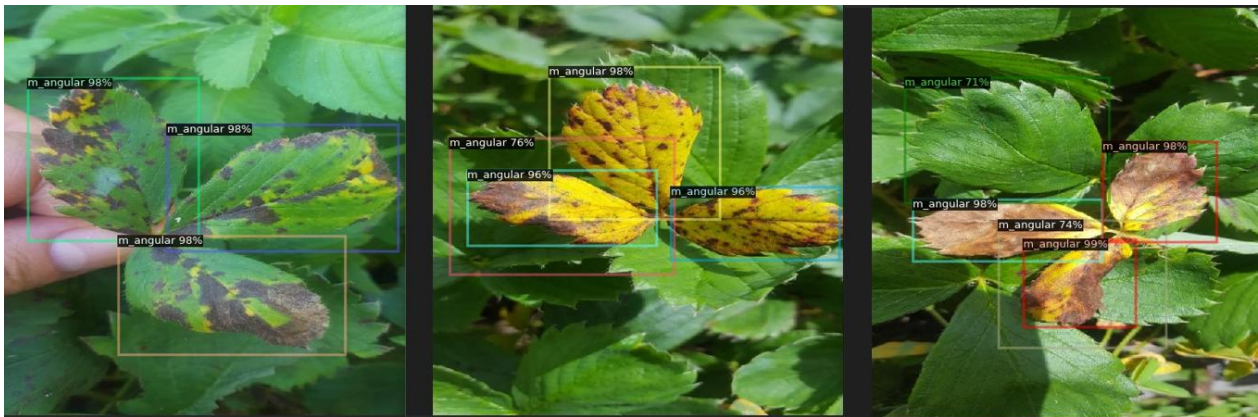
**Figura 60.** Predicciones Yolov7

<sup>10</sup> Tabla comparativa de resultados de modelos implementados: <https://n9.cl/inferencias>





**Figura 61.** Predicciones YOLOv8



**Figura 62.** Predicciones con Faster R-CNN

**Tabla 16.** Comparación de los resultados de predicción de los modelos implementados en el conjunto de datos de prueba

Modelo	Predicciones Correctas	Predicciones Incorrectas	Objetos sin etiquetar
Yolov5	100	4	7
Yolov7	44	3	63
Yolov8	96	7	11
Faster R-CNN	98	46	9

De acuerdo a los resultados, se puede determinar que de los 107 objetos presentes en el set de prueba<sup>11</sup>:

Al evaluar los resultados de inferencia sobre el conjunto de datos de prueba presentados en la Tabla 16, se observó una notable diferencia entre los modelos probados (YOLOv5, YOLOv7, YOLOv8 y Faster R-CNN) en cuanto a su capacidad para detectar correctamente la mancha angular en las hojas del cultivo de fresa.

- **Yolov5:** Identificó correctamente 100 instancias, mientras que solo cometió 4 errores y dejó sin etiquetar 7 objetos. Lo cual indica un buen equilibrio entre precisión y cantidad de detecciones correctas.
- **Yolov7:** el desempeño de este modelo resultó ser considerablemente más bajo. Aunque cometió solo 3 errores, dejó sin etiquetar 63 objetos. Estos resultados indican que este modelo tiene problemas para detectar las hojas enfermas.
- **Yolov8:** de acuerdo a sus resultados tuvo un rendimiento intermedio, realizó 96 predicciones correctas, pero también cometió 7 errores y dejó sin etiquetar 11 objetos. Aunque su desempeño fue aceptable, se encuentra por debajo de Yolov5 y Faster R-CNN.
- **Faster R-CNN:** Fue capaz de identificar 98 instancias correctamente. Sin embargo, la cantidad de predicciones incorrectas fue considerable (46), aunque solo dejó 9 objetos sin etiquetar. A pesar de los falsos positivos, el resultado de la precisión promedio indica que es capaz de identificar la clase objetivo en la mayoría de los casos.

Una vez realizada la comparación de las inferencias realizadas por cada modelo en el conjunto de datos de prueba, se identificó que, Yolov5 y Faster R-CNN mostraron un mejor balance en términos de verdaderos positivos (detecciones correctas) y promedio de precisión en la detección de mancha angular en la hoja de fresa. Yolov5 es eficiente y mantiene un bajo nivel de errores, mientras que Faster R-CNN alcanza una mejor precisión, pero con un mayor número de falsos positivos (detecciones incorrectas). Por lo tanto, debido a que en esta investigación se busca una detección precisa y consistente de la mancha angular, se ha elegido YOLOv5 para avanzar a la siguiente fase del estudio, ya que mostró un rendimiento confiable en la identificación de la enfermedad, lo que es importante evitar diagnósticos erróneos y tratamientos innecesarios.

---

<sup>11</sup> Conjunto de pruebas en fase de entrenamiento:  
[https://drive.google.com/drive/folders/1Kvsfff70iz08soEnLeuPy9wDVKdus62P?usp=drive\\_link](https://drive.google.com/drive/folders/1Kvsfff70iz08soEnLeuPy9wDVKdus62P?usp=drive_link)

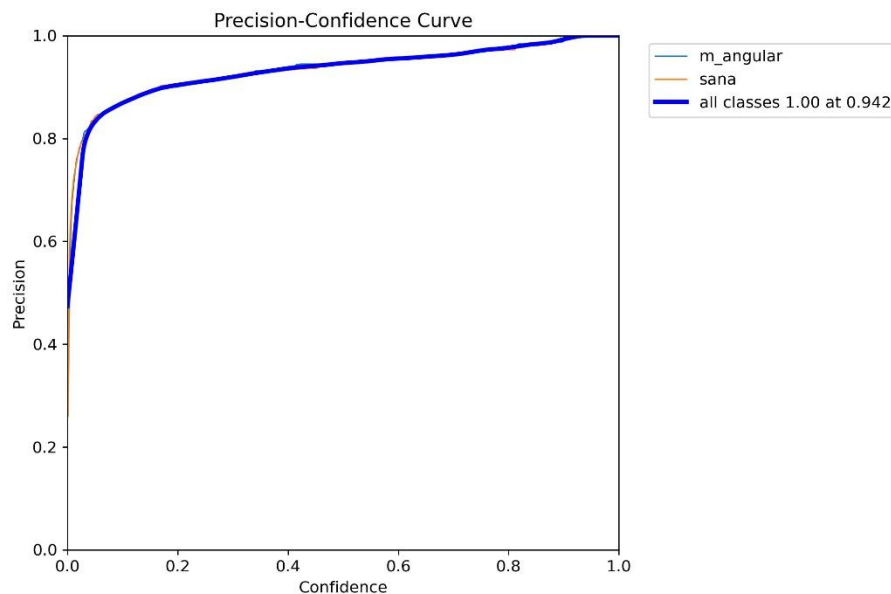


### 6.1.5.1 Experimento de implementación de clases

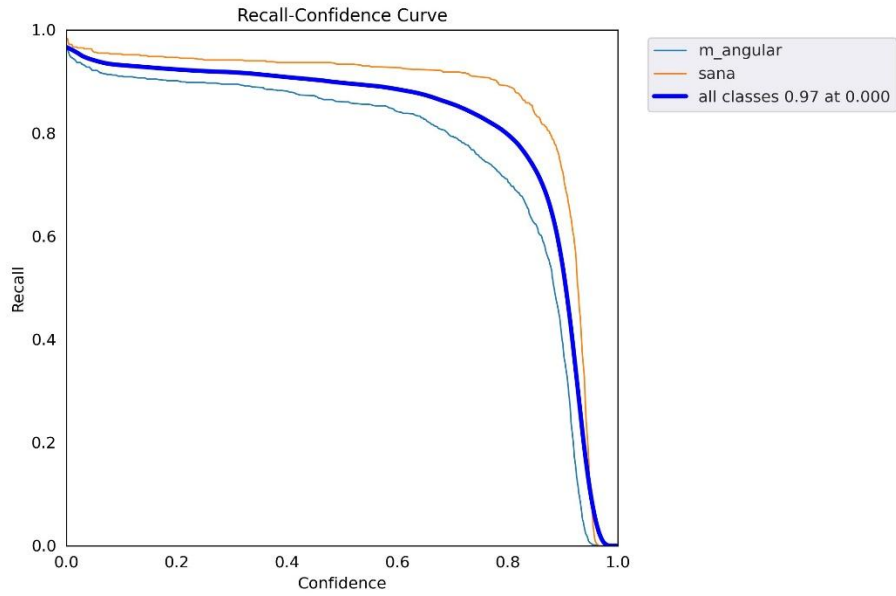
En la búsqueda de mejorar el resultado de las métricas de detección del modelo Yolov5, se experimentó realizando un nuevo entrenamiento, añadiendo una nueva clase denominada “sanas”, la cual, corresponde a ejemplares de hojas saludables en el cultivo de fresa, para ello se añadió 900 imágenes de hojas sanas al set de datos *Dataset-1*, obteniendo los siguientes resultados tanto en parámetros de identificación de objetos, como en matriz de confusión:

**Tabla 17.** Conjunto de datos con dos clases (DATASET-2)

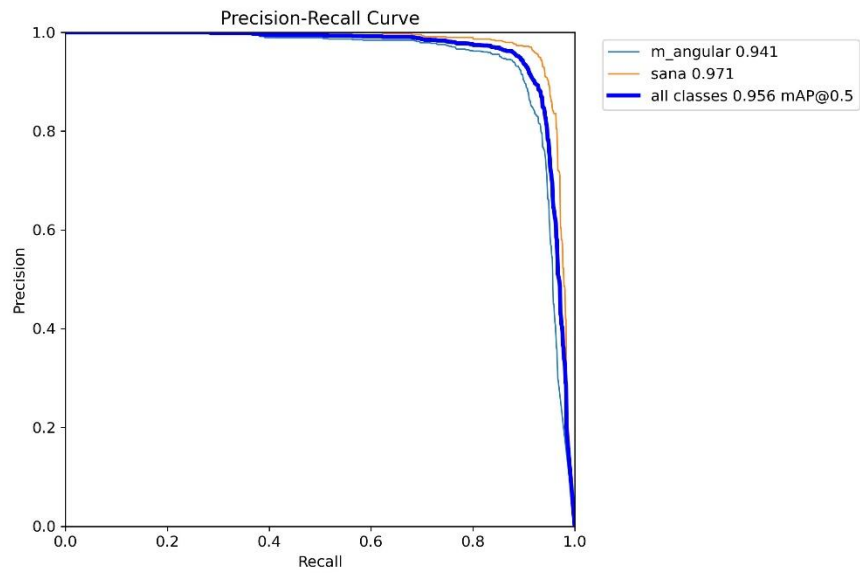
Conjunto de Datos	Total de imágenes
DATASET-1	2087
SANAS	900
<b>DATASET-2</b>	<b>2987</b>



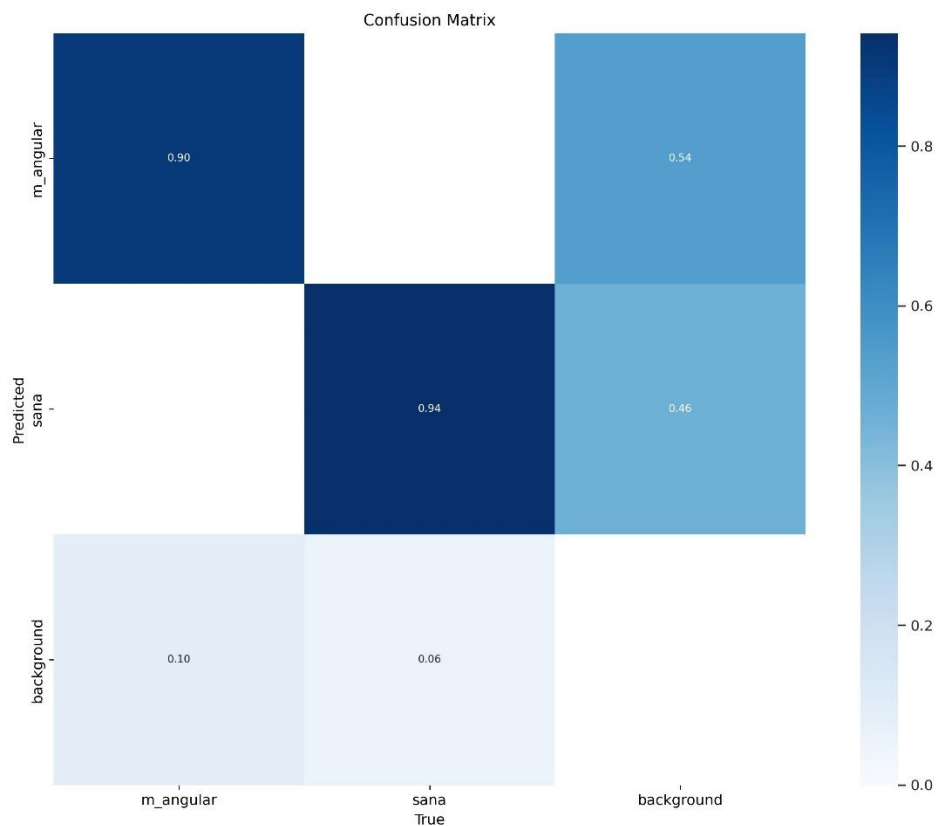
**Figura 63.** Curva de precisión de los resultados en la clase “sana” y “m\_angular”



**Figura 64.** Curva de confianza de la sensibilidad de los resultados en la clase “sana” y “m\_angular”



**Figura 65.** Curva de precisión- sensibilidad de los resultados en la clase “sana” y “m\_angular”



**Figura 66.** Matriz de confusión con la clase "sana" y "m\_angular".

Al comparar las matrices de confusión entre el modelo original, que solo consideraba una clase (m\_angular), y el modelo que incorpora dos clases (sana y m\_angular), se observan mejoras significativas en la capacidad de discriminación.

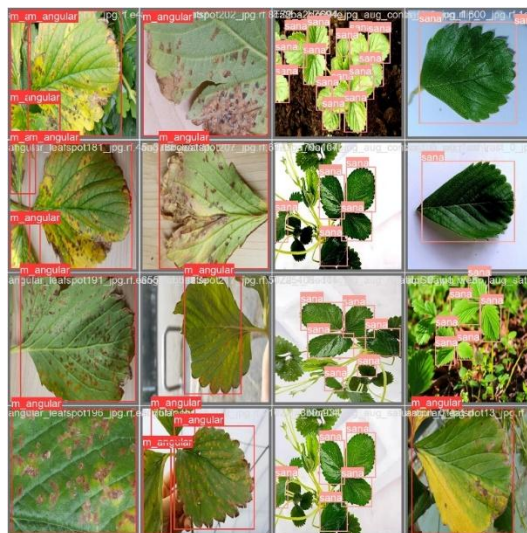
En el primer caso, donde solo se consideraba la clase “m\_angular”, el modelo lograba un aceptable índice de clasificación correcta (0.88), pero presentaba un rendimiento deficiente al clasificar como background un índice de 0.12 en el cuadrante de falsos positivos.

En contraste, en el segundo caso, se logró un aumento sustancial en la capacidad del modelo para detectar mancha angular en las hojas de fresa, alcanzando un índice de verdaderos positivos de 0.90. La clase de hojas sanas presenta un índice de verdaderos positivos de 0.94, es importante destacar que este rendimiento se logró incluso con un conjunto de datos de entrenamiento más limitado para esta clase.

Finalmente se compararon las métricas obtenidas por cada modelo, siendo Yolov5 entrenado con las clases “mancha angular” y “sana” superior.

**Tabla 18.** Comparación de métricas entre el modelo Yolov5 con una clase y Yolov5 con dos clases

Modelo	Clase	Precisión	Sensibilidad	Puntuación F1	mAP
Yolov5	mancha_angular	0.93	0.95	0.88	0.92
Yolov5	Mancha_angular + Sanas	0.94	0.97	0.92	0.95



a)

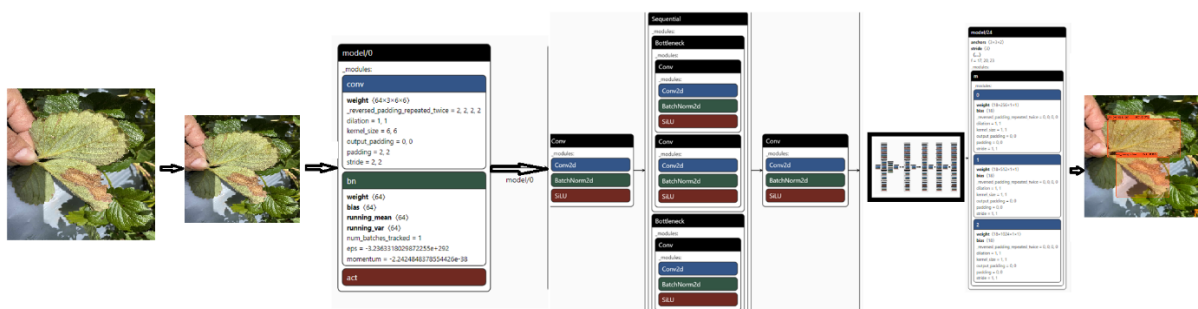


b)

**Figura 67.** Predicciones realizadas por el modelo entrenado con dos clases. a) Etiquetas originales b) Predicciones realizadas por el modelo

En base a los resultados obtenidos se concluye que, la incorporación de múltiples clases y el ajuste del modelo para reconocer hojas sanas, presenta una mejora en la capacidad del modelo para detectar mancha angular en las hojas de fresa, a pesar de las limitaciones en el conjunto de datos de entrenamiento para la clase “sanas”, por lo que no se requiere que todas las hojas sanas de un cultivo sean identificadas, siendo más relevante obtener resultados significativos en la detección de hojas con mancha angular.

En la Figura 68 se puede visualizar la estructura del modelo deep learning finalmente implementado y entrenado con dos clases, mismo que será integrado en un software para mayor facilidad de uso, la estructura<sup>12</sup> completa, así como los resultados de entrenamiento, se encuentran alojados en un repositorio en Google Drive<sup>13</sup>.



**Figura 68.** Estructura de modelo YOLOv5 entrenado

## 6.2 Objetivo 2: Construir un prototipo de software e integrar el modelo de red neuronal utilizando la metodología XP.

### 6.2.1 Fase 5: Implementación

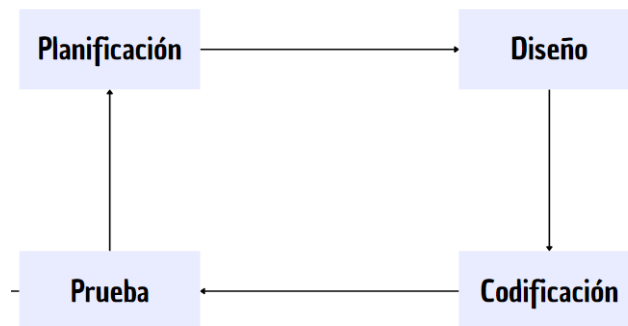
En esta sección se lleva a cabo la última fase de la metodología CRISP-MD, que comprende la construcción de un prototipo de aplicación móvil, que utiliza una Red Neuronal Convolutacional, para el diagnóstico de mancha angular en las plantas de fresa, permitiendo al usuario que a través de una fotografía pueda detectar si hay o no presencia de la infección.

<sup>12</sup>Gráfico de la estructura de YOLOv5 entrenado: <https://n9.cl/yolov5estructura>

<sup>13</sup>Resultados de entrenamiento de YOLOv5 con 2 clases: <https://n9.cl/resultfinal>

## 6.2.2 Metodología de Desarrollo del prototipo de aplicación móvil

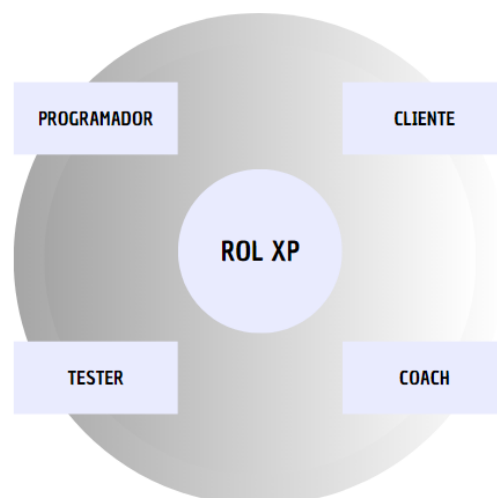
Para llevar a cabo con éxito los restos que conlleva el desarrollo de una aplicación móvil, se siguió la metodología XP orientada al desarrollo de aplicaciones móviles. En la Figura 69 se visualizan las fases de esta metodología y las que serán utilizadas en este proyecto, estas son: planificación, diseño, codificación y pruebas.



**Figura 69.** Fases de la metodología XP aplicadas en la investigación

## 6.2.3 Actores

Inicialmente se establecen los actores, es decir, las personas que participan en el proceso de desarrollo teniendo en cuenta que esta metodología se basa en la colaboración entre los miembros del equipo. En la Figura 70 se muestra los roles que intervinieron en este proyecto.



**Figura 70.** Roles de la metodología XP que son parte del TT

Hay que tener en cuenta que el desarrollo de este proyecto es realizado por una sola persona, por lo cual, los roles definidos en XP fueron ocupados en algunos casos por el asesor del proyecto.

Algunos roles como Consultor y gestor o Big Boss, no fueron tomados en cuenta para este proyecto, porque el tamaño tan reducido del equipo de trabajo no lo permitía.

**Tabla 19.** Descripción de los roles que son parte del desarrollo del TT

<b>ACTOR</b>	<b>DESCRIPCIÓN</b>	<b>ENCARGADO</b>
<b>Programador</b>	Produce el código del sistema y escribe las pruebas unitarias	Diana González
<b>Cliente</b>	Se encarga de escribir las historias de usuario y las pruebas funcionales para validar la implementación del sistema.	Agrónomo
<b>Encargado de pruebas (Tester)</b>	Realiza pruebas regularmente y comparte los resultados con el equipo.	Diana González/Agrónomo
<b>Entrenador (Coach)</b>	Guía y vigila que se sigan correctamente las prácticas de XP, es responsable del proceso global	Director de Tesis

## 6.2.4 Planificación

### 6.2.4.1 Historias de Usuario

En esta fase se generó las historias de usuario, con la finalidad de obtener una comprensión más precisa de los requerimientos del usuario final.

**Tabla 20.** Historia de Usuario N°01 – Integrar modelo entrenado

Historia de Usuario	
Número: 01	Nombre Historia de Usuario: Integrar modelo entrenado
Actor: Usuario	Iteración Asignada: 1
Prioridad: Alta	
Riesgo en Desarrollo: medio	Programador Responsable: Diana González
Descripción: Se desea que el modelo entrenado se integre en la app móvil .	
Observaciones: -	

**Tabla 21.** Historia de Usuario N°02 – Información principal

Historia de Usuario	
Número: 02	Nombre Historia de Usuario: Información principal
Actor: Usuario	Iteración Asignada:2
Prioridad: Alta	
Riesgo en Desarrollo: bajo	Programador Responsable: Diana González
Descripción: Se desea que el usuario obtenga una información previa de la enfermedad a detectar.	
Observaciones: - Esta opción se encuentra en la pantalla principal.	

**Tabla 22.** Historia de Usuario N°03 – Seleccionar foto de Galería

Historia de Usuario	
Número: 03	Nombre Historia de Usuario: Seleccionar foto de galería
Actor: Usuario	Iteración Asignada:3
Prioridad: Alta	
Riesgo en Desarrollo: bajo	Programador Responsable: Diana González
Descripción: El usuario podrá acceder a la galería de imágenes y seleccionar una foto para su detección	
Observaciones: -	

**Tabla 23.** Historia de Usuario N° 04 – Capturar foto con la cámara

Historia de Usuario	
Número: 04	Nombre Historia de Usuario: Capturar foto con la cámara
Usuario: Usuario	Iteración Asignada:3
Prioridad: Alta	
Riesgo en Desarrollo: bajo	Programador Responsable: Diana Gonzalez
Descripción: El usuario podrá abrir la cámara del dispositivo móvil y tomar una foto para su detección	
Observaciones: -	



**Tabla 24.** Historia de Usuario N° 05 – Mostrar Resultado

Historia de Usuario	
Número: 05	Nombre Historia de Usuario: Mostrar Resultado
Actor: Usuario	Iteración Asignada:4
Prioridad: Alta	
Riesgo en Desarrollo: bajo	Programador Responsable: Diana Gonzalez
Descripción: El usuario podrá realizar la detección de la clase mancha angular en la foto seleccionada y ver el resultado del mismo.	
Observaciones: -	

Seguidamente se realizó la priorización de las historias de usuario, en la Tabla 25 se observan los valores otorgados a cada una.

**Tabla 25.** Valores de priorización de las Historias de Usuario

Código de Historia de Usuario	Nombre	Prioridad	Riesgo
HU01	Integrar el modelo entrenado	Alta	Medio
HU02	Información principal	Alta	Bajo
HU03	Seleccionar foto de galería	Alta	Bajo
HU04	Capturar foto con la cámara	Alta	Bajo
HU05	Mostrar Resultado	Alta	Bajo

#### 6.2.4.2 Estimación e Iteración de Historias de Usuario

En esta etapa se estimó el tiempo requerido para la ejecución de cada historia de usuario, una vez concluido se pudo obtener un plan de entregables distribuidos en iteraciones, en el cual participó el programador, el jefe del proyecto(coach) y el cliente.

Se consideró 5 días laborables por semana, estableciendo el día de trabajo equivalente a 8 horas, programando 40 horas semanales como sugiere la metodología de Programación Extrema.

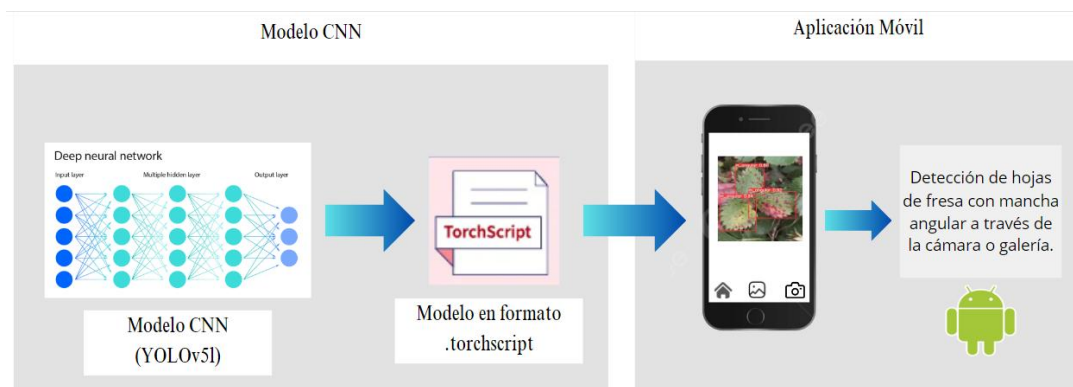
**Tabla 26.** Estimación e iteración de HU

Iteraciones	Historias de Usuario	Día Estimado
Primera Iteración	Integrar el modelo entrenado	5 días
Segunda Iteración	Información principal	5 días
Tercera Iteración	Seleccionar foto de galería	5 días
	Capturar foto con la cámara	5 días
Cuarta iteración	Mostrar Resultado	5 días
<b>Total</b>		<b>25 días</b>

### 6.2.5 Diseño

En la Figura 71 se representa el diseño de la aplicación, la misma que se conforma por dos principales componentes que son: primero, el modelo de red neuronal entrenado y validado previamente, convertido en un formato compatible para dispositivos móviles y segundo, la aplicación móvil.

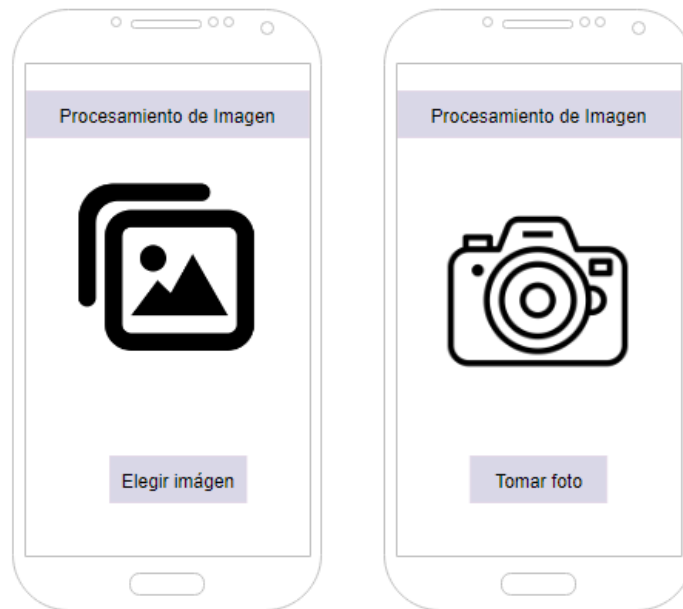
La aplicación móvil hace uso de la red neuronal para realizar el proceso de detección de mancha angular en la hoja de fresa, a través de imágenes localizadas en la galería o utilizando la cámara del dispositivo móvil.



**Figura 71.** Diseño del Prototipo móvil

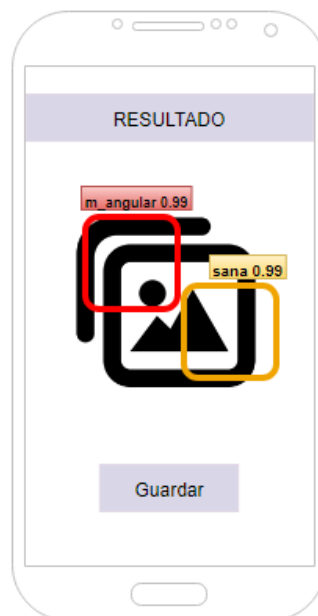
Para el desarrollo del prototipo se utilizó el software gratuito draw.io, que permite de manera sencilla plasmar las ideas para diseñar las interfaces del software. En este prototipado se realizaron diseños básicos y comprensibles para el usuario.

En el primer módulo se realizó los prototipos de “Seleccionar foto de galería” y “Capturar foto con la cámara”, Figura 72.



**Figura 72.** Interfaces de ingreso de imágenes

En el segundo módulo de resultados se realizó el diseño de la visualización de las detecciones obtenidas por el modelo de detección, como se indica en la Figura 73.



**Figura 73.** Interfaz de vista de resultados

## 6.2.6 Codificación

### 6.2.6.1 Recursos de Software

Los recursos de software son los programas utilizados para el desarrollo de la aplicación móvil, mismo que se describen en la Tabla 27.

**Tabla 27.** Recursos software

Nombre	Descripción	Versión
Visual Studio	Entorno de desarrollo integrado	3.2.0
Flutter	Framework de desarrollo de código abierto	
Celular Samsung	Emulador físico	A23

### 6.2.6.2 Primera Iteración

#### - HU01- Integrar el modelo entrenado

Esta etapa inició con la conversión del modelo CNN a un formato compatible. Dado que se utilizó el framework de código abierto Flutter para la integración del modelo de detección en un entorno móvil, se debió realizar la conversión al formato “torchscript”, ya que esto permite integrarlo de manera más sencilla en la aplicación, además, minimiza la dependencia de PyTorch, lo que reduce el tamaño de la aplicación y hace más fácil su distribución. Para ello, como se muestra en la línea de código de la Figura 74, se utilizó el script Python “*export.py*”, también se definió la ruta del archivo de pesos del modelo entrenado “*best.pt*” generada al finalizar el entrenamiento y se indica el formato de exportación del modelo “*--include torchscript*”. En la Figura 75 se puede observar el modelo original entrenado y el modelo convertido, generado luego de la ejecución del código anterior.

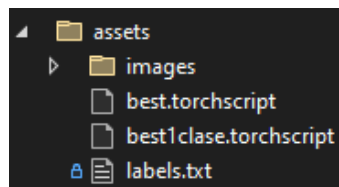
```
!python export.py --weights /content/best.pt --include torchscript
```

**Figura 74.** Conversión del modelo al formato torchscript



**Figura 75.** Pesos del modelo entrenado y convertido a torchscript

Una vez convertido el modelo, se crea una carpeta denominada “assets” (Ver Figura 76) dentro del proyecto de la Aplicación móvil, esta carpeta esta designada para almacenar archivos estáticos que serán utilizados en lo posterior, en este directorio se ubica el modelo convertido a *.torchscript*, entre otros recursos importantes como: el archivo *labels.txt*, que contiene las clases correspondientes al modelo, y la carpeta “imágenes”, que alberga las imágenes utilizadas en la aplicación. El código completo del proyecto se encuentra alojado en un repositorio de GitHub<sup>14</sup>.



**Figura 76.** Integración del modelo entrenado a la App móvil

Para la integración del modelo se utilizó una función asincrónica llamada `loadModel()` que se encarga de cargar el modelo “best.torchscript” desde la carpeta “assets”, utilizando la biblioteca `FlutterPytorch`, que recibe como parámetros la ruta del modelo, el número de clases del modelo, el tamaño de la imagen de entrada para la detección de objetos, y la ruta del archivo que contiene las etiquetas correspondientes a las clases del modelo (Ver Figura 77).

```
Future loadModel() async {
  String pathObjectDetectionModel = "assets/best.torchscript";
  try {
    _objectModel = await FlutterPytorch.loadObjectDetectionModel(
      // cambiar el # 2 con el # de clases del modelo
      pathObjectDetectionModel,
      2,
      640,
      640,
      labelPath: "assets/labels.txt");
  } catch (e) {
    if (e is PlatformException) {
      print("only supported for android, Error is $e");
    } else {
      print("Error is $e");
    }
  }
}
```

**Figura 77.** Integración del modelo en la app

<sup>14</sup> Código de App móvil: <https://n9.cl/manchaangularappcodigo>

### 6.2.6.3 Segunda Iteración

#### - HU02 – Pantalla e información principal

En esta ventana se ubica la vista principal que tiene el usuario al abrir la aplicación, además contiene información básica y algunas recomendaciones para tratar de manera orgánica el cultivo en caso de presentar la enfermedad o querer prevenirla. En la Figura 78 se observa el código correspondiente para llevar a cabo esta tarea.

```
Expanded(
  child: Stack(
    children: [
      Expanded(
        child: SizedBox(
          width: double.infinity,
          height: double.infinity,
          child: Image.asset(
            "assets/images/$nombreImagen.png", //Carga imagen portada
            fit: BoxFit.cover,
            alignment: Alignment.topCenter,
          ),
        ),
      ),
      nombreImagen == 'portadaPrincipal'
        ? Positioned(
            bottom: 10,
            child: Container(
              height: 150,
              width: MediaQuery.of(context).size.width,
              child: Column(
                children: [
                  IconButton(
                    onPressed: () {
                      //Aqui va toda la descripción de la enfermedad
                    },
                    icon: const Icon(
                      Icons.info_outline,
                    ),
                    color: const Color.fromARGB(255, 57, 117, 58),
                    iconSize: 90.0,
                    padding: const EdgeInsets.all(0.0),
                  ),
                  const SizedBox(
                    height: 10,
                  ),
                  const Text(
                    'Información de la Enfermedad',
                    style: TextStyle(
                      fontSize: 18,
                      fontStyle: FontStyle.italic,
                    ),
                  ),
                ],
              ),
            ),
          )
        : Container(),
    ],
  ),
)
```

Figura 78. Módulo de interfaz gráfica pantalla principal

### 6.2.6.4 Tercera Iteración

#### - HU02 / HU03 - Cargar Imágen

Para llevar a cabo una correcta precisión es necesario el ingreso de datos ya sea mediante la galería de fotos o una imagen capturada al instante utilizando la cámara del dispositivo. Para ello se implementó un método que, dependiendo del parámetro “source”, permite dos opciones: seleccionar una imagen de galería o la utilización de la cámara para capturar una foto (método .pickImage), este método se visualiza en la Figura 79.

<b>ImageSource.gallery</b>	<b>ImageSource.camera</b>
a) Galería	b) Cámara

```

Future<void> runObjectDetection(context, source) async {
  //pick an image
  final XFile? image =
    await _picker.pickImage(source: source, maxWidth: 400, maxHeight:
400);
  if (image != null) {
    objDetect = await _objectModel.getImagePrediction(
      await File(image.path).readAsBytes(),
      minimumScore: 0.1,
      IOUThershold: 0.3);
    setState(() {
      _image = File(image.path);
      getImageDimensions(context);
    });
  }
}

```

c) Método

**Figura 79.** Método de importación de imagen para la detección

### 6.2.6.5 Cuarta Iteración

#### - HU04 – Mostrar Resultado

Para llevar a cabo esta tarea, como se muestra en la Figura 80 se utiliza un modelo de detección de objetos, referido como `_objectModel`, para obtener predicciones sobre la imagen mediante la invocación del método `getImagePrediction()`. Una vez realizada la detección se muestra el resultado en pantalla con las detecciones correspondientes encerrando las regiones detectadas en cuadros delimitadores.

```

Container(
  height: imageHeight,
  width: MediaQuery.of(context)
    .size
    .width,
  child: objDetect.isNotEmpty
    ? image == null
      ? const Center(child: Text('Esperando imagen'))
      : objectModel != null
        ? objectModel!.renderBoxesOnImage(
            image!,
            objDetect,
          )
        : const Text('Model not initialized.')
    : image == null
      ? const Center(
          child: Text('Esperando imagen ...'))
      : Image.file(image!),
),

```

**Figura 80.** Código para la detección de la enfermedad


### 6.2.6.6 Descripción de Interfaces de Usuario

Las interfaces se realizaron teniendo en cuenta que ésta, se encuentra orientada a los agricultores, por lo tanto, se busca que sea sencilla y entendible para el usuario.

**Tabla 28.** Pantalla de Inicio

Interfaz	Descripción
	Esta es la primera pantalla que se muestra al ejecutar la aplicación

**Tabla 29.** Información de enfermedad

Interfaz	Descripción
	Esta es pantalla que se muestra al seleccionar la información de la enfermedad



**Tabla 30.** Pantalla con selección de cámara

Interfaz	Descripción
	<p>Esta es la pantalla que se muestra al seleccionar la opción de cámara para el diagnóstico de la enfermedad</p>

**Tabla 31.** Pantalla con selección de Galería

Interfaz	Descripción
	<p>Esta es la pantalla que se muestra al seleccionar la opción de galería para el diagnóstico de la enfermedad con imágenes guardadas en el dispositivo.</p>

**Tabla 32.** Pantalla con resultados de detección

Interfaz	Descripción
	<p>Esta es la pantalla que muestra la imagen con las detecciones realizadas por la aplicación móvil a través el modelo CNN.</p>

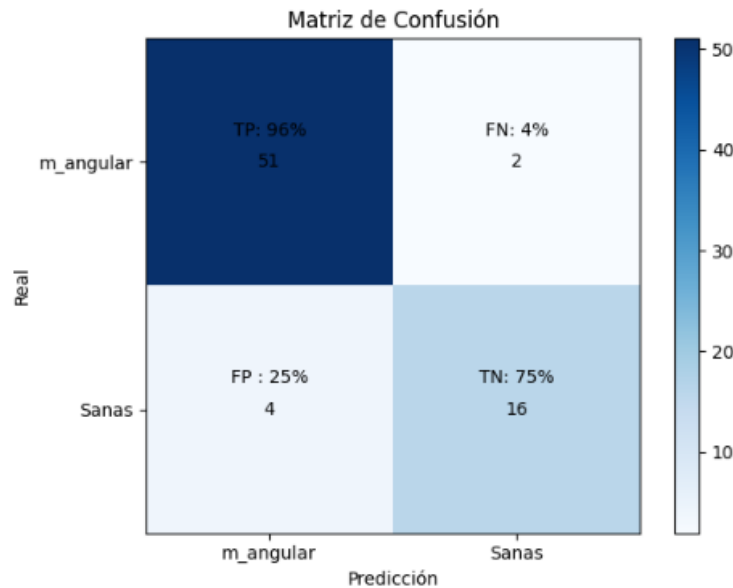
### 6.2.7 Pruebas

Para llevar a cabo la evaluación del modelo integrado al prototipo de Aplicación móvil, se hizo uso de la técnica de Zero-Shot Learning, la cual consiste en usar un nuevo conjunto de datos que el modelo no conoce, para este caso se utilizó 40 imágenes<sup>15</sup> capturadas en un entorno real, divididas en 20 imágenes con hojas enfermas y 20 imágenes con hojas sanas, con la denominación “prueba\_#”. Para ello, fue necesario el traslado al sector Carboncillo del cantón Saraguro, donde se encuentra el cultivo de fresa del Señor Carlos Poma (Ver **Anexo 5**) mismo que esta al aire libre, por lo que hay más incidencia de esta enfermedad al tener más contacto con distintos factores de la naturaleza como la lluvia, viento, humedad, etc, debido a que en este sector no existe conectividad a la red de internet, esta app resulta útil al ser offline.

Durante esta fase, se contó con la participación del Ing. Miguel Condolo, Técnico de Apoyo Agropecuario de la Dirección de Gestión Ambiental y Producción del GADMIS, quien aportó su

<sup>15</sup> Conjunto de pruebas en fase de pruebas del prototipo:  
[https://drive.google.com/drive/folders/146ALjFezxbz2Nn23ojDBTHNfsiPzsOYX?usp=drive\\_link](https://drive.google.com/drive/folders/146ALjFezxbz2Nn23ojDBTHNfsiPzsOYX?usp=drive_link)

experiencia como experto en el campo para identificar correctamente las hojas enfermas. Posteriormente, se procedió a evaluar el modelo integrado en la aplicación móvil, generando como resultado una matriz de confusión que proporcionó una visión clara del rendimiento del mismo en entornos reales. Los resultados se observan en la Figura 81.



**Figura 81.** Matriz de confusión de prueba del modelo en la fase de pruebas

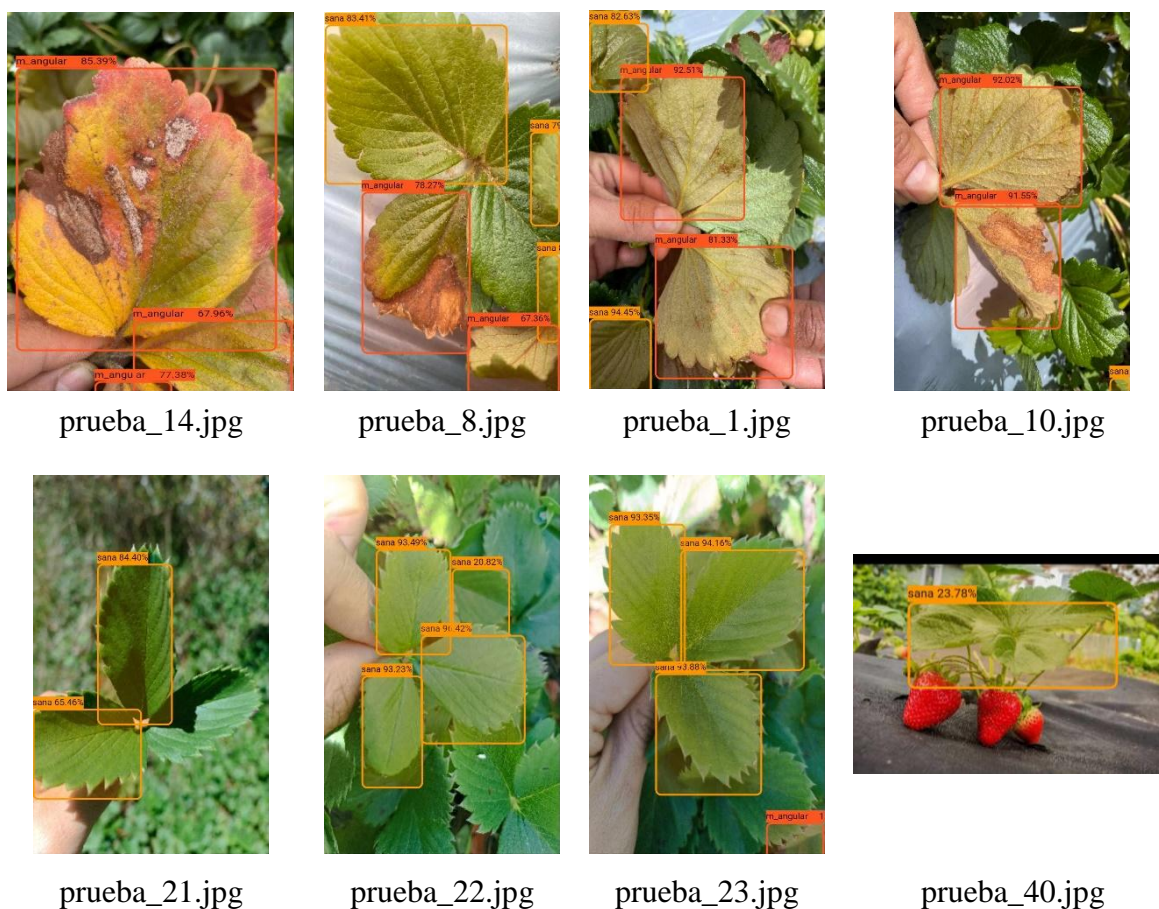
De acuerdo a los resultados obtenidos se tiene que:

- Verdaderos Positivos (VP): existen 51 instancias que fueron correctamente clasificadas como "m\_angular", esto representa un 96% de los valores positivos.
- Falsos Negativos (FN): pertenecen a la clase negativa que fueron incorrectamente clasificadas como "Hojas Sanas" (0 falsos negativos), el modelo clasificó incorrectamente 2 hojas enfermas como sanas que corresponde al 4%.
- Falsos Positivos (FP): perteneces a las instancias que fueron incorrectamente clasificadas como "m\_angular", cuando realmente son hojas sanas, el modelo clasifico incorrectamente 4 instancias que corresponde a un 25% del total de hojas saludables.
- Verdaderos Negativos (VN): corresponde a las imágenes que fueron correctamente clasificadas como "Sanas", clasificando el modelo correctamente 16 imágenes que representa un porcentaje del 75%.

Mediante los resultados obtenidos en la matriz de confusión anteriormente descrita, se calcularon las métricas de evaluación con los siguientes valores resultantes que se visualizan en la Tabla 33.

**Tabla 33.** Resultados de métricas de evaluación final del modelo integrado en el prototipo móvil

Medida	Valor	Derivaciones
<b>Sensibilidad/recall</b>	0.9273	$TPR = VP / (VP + FN)$
<b>Especificidad</b>	0.8889	$SPC = VN / (FP + VN)$
<b>Precisión</b>	0.9623	$VPP = VP / (VP + FP)$
<b>Exactitud</b>	0.9178	$ACC = (VP + VN) / (P + N)$
<b>Puntuación F1</b>	0.9444	$F1 = 2VP / (2VP + FP + FN)$



**Figura 82.** Detecciones realizadas por la aplicación móvil

## 7. Discusión

Al término del proceso de implementación del modelo basado en la arquitectura de redes neuronales convolucionales(CNN) para identificar la enfermedad de mancha angular en las hojas de fresa, se procede a analizar los resultados obtenidos, abordando cada objetivo y destacando los resultados más significativos en cada uno de ellos, resaltando las contribuciones más importantes que han surgido durante el desarrollo de este Trabajo de Titulación.

### **Objetivo 1. Implementar un modelo de red neuronal convolucional y el conjunto de datos necesarios para la detección de la Mancha Angular en la fresa.**

La metodología CRISP-ML ha demostrado ser un marco metodológico adaptable, capaz de ajustar sus etapas de manera eficaz para facilitar la realización del proyecto en cuestión.

La creación de un nuevo conjunto de datos con 500 imágenes, ha ampliado la diversidad de imágenes disponibles de hojas de fresa con mancha angular. Durante la revisión de trabajos relacionados, se observó que solo existía evidencia de un conjunto de datos con imágenes de calidad que abordaba la enfermedad específica requerida para este proyecto [27] [28]. Por lo tanto, las imágenes generadas en este trabajo se pueden considerar una contribución significativa dentro de este ámbito, al agregar variedad y enriquecer el repositorio de datos disponible para futuras investigaciones, ya que este conjunto se distingue al utilizar fondos provenientes de los propios sembríos, permitiendo al modelo aprender a identificar la enfermedad en entornos auténticos.

De acuerdo al análisis de los trabajos relacionados, se observa que en el proceso de tratamiento de datos se emplean distintas técnicas de aumento de datos utilizando algoritmos implementados en Python o Matlab. Para el etiquetado de imágenes se utilizaron herramientas como LabelImg, LabelMe, entre otras. En este trabajo, se optó por la herramienta Roboflow durante la fase de preparación de datos, ya que, a diferencia de los recursos utilizados en otros estudios, esta permite etiquetar, estandarizar y aplicar distintas técnicas de aumento de datos de manera automática. También contribuye en la generación de un conjunto de datos balanceado, distribuyéndolo automáticamente en entrenamiento y validación. Al ser una herramienta en la nube permite gestionar las imágenes directamente desde Colab, agilizando el flujo de trabajo.

De la misma manera con el análisis de los trabajos relacionados en la RSL, se pudo evidenciar que en su gran mayoría se han utilizado modelos de clasificación, por lo que para este TT se filtraron solo aquellos trabajos destinados a identificar la parte enferma de un cultivo. No se encontró

trabajos relacionados a la identificación de mancha angular en la hoja de fresa, por lo que se analizaron investigaciones con objetivos similares como es el caso de [40], [41], [42], [43], [44], [45], [46], [47], [48], [49], [50], [51], y [52]. En base a esto, se obtuvo tres modelos donde el umbral de aceptación de la métrica más significativa en el campo de la detección de objetos es la precisión. Esto condujo a la selección de tres modelos con el mejor rendimiento, siendo Faster RCNN, Yolov5 y Yolov7, como parte de este análisis y experimentación, se evalúa el modelo Yolov8 ya que, al ser una versión reciente, no se encontraron trabajos relacionados que lo hayan implementado.

Se llevó a cabo el entrenamiento de estos modelos en Google Colab con 200 épocas para cada uno, utilizando un mismo conjunto de datos y parámetros. Posteriormente se realizó la comparación de las métricas de matriz de confusión con el fin de determinar el modelo con mayor rendimiento. Durante este experimento, el modelo Faster RCNN, obtuvo la precisión más baja (82%) a comparación de los modelos Yolov7 y Yolov8 que alcanzaron un 89% y 92% respectivamente, por su parte el modelo Yolov5, superó a todos, alcanzando una tasa de precisión del 93%, equiparándose con los resultados conseguidos en otro estudio con enfoques similares como [49] y [50], que aplicaron técnicas similares, incluyendo el modelo Yolov5, obteniendo una precisión de 85.35% y 94% respectivamente. Aunque el modelo Yolov5 demostró un rendimiento superior frente a los demás modelos, se buscó mejorar su desempeño añadiendo una categoría de imágenes sanas al conjunto de datos, esto, con la finalidad de ampliar su capacidad de generalización, ya que, el aprender nuevas características, le permite diferenciar con mayor precisión las hojas enfermas de las sanas reduciendo los falsos positivos, como resultado se obtuvo una mejora en la precisión, sensibilidad, puntaje F1 y mAP, alcanzando un 94%, 97%, 92% y 94% respectivamente. Considerando estos resultados se comprobó que el modelo Yolov5, identifica de manera eficiente las hojas de fresa que presentan características de mancha angular, y sobre todo superó el índice de precisión del 85%, obtenido en estudios anteriores.

## **Objetivo 2: Construir un prototipo de software e integrar el modelo de red neuronal utilizando la metodología XP.**

La construcción de un prototipo de aplicación móvil integrando el modelo entrenado, se destaca de los otros trabajos relacionados ya que estos no contemplan esta herramienta en su implementación, además la metodología XP empleada en este proceso de desarrollo del software

permitió llevar cada fase de la construcción del software de manera ordenada adaptando algunas de sus fases al tiempo y tamaño del proyecto llevado a cabo.

La integración del modelo en una aplicación móvil implica varios desafíos técnicos que incluyen el procesamiento de imágenes y la ejecución de inferencias en dispositivos con recursos limitados. Para abordar esto, se convirtió el modelo a formato Torchscript, lo que permitió una implementación liviana y compatible con el entorno de desarrollo basado en Flutter. Dado que la aplicación está destinada a ser utilizada por agricultores en el campo, donde la conectividad a Internet puede ser limitada o inexistente, se diseñó para funcionar completamente offline, lo que permite al usuario identificar la enfermedad en tiempo real sin depender de una conexión a internet.

Las pruebas finales se realizaron en un cultivo ubicado en la comunidad de Tuchin, con la finalidad de capturar imágenes no vistas por el modelo durante su etapa de entrenamiento, para validar las detecciones realizadas por la herramienta móvil, se contó con la colaboración del agrónomo. Para esta prueba se capturaron como muestra 40 imágenes, distribuidas en 20 fotografías con 53 instancias de hojas enfermas y 20 de hojas sanas.

El modelo integrado en la app móvil, identificó correctamente 51 hojas enfermas de 53 y señaló incorrectamente como enfermas a 4 de 20 hojas sanas. En métricas de matriz de confusión esto corresponde a una sensibilidad de 92.73%, especificidad de 88.89%, precisión 96.23%, exactitud de 91.78% y puntuación F1 de 94.44%. En base a estos resultados, se da respuesta a la pregunta de investigación planteada en este Trabajo de Titulación: **¿En qué medida podrá el modelo predictivo basado en deep learning identificar la enfermedad de la Mancha Angular en las hojas de fresa del Cantón Saraguro?** indicando que el modelo entrenado no solo es capaz de detectar la presencia de la enfermedad de la Mancha Angular con buena precisión, sino que también ofrece una evaluación equilibrada entre la sensibilidad y la especificidad. Por lo tanto, el modelo predictivo basado en deep learning es efectivo para la identificación de la Mancha Angular en las hojas de fresa del Cantón Saraguro, pudiendo representar una herramienta confiable para el monitoreo y la gestión de esta enfermedad.

## 8. Conclusiones

Como resultado del desarrollo del modelo deep learning para la detección de mancha angular en la hoja de fresa, se puede concluir que:

- La aplicación de la metodología CRISP- ML resultó ser una metodología muy útil que permitió llevar a cabo el proceso de desarrollo del modelo de manera eficiente desde la preparación de datos hasta la implementación e integración del modelo en el prototipo móvil para ser utilizado en entornos reales, dando como resultado una herramienta eficiente para detectar hojas con mancha angular en el cultivo de fresa.
- La construcción de un set de datos propio, con imágenes capturadas en entornos reales y cultivos locales, ha permitido obtener resultados positivos en cuanto a la detección de la mancha angular en la hoja de fresa en los cultivos del cantón Saraguro, siendo un gran aporte en este campo de investigación.
- El desarrollo de una RSL hizo posible identificar a los modelos Faster R-CNN, Yolov5 y Yolov7, como las arquitecturas con mayor precisión al momento de identificar enfermedades foliares en estudios recientes.
- La implementación de los modelos seleccionados en la RSL permitió comparar sus resultados al entrenarlos bajo las mismas condiciones y con los mismos valores de hiperparámetros, obteniendo así que, en la fase de entrenamiento el modelo Yolov5 alcanzó un mayor rendimiento sobre Faster RCN, Yolov7 y Yolov8, logrando un grado de precisión del 93%.
- El entrenamiento del modelo Yolov5 incorporando una clase adicional de hojas “sanas” al conjunto de datos, contribuyó a una leve pero significativa mejora alcanzando un 94% en la medida de precisión.
- La construcción de una aplicación móvil offline proporciona una herramienta portable y útil para los agricultores, especialmente para aquellos en áreas rurales sin acceso a internet.
- La validación del modelo integrado en la aplicación móvil, durante la fase final de pruebas, realizada mediante prácticas de campo en un cultivo real y comparando los resultados con los de un experto, mostró que logró una precisión del 96%, una sensibilidad del 92%, una exactitud del 91% y una puntuación F1 de 94%. Estos resultados confirman que el modelo es efectivo al detectar la enfermedad de mancha angular en condiciones reales de campo.



## 9. Recomendaciones

De acuerdo a las diferentes experiencias obtenidas durante el desarrollo del presente Proyecto de Titulación se presentan las siguientes recomendaciones:

- Ampliar el número de enfermedades de la fresa en el conjunto de datos con el fin de entrenar un modelo que reconozca una mayor variedad, brindando una ayuda más completa al agricultor.
- Incluir un sistema de actualización que permita la integración de nuevos datos y reentrenamiento periódica del modelo, lo que aumentaría su precisión y capacidad de adaptación a cambios. Además, desarrollar un software que consuma el modelo desde la nube lo que permitiría tener acceso a un modelo entrenado más actualizado, permitiendo a la aplicación adaptarse a entornos dinámicos y cambiantes.
- Implementar funcionalidades de visión por computadora para realizar la detección de mancha angular en la hoja de fresa con video en tiempo real.
- Capacitar a los agricultores de la Escuela Agroecológica del GAD municipal de Saraguro, en el uso de nuevas tecnologías, como la herramienta desarrollada en este estudio, que faciliten la automatización en la detección de enfermedades en sus cultivos.

## 10. Bibliografía

- [1] J. D. Janse *et al.*, “Bacterial leaf blight of strawberry (*Fragaria (x) ananassa*) caused by a pathovar of *Xanthomonas arboricola*, not similar to *Xanthomonas fragariae* Kennedy & King. Description of the causal organism as *Xanthomonas arboricola* pv. *fragariae* (pv. nov., comb. nov.),” vol. 50, no. 6, pp. 653–665, doi: 10.1046/j.1365-3059.2001.00644. x.
- [2] D.-R. Kim, G. Gang, C.-W. Jeon, N. J. Kang, S. Lee, and Y.-S. Kwak, “Epidemiology and Control of Strawberry Bacterial Angular Leaf Spot Disease Caused by *Xanthomonas fragariae*,” *Plant Pathol J*, vol. 32, no. 4, pp. 290–299, Aug. 2016, doi: 10.5423/PPJ.OA.01.2016.0007.
- [3] L. Á. Munárriz, *Fundamentos de inteligencia artificial*, vol. 1. Editum, 1994.
- [4] E. Alpaydin, *Introduction to Machine Learning*, 2nd ed. Cambridge, MA: MIT Press, 2010.
- [5] “Clasificación con datos desbalanceados Aprende Machine Learning.” [Online]. Available: <https://www.aprendemachinelearning.com/clasificacion-con-datos-desbalanceados/>
- [6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.
- [7] A. Kaehler and G. Bradski, *Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library*. “O’Reilly Media, Inc.,” 2016.
- [8] “Hagámoslo fácil: Deep Learning y Redes convolucionales.” Accessed: May 10, 2024. [Online]. Available: <https://blog.spainbs.com/2020/10/891/hagamoslo-facil-deep-learning-y-redes-convolucionales>
- [9] “Introducción a las redes neuronales convolucionales: Guía completa de las CNN en el aprendizaje profundo | DataCamp.” Accessed: May 10, 2024. [Online]. Available: <https://www.datacamp.com/es/tutorial/introduction-to-convolutional-neural-networks-cnns>
- [10] P. P. Torralba, “Qué son las Redes Neuronales Convolucionales,” *Thinking for Innovation*, Sep. 2022, Accessed: May 10, 2024. [Online]. Available: <https://www.iebschool.com/blog/redes-neuronales-convolucionales-big-data/>
- [11] Í. C. Bengi, “YOLOv5: Revolucionando la detección de objetos en tiempo real,” *HashDork*. Jan. 2023. [Online]. Available: <https://hashdork.com/es/yolov5/>
- [12] “Evolution of yolo algorithm and yolov5: the state-of-the-art object detection algorithm”.

- [13] “Explicación de la arquitectura YOLOv7.” [Online]. Available: <https://www.plugger.ai/blog/yolov7-architecture-explanation>.
- [14] “YOLOv7: El algoritmo de detección de objetos más rápido (2024) - viso.ai.” Accessed: May 10, 2024. [Online]. Available: <https://viso.ai/deep-learning/yolov7-guide/>
- [15] “YOLOv7 A breakdown of how it works - Online Store.” [Online]. Available: [https://goodtopssm.life/product\\_details/100183118.html](https://goodtopssm.life/product_details/100183118.html)
- [16] “Understanding YOLOv8 Architecture, Applications & Features,” Labellerr. Apr. 2023. [Online]. Available: <https://www.labellerr.com/blog/understanding-yolov8-architecture-applications-features/>
- [17] “Resumen de arquitectura - Ultralytics YOLOv8 Docs.” Accessed: May 10, 2024. [Online]. Available: [https://docs.ultralytics.com/es/yolov5/tutorials/architecture\\_description/](https://docs.ultralytics.com/es/yolov5/tutorials/architecture_description/)
- [18] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2001, doi: 10.1109/CVPR.2001.990517.
- [19] “R-CNN más rápido: detección de objetos. Desmitificando la detección de objetos | de Vikash Kumar Thakur | El centro profundo | Medio.” Accessed: May 08, 2024. [Online]. Available: <https://medium.com/thedeephub/faster-r-cnn-object-detection-5dfe77104e31>
- [20] “Metodología CRISP-DM - Adictos al trabajo Tutoriales.” Accessed: May 07, 2024. [Online]. Available: <https://www.adictosaltrabajo.com/2021/01/14/metodologia-crisp-dm/#2>
- [21] S. F. Tamayo Proaño, “Desarrollo de un modelo de machine learning para predecir el rendimiento académico de los estudiantes universitarios en términos de aprobación o reprobación de materias utilizando la metodología CRISP-DM.” PUCE - Quito, Jul. 17, 2023. Accessed: May 07, 2024. [Online]. Available: <https://repositorio.puce.edu.ec/handle/123456789/40513>
- [22] “Google Colab, una nueva herramienta para IA y Data Analysis.” Accessed: May 07, 2024. [Online]. Available: <https://www.tokioschool.com/noticias/google-colab/>
- [23] “Aprendizaje Zero-Shot: Cuando las Máquinas Aprenden lo Nunca Visto | de Patricio Bustamante M. | Medio.” Accessed: May 08, 2024. [Online]. Available: <https://medium.com/@aulasimple/zero-shot-learning-cuando-las-m%C3%A1quinas-aprenden-lo-nunca-visto-4a5df0321ed8>

- [24] “La matriz de confusión y sus métricas – Inteligencia Artificial –.” [Online]. Available: <https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/>
- [25] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” vol. 45, no. 4, pp. 427–437, doi: <https://doi.org/10.1016/j.ipm.2009.03.002>.
- [26] D. Powers, “Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation,” vol. 2.
- [27] U. Afzaal, B. Bhattarai, Y. R. Pandeya, and J. Lee, “An Instance Segmentation Model for Strawberry Diseases Based on Mask R-CNN,” *Sensors 2021, Vol. 21, Page 6565*, vol. 21, no. 19, p. 6565, Sep. 2021, doi: 10.3390/S21196565.
- [28] “Conjunto de datos de detección de enfermedades de la fresa | Kaggle.” Accessed: May 07, 2024. [Online]. Available: <https://www.kaggle.com/datasets/usmanafzaal/strawberry-disease-detection-dataset>
- [29] “LabelImg para anotación de imágenes - viso.ai.” Accessed: May 07, 2024. [Online]. Available: <https://viso.ai/computer-vision/labelimg-for-image-annotation/>
- [30] T. Telecomunicaci´ and T. De Barcelona, “Detección de objetos y anotación de imágenes en el iPhone,” Feb. 2013, Accessed: May 07, 2024. [Online]. Available: <https://upcommons.upc.edu/handle/2099.1/18241>
- [31] P. Kakani and S. Vyas, “AUTOMATED CATALOG GENERATION USING DEEP LEARNING”, doi: 10.56726/IRJMETS44010.
- [32] U. Carlos, I. De Madrid, V. Galán, C. Tutora, and E. Castro Galán, “PROYECTO FIN DE CARRERA,” 2015.
- [33] “XP - Extreme Programing Ingenieria de Software.” [Online]. Available: [https://ingenieriadesoftware.mex.tl/52753\\_XP—Extreme-Programing.html](https://ingenieriadesoftware.mex.tl/52753_XP—Extreme-Programing.html)
- [34] “Metodología Ágil Programación Extrema XP”.
- [35] P. Letelier and M. C. Penadés, “Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP),” Apr. 2006. [Online]. Available: <http://www.cyta.com.ar/ta0502/v5n2a1.htm>
- [36] P. De Titulación, I. Joffre, and V. L. Acurio, “Estudio comparativo de los frameworks del desarrollo móvil nativo ‘Flutter’ y ‘React Native,’” 2021, Accessed: May 06, 2024. [Online]. Available: <http://dspace.utb.edu.ec/handle/49000/10516>

- [37] V. Asensi-Artiga and A. Parra-Pujante, “El método científico y la nueva filosofía de la ciencia,” *Anales de Documentación*, vol. 5, pp. 9–19, 2002, [Online]. Available: <https://revistas.um.es/analesdoc/article/view/2251>
- [38] J. Lopera, C. Ramirez, M. Zuluaga, and J. Ortiz, *El método analítico*. Centro de Investigaciones Sociales y Humanas Universidad de Antioquia, 2010. [Online]. Available: <http://repository.eafit.edu.co/handle/10784/26204>
- [39] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, “Systematic literature reviews in software engineering—a systematic literature review,” *Inf Softw Technol*, vol. 51, no. 1, pp. 7–15, 2009.
- [40] S. Mousavi and G. Farahani, “A Novel Enhanced VGG16 Model to Tackle Grapevine Leaves Diseases with Automatic Method,” *IEEE Access*, vol. 10, pp. 111564–111578, Mar. 2022, doi: 10.1109/ACCESS.2022.3215639.
- [41] J. and L. R. and L. Q. and Y. J. and M. Y. and L. Y. and W. Y. and Z. R. and M. H. and L. Y. and Z. L. and Z. L. and Z. X. and K. J. and Y. W. and M. L. Tan Zhihao and Shi, “Fast anther dehiscence status recognition system established by deep learning to screen heat tolerant cotton,” *Plant Methods*, vol. 18, no. 1, p. 53, Apr. 2022, doi: 10.1186/s13007-022-00884-0.
- [42] T. and J. A. and M. M. and R. J. and K. J. and H. A. Nawaz Marriam and Nazir, “A robust deep learning approach for tomato plant leaf disease localization and classification,” *Sci Rep*, vol. 12, no. 1, p. 18568, Nov. 2022, doi: 10.1038/s41598-022-21498-5.
- [43] I. D. Carreño Chero and A. J. Obregon Mauricio, “Modelo YOLO (You Only Look Once) para el diagnóstico preliminar de la enfermedad de Huanglongbing en la planta de limón en la región Piura, 2023,” *Repositorio Institucional - UCV*, 2023, Accessed: May 03, 2024. [Online]. Available: <https://repositorio.ucv.edu.pe/handle/20.500.12692/135325>
- [44] Y. Li, J. Wang, H. Wu, Y. Yu, H. Sun, and H. Zhang, “Detection of powdery mildew on strawberry leaves based on DAC-YOLOv4 model,” vol. 202, p. 107418, doi: 10.1016/j.compag.2022.107418.
- [45] D. Kerre and H. Muchiri, “Detecting the Simultaneous Occurrence of Strawberry Fungal Leaf Diseases with a Deep Normalized CNN,” in *2022 7th International Conference on Machine Learning Technologies (ICMLT)*, in ICMLT 2022. New York, NY, USA: Association for Computing Machinery, 2022, pp. 147–154. doi: 10.1145/3529399.3529424.
- [46] F. Jakjoud, A. Hatim, and A. Bouaaddi, “Deep Learning Application for Plant Diseases Detection,” in *Proceedings of the 4th International Conference on Big Data and Internet of Things*, in BDIoT’19. New York, NY, USA: Association for Computing Machinery, 2020. doi: 10.1145/3372938.3372983.

- [47] A. A. Alfarisy, Q. Chen, and M. Guo, "Deep Learning Based Classification for Paddy Pests & Diseases Recognition," in *Proceedings of 2018 International Conference on Mathematics and Artificial Intelligence*, in ICMAI '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 21–25. doi: 10.1145/3208788.3208795.
- [48] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, and N. Batra, "PlantDoc: A Dataset for Visual Plant Disease Detection," in *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, in CoDS COMAD 2020. New York, NY, USA: Association for Computing Machinery, 2020, pp. 249–253. doi: 10.1145/3371158.3371196.
- [49] Z. Zhang, Y. Qiao, Y. Guo, and D. He, "Deep Learning Based Automatic Grape Downy Mildew Detection," *Front Plant Sci*, vol. 13, p. 872107, Jun. 2022, doi: 10.3389/FPLS.2022.872107/BIBTEX.
- [50] N. Amarasingam, F. Gonzalez, A. S. A. Salgadoe, J. Sandino, and K. Powell, "Detection of White Leaf Disease in Sugarcane Crops Using UAV-Derived RGB Imagery with Existing Deep Learning Models," *Remote Sensing 2022, Vol. 14, Page 6137*, vol. 14, no. 23, p. 6137, Dec. 2022, doi: 10.3390/RS14236137.
- [51] J. Liu and X. Wang, "Tomato Diseases and Pests Detection Based on Improved Yolo V3 Convolutional Neural Network," *Front Plant Sci*, vol. 11, p. 521544, Jun. 2020, doi: 10.3389/FPLS.2020.00898/BIBTEX.
- [52] Md. J. A. Soeb *et al.*, "Tea leaf disease detection and identification based on YOLOv7 (YOLO-T)," *Sci Rep*, vol. 13, no. 1, p. 6078, 2023, doi: 10.1038/s41598-023-33270-4.

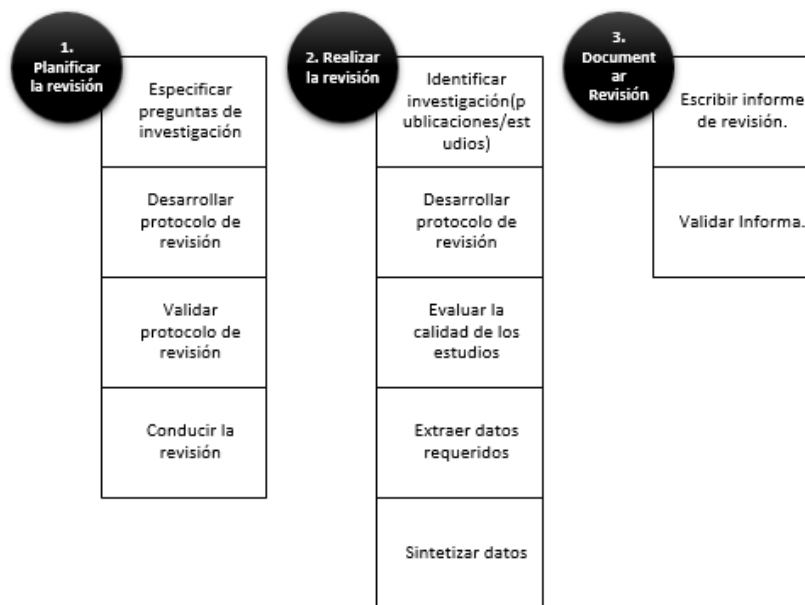
## 11. Anexos

### Anexo 1. Revisión Sistemática de Literatura

La presente Revisión Sistemática de Literatura (RSL) se encuentra dividida en las siguientes secciones: Sección 1 **Materiales y métodos**, en donde se describe la metodología propuesta por Bárbara Kitchenham para la redacción de Revisiones de Literatura, la cual se aplicó en este proyecto, posteriormente, en la sección 2 **Resultados**, se describe el resultado de la aplicación de cada una de las fases de la metodología, resaltando sobre todo los trabajos relacionados, mismos que fueron analizados minuciosamente para responder a las preguntas de investigación planteadas en la sección 1. Así mismo en la sección 3 **Discusión**, se analizó los resultados obtenidos durante todo el proceso de revisión y finalmente, se plantean las **Conclusiones** resultantes del desarrollo de la RSL.

#### 1. Materiales y métodos

La RSL, se basa en el proceso establecido en la metodología de Bárbara Kitchenham[1], dicho proceso consta de tres fases fundamentales:



**Figura 83.** Proceso de la metodología de Bárbara Kitchenham

En vista de los puntos expuestos en la Figura 83, en la presente RSL, no se ejecutaron todos los pasos propuestos por Kitchenham, por consiguiente, se aprovechó la flexibilidad que brinda al investigador esta metodología, permitiendo adaptarla de acuerdo al alcance y necesidad requerida

para cada investigación. De acuerdo con esto, en la Tabla 34, se presentan las fases y tareas que fueron consideradas:

**Tabla 34.** Fases que comprenden el proceso de RSL basado en la metodología de Bárbara Kitchenham

Fases	Tareas
<i>Planificar la revisión</i>	<ul style="list-style-type: none"> <li>Identificar la necesidad de realizar una revisión.</li> <li>Definir las preguntas de investigación.</li> <li>Crear el protocolo de revisión.</li> </ul>
<i>Realizar revisión</i>	<ul style="list-style-type: none"> <li>Identificar fuentes y estudios pertinentes.</li> <li>Elegir los estudios primarios.</li> <li>Sintetizar los datos obtenidos</li> </ul>
<i>Documentar la revisión</i>	<ul style="list-style-type: none"> <li>Redactar informe de revisión.</li> </ul>

## 2. Resultados

### 2.1 Planificación de la revisión

#### 2.1.1 Identificación de la necesidad de realizar una revisión.

La presente RSL tiene como finalidad identificar modelos basados en redes neuronales convolucionales más utilizados en los últimos años para el diagnóstico de enfermedades foliares, especialmente en la fresa, para definir un modelo o arquitectura capaz de determinar la enfermedad de la mancha angular en la hoja de la planta de fresa, con un nivel de eficiencia aceptable rápida y precisa, así mismo la revisión de las métricas utilizadas para la evaluación del desempeño de estos modelos de detección.

#### 2.1.2 Definición las preguntas de investigación

En esta fase se establecieron tres preguntas con el objetivo de evaluar la calidad de los documentos encontrados, de forma que, se pueda seleccionar a aquellos que permitan responder a las mismas, manteniendo siempre el enfoque durante este proceso investigativo. En la Tabla 35 se visualizan las interrogantes planteadas en esta RSL:

**Tabla 35.** Preguntas de Investigación

ID	Preguntas de Investigación
PI_1	¿Cuáles son los modelos o arquitecturas basados en CNN más precisos, empleados hasta la fecha, para identificar enfermedades en las hojas?
PI_2	¿Cuáles son las métricas utilizadas para evaluar el rendimiento de los modelos o arquitecturas para la detección de objetos?



### 2.1.3 Creación del protocolo de revisión

#### 2.1.3.1 Estrategia de búsqueda

Richardson et al. propusieron el método PICOC en 1995, se utiliza para definir el protocolo de la RSL. Este ámbito debe contribuir en el proceso de análisis de los artículos seleccionados para dar respuesta a las preguntas de investigación. El *PICOC* ayuda a estructurar los cinco elementos de una pregunta clínica.

- **P = Población:** ¿Quién?
- **I = Intervención:** ¿Qué o cómo?
- **C = Comparación:** ¿Comparado con qué?
- **O = Resultado:** ¿Qué estás tratando de lograr / mejorar?
- **C = Contexto:** ¿En qué tipo de organización / circunstancias?

No obstante, de la nemotécnica PICOC, para la presente RSL no se considera el criterio de *Comparación*, debido a que el criterio de intervención es lo que se desea identificar más no comparar, por lo tanto, se emplearon sólo cuatro componentes: Population (P), Intervention (I), Outcomes (O) y Context (C), como se indica en la Tabla 36.

**Tabla 36.** Palabras claves en el método PICOC

<i>Población (P):</i>	leaf diseases
<i>Intervención (I):</i>	convolutional neural networks, deep learning, CNN, computer vision, machine learning, Detection, Identification
<i>Resultados (O):</i>	Model, Architecture, metrics, images
<i>Contexto (C):</i>	artificial intelligence,neural networks

#### 2.1.3.2 Fuentes bibliográficas

En cuanto a las fuentes bibliográficas, en la Tabla 37 se enlistan las bibliotecas virtuales que se ha seleccionado para llevar a cabo la presente revisión:

**Tabla 37.** Fuentes de búsqueda

<b>Fuente de Búsqueda</b>	<b>Dirección Web</b>
<i>IEEE</i>	<a href="https://ieeexplore.ieee.org/Xplore/home.jsp">https://ieeexplore.ieee.org/Xplore/home.jsp</a>
<i>ACM Digital Library</i>	<a href="https://dl.acm.org/">https://dl.acm.org/</a>
<i>Springer Link</i>	<a href="https://link.springer.com/">https://link.springer.com/</a>
<i>Scopus</i>	<a href="http://www.scopus.com">http://www.scopus.com</a>

### 2.1.3.3 Definir palabras clave para el problema de investigación

Se seleccionaron a partir de la relación con cada una de las categorías del **PICOC**. Estas son:

*Leaf diseases, convolutional neural networks, deep learning, CNN, computer vision, machine learning, Detection, Identification, Model, Architecture, metrics, datasets, artificial intelligence, neural networks.*

### 2.1.3.4 Cadenas de búsqueda

Para determinar la cadena de búsqueda principal se definió con base a las palabras claves obtenidas en el PICOC formando expresiones y siendo concatenados a través de los operadores lógicos “AND/OR”. En la Tabla 38 se indican las cadenas establecida para cada biblioteca virtual seleccionada.

**Tabla 38.** Cadenas de búsqueda por cada base de datos científica

<b>Base de Datos Científica</b>	<b>Cadena de Búsqueda</b>
IEEE	("leaf diseases") AND ("CNN" OR "computer vision" OR "convolutional neural networks" OR "deep learning" OR "Detection" OR "Identification" OR "machine learning") AND ("architectures" OR "metrics" OR "models") filtros aplicados: date: 2018-2023 document: conferences
ACM	("leaf diseases") AND ("CNN" OR "computer vision" OR "convolutional neural networks" OR "deep learning" OR "Detection" OR "Identification" OR "machine learning") AND ("architectures" OR "metrics" OR "models") date: 2018-2023
SPRINGER LINK	("leaf diseases") AND ("CNN" OR "computer vision" OR "convolutional neural networks" OR "deep learning" OR "Detection"

	OR "Identification" OR "machine learning") AND ("architectures" OR "metrics" OR "models") filtros aplicados: date: 2018-2023 document: artículos
SCOPUS	TITLE-ABS-KEY ( ( ( "leaf diseases" ) AND ( "Convolutional Neural Network" ) ) OR ( "Deep Learning" OR "Detection" OR "Recognition" ) ) ) AND ( LIMIT-TO ( PUBYEAR , 2023 ) OR LIMIT-TO ( PUBYEAR , 2022 ) OR LIMIT-TO ( PUBYEAR , 2021 ) OR LIMIT-TO ( PUBYEAR , 2020 ) OR LIMIT-TO ( PUBYEAR , 2019 ) )

### 2.1.3.5 Criterios de inclusión y exclusión

Teniendo en cuenta los resultados expuestos al aplicar las cadenas de búsqueda y con la finalidad de poder filtrar información, que sea relevante para el tema de estudio, se determinó criterios para la selección de estudios, basándose en las preguntas de investigación y según la metodología Barbara Kitchenham. Para incluir o excluir los estudios fue necesario establecer criterios de inclusión (CI) y criterios de exclusión (CE), los cuales se describen en la Tabla 39 y Tabla 40.

**Tabla 39.** Criterios de Inclusión

ID	Criterios de Inclusión
CI_1	Artículos científicos publicados en revistas o congresos
CI_2	Estudios publicados a partir del 2018
CI_3	Estudios que en el título o resumen contenga las palabras claves.
CI_4	Estudios cuyo título tenga relación con el tema de investigación
CI_5	Estudios en español e inglés

**Tabla 40.** Criterios de exclusión

ID	Criterios de Exclusión
CE_1	Trabajos mal estructurados y poco claros.
CE_2	Publicaciones informales que no siguen una metodología científica
CE_3	Estudios que no contribuyan a las preguntas de investigación
CE_4	Estudios duplicados
CE_5	Todos los que no cumplen con los criterios de inclusión.

## 2.2 Realizar la revisión

### 2.2.1 Identificación de fuentes y estudios relevantes

La presente Revisión de Literatura se lleva a cabo con el afán dar respuesta a las preguntas de investigación planteadas, a través de la búsqueda de estudios primarios que contengan información relacionada con la implementación de modelos basados en redes neuronales convolucionales para el diagnóstico de enfermedades en la hoja de la fresa o similares. Por ello, la selección de los estudios primarios se llevó a cabo comparando los resultados de diversas fuentes, asegurando así que la información sea de alta calidad y se centre exclusivamente en el objetivo de esta RSL.

### 2.2.2 Seleccionar estudios primarios.

Los estudios obtenidos al ejecutar las cadenas de búsqueda en cada base de datos científica se presentan en la Tabla 41, los mismos, son sometidos a tres análisis que permitirán seleccionar aquellos artículos que cumplan con los criterios de inclusión y exclusión.

**Tabla 41.** Resultados de Búsqueda

<b>Base de Datos Científica</b>	<b>Número de estudios encontrados</b>
IEEE Digital Library	13
ACM	33
Springer Link	50
Scopus	82
TOTAL	178

#### 2.2.2.1 Análisis de título y resumen

Durante esta etapa se analiza únicamente el título y resumen de cada uno de los documentos seleccionados, al culminar la lectura y análisis de la información se decidió si el artículo es aceptado o rechazado.

En la Tabla 42, se presenta el número de documentos aceptados durante esta primera revisión, en el caso de los estudios repetidos en dos o más base de datos, se aceptará el que tenga mayor presencia.

**Tabla 42.** Número de artículos aceptados, descartados y duplicados resultantes del análisis uno

<b>Base de Datos Científica</b>	<b>Documentos Aceptados</b>	<b>Documentos Rechazados</b>	<b>Documentos Duplicados</b>
IEEE Digital Library	8	4	1
ACM	19	14	0
Springer Link	11	39	0
Scopus	34	46	2
TOTAL	72	103	3

### 2.2.2.2 Análisis de Introducción, Métodos y Conclusiones

Tras finalizar el primer análisis, se lee e inspecciona la introducción, métodos y conclusiones de cada estudio o artículo científico aceptado en la fase anterior, se da un peso de 1 a los documentos que se incluirán y con 0 a los que no contienen información relevante por lo que se excluyen.

En la Tabla 43 se muestra el número de documentos incluidos en el segundo análisis, los estudios que no fueron accesibles por ningún medio para su inspección y lectura de introducción, métodos y conclusiones fueron descartados.

**Tabla 43.** Número de artículos aceptados, descartados y duplicados resultantes del análisis dos

<b>Base de Datos Científica</b>	<b>Documentos Aceptados</b>	<b>Documentos Rechazados</b>
IEEE Digital Library	7	1
ACM	10	9
Springer Link	8	3
Scopus	19	15
TOTAL	44	28

### 2.2.2.3 Análisis mediante lectura crítica de artículos

Concluido el análisis anterior se hace una lectura crítica de los documentos aceptados. Al culminar esta fase analítica se procede a marcar el estudio como aceptado o descartado, teniendo como resultado lo que se indica en la Tabla 44.

**Tabla 44.** Número de artículos aceptados, descartados y duplicados resultantes del análisis tres

<b>Base de Datos Científica</b>	<b>Documentos Aceptados</b>	<b>Documentos Rechazados</b>
IEEE Digital Library	6	1
ACM	8	2
Springer Link	5	3
Scopus	9	10
<b>TOTAL</b>	<b>28</b>	<b>16</b>

Finalizado este último análisis, se obtuvo un total de 28 trabajos aceptados (TA), los cuales cumplen con los criterios de aceptación y sobre todo que contienen información que permiten contestar las preguntas de investigación de la presente RSL, dichos estudios se presentan en la Tabla 45.

**Tabla 45.** Trabajos aprobados (TA) obtenidos durante la fase de selección de estudios

<b>Trabajo Analizado</b>	<b>Título</b>	<b>Referencia</b>
<b>TA01</b>	Identification of Maize Leaf Diseases Using Improved Deep Convolutional Neural Networks	[2]
<b>TA02</b>	Automatic Recognition of Soybean Leaf Diseases Using UAV Images and Deep Convolutional Neural Networks	[3]
<b>TA03</b>	Beans Leaf Diseases Classification Using MobileNet Models	[4]
<b>TA04</b>	Modelo YOLO (You Only Look Once) para el diagnóstico preliminar de la enfermedad de Huanglongbing en la planta de limón en la Región Piura	[5]
<b>TA05</b>	A Novel Enhanced VGG16 Model to Tackle Grapevine Leaves Diseases with Automatic Method	[6]
<b>TA06</b>	Early Disease Classification of Mango Leaves Using Feed-Forward Neural Network and Hybrid Metaheuristic Feature Selection	[7]
<b>TA07</b>	Fast anther dehiscence status recognition system established by deep learning to screen heat tolerant cotton	[8]

<b>TA08</b>	A robust deep learning approach for tomato plant leaf disease localization and classification	[9]
<b>TA09</b>	Classification of physiological disorders in apples using deep convolutional neural network under different lighting conditions	[10]
<b>TA10</b>	Detection of powdery mildew on strawberry leaves based on DAC-YOLOv4 model	[11]
<b>TA11</b>	Detecting the Simultaneous Occurrence of Strawberry Fungal Leaf Diseases with a Deep Normalized CNN	[12]
<b>TA12</b>	The Identification of Corn Leaf Diseases Based on Transfer Learning and Data Augmentation	[13]
<b>TA13</b>	Deep Learning Application for Plant Diseases Detection	[14]
<b>TA14</b>	Deep Learning Based Classification for Paddy Pests & Diseases Recognition	[15]
<b>TA15</b>	PlantDoc: A Dataset for Visual Plant Disease Detection	[16]
<b>TA16</b>	Detection and Classification System for Cashew Plant Diseases Using Convolutional Neural Network	[17]
<b>TA17</b>	Fruit Leaf Pests and Diseases Identification Based on Data Enhancement and Transfer Learning	[18]
<b>TA18</b>	Palm tree disease detection and classification using residual network and transfer learning of inception ResNet	[19]
<b>TA19</b>	Design of Efficient Methods for the Detection of Tomato Leaf Disease Utilizing Proposed Ensemble CNN Model	[20]
<b>TA20</b>	DCNet: DenseNet-77-based CornerNet model for the tomato plant leaf disease detection and classification	[21]
<b>TA21</b>	Deep Learning Utilization in Agriculture: Detection of Rice Plant Diseases Using an Improved CNN Model	[22]

<b>TA22</b>	Deep Learning Based Automatic Grape Downy Mildew Detection	[23]
<b>TA23</b>	A Backward Regressed Capsule Neural Network for Plant Leaf Disease Detection	[24]
<b>TA24</b>	Detection of White Leaf Disease in Sugarcane Crops Using UAV-Derived RGB Imagery with Existing Deep Learning Models	[25]
<b>TA25</b>	Tomato Diseases and Pests Detection Based on Improved Yolo V3 Convolutional Neural Network	[26]
<b>TA26</b>	A deep learning-based approach in classification and validation of tomato leaf disease	[27]
<b>TA27</b>	Convolutional Neural Networks for Texture Feature Extraction. Applications to Leaf Disease Classification in Precision Agriculture	[28]
<b>TA28</b>	Tea leaf disease detection and identification based on YOLOv7 (YOLO-T)	[29]

### 2.2.3 Síntesis de datos

Concluido el respectivo análisis de los TA, se encontró información relevante sobre los modelos CNN utilizados para identificar enfermedades a partir de hojas de diferentes especies de plantas. También se obtuvo información sobre el nivel de precisión, los métodos empleados para obtener el conjunto de datos y la cantidad de imágenes utilizadas para el entrenamiento de estos, además se distinguió las métricas aplicadas en la evaluación de cada modelo. Un resumen de la información recopilada se detalla en la Tabla 46, en base a la cual se pudo dar contestación a las preguntas de investigación planteadas en la presente RSL, dentro de la **Sección 2.1.2**

- **PI\_1: ¿Cuáles son los modelos más precisos empleados hasta la fecha, para identificar enfermedades en las hojas?**

En esta pregunta de investigación se enumeró diferentes modelos orientados a la clasificación y detección de enfermedades en hojas de distintas especies de plantas, en la Tabla 46 se presentan los modelos empleados en cada estudio analizado. Se observa que algunos estudios emplean dos o más modelos, algunos autores combinan modelos para obtener mejores resultados o aplican alguna técnica para mejorar la precisión de este.



**Tabla 46.** Identificación de Modelos CNN en los TA

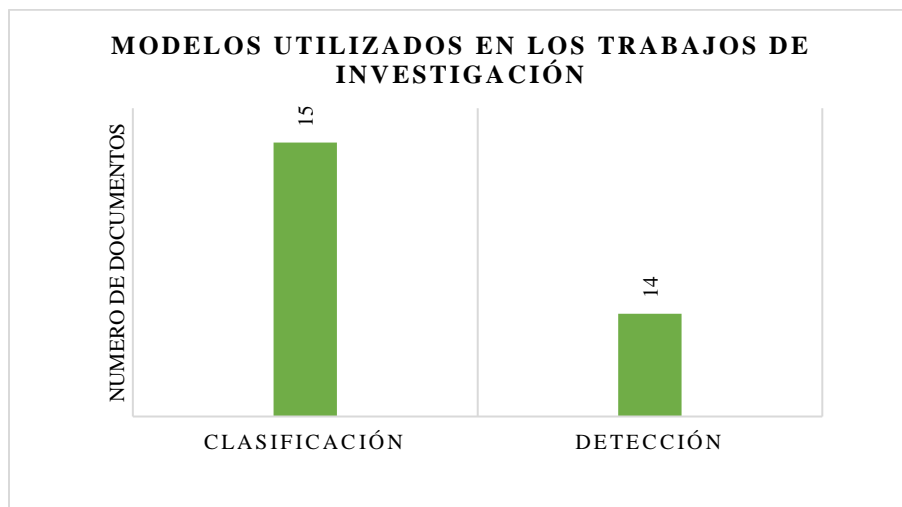
TA	Objeto de estudio	#Clases/ Enfermedades	Modelos	Precisión	N° Imágenes	Categoría
TA01	Hoja de maíz	8	GoogLeNet	98.90%	3060	Clasificación
			Cifar10	98.80%		
TA02	Hoja de soya	1	Inceptionv3	99.04%	500	Clasificación
			VGG-19	99.02%		
			Xception	98.56%		
TA03	Hoja del frijol	3	MobileNet	97%	1296	Clasificación
TA04	Limón	1	Yolov5	67%	5475	Detección
TA05	Hoja de uva	4	FasterRCNN con:		10 000	Detección
			VGG16	98.15%		
			GoogleNet	97.73%		
			ResNet50	99.16%		
			AlexNet	97.76%		
VGG16mejorado	99.60%					
TA06	Hoja de mango	4	VGG16	77.16%	450	Clasificación
			AlexNet	69.73%		
			ResNet 50	64.13%		
TA07	Anteras de Algodón	1	YOLOv5 con Darknet	61.88%	38.895	Detección
			Faster-RCNN con Resnet 101	87.12%		
TA08	Hoja Tomate	10	Faster-RCNN con:		54,306	Detección
			GoogleNet	87%		
			Xception	88%		
			Resnet-34	99%		
TA09	Hoja de manzana	1	Xception	0.994%	-	Clasificación
TA10	Hoja de Fresa	2	Faster-RCNN	55.7%	764	Detección
			Yolov3-tiny	69.4%		

			Yolov4-tiny	68.9%		
			Yolov5s	77.5%		
			Yolov4-DAC	75.5%		
<b>TA11</b>	Hoja de fresa	5	Modelo propio	98%	1134	Detección
<b>TA12</b>	Hojas de maiz	4	Vgg16	90.17%	4354	Clasificación
			Vgg19	93.08%		
			Resnet18	95.95%		
<b>TA13</b>	Hoja de café	4	SGD	90%	13692	Detección
<b>TA14</b>	Hojas de arroz	13	CaffeNet	87%	4511	Detección
<b>TA15</b>	Varias especies	27	MobileNet	mAP (at iou)=32.8	2598	Detección
			Faster R-CNN-InceptionResnetV2	mAP (at iou)=38.9		
<b>TA16</b>	Anacardo	3	ResNet-50	97.22%	1050	Clasificación
<b>TA17</b>	Hojas de manzana, cereza y uva	10	LeNet-5	60%	1000	Clasificación
			Inception V3	94.2%		
			Vgg16	91.04%		
<b>TA18</b>	Hoja de palmeras	1	ResNet	99,05%	2631	Clasificación
<b>TA19</b>	Hoja de tomate		MobileNetV3Small	98%	18160	Clasificación
			EfficientNetV2L	99%		
			InceptionV3	96%		
			MobileNetV2	97%		
<b>TA20</b>	Hoja de tomate	9	CornerNet basado en DenseNet-77	99.62%	----	Clasificación
			GoogleNet	87.09%		
			ResNet101	89.95%		
			Xception	88.25%		

			Vgg19	90.39%		
			SEresNet50	96.77%		
<b>TA21</b>	Hoja de arroz	6	VGG19	96.20%	----	Clasificación
			YOLOv5-CA	85.59%		
<b>TA22</b>	Hoja de uva	1	FASTER-RCNN	79.97%	820	Detección
			Yolo v4	82.69%		
			Yolov5	85.35%		
			AlexNet	84.5%		
			ResNet	88.6%		
			VGGNet	99.3%		
<b>TA23</b>	Hoja de tomate	10	InceptionV3	97.87%	9080	Clasificación
			VGG16	99.14%		
			Modelo cnn con regresion	99.7%		
			Yolov5	95%		
<b>TA24</b>	Hoja de caña	3	Yolor	87%	1680	Detección
			DTR	77%		
			FasterRCNN	90%		
			SSD	84.32%		
<b>TA25</b>	5 especies de hojas	12	Faster RCNN	90.67%	15000	Detección
			Yolov3	88.31%		
			VGG 16	98.77%		
			AlexNet	97.69%		
<b>TA26</b>	Hoja de tomate	8	GoogLeNet	93.73%	10800	Clasificación
			MobileNetv2	95.25%		
			QueueNet	90.86%		
			ResNet18	89.76%		
<b>TA27</b>	12 especies de hojas	34	AlexNet	88.5%	14997	Clasificación
			Vgg16	93.94%		
			LeNet	90.23%		
<b>TA28</b>	hoja de Té	5	Yolov5	95%	4000	Detección
			YOLOv7	96.7%		

En la Figura 84 se observa de manera jerárquica los modelos CNN más utilizados y con mejores resultados obtenidos en los estudios seleccionados, teniendo en cuenta que los mismos, han permitido identificar entre una y más enfermedades a través de la entrada de una imagen de la hoja de la planta sana o enferma.

La Tabla 46 indica de manera general los resultados obtenidos en cuanto al número de imágenes que conforman el dataset utilizado durante el entrenamiento, prueba y validación del modelo, de la misma manera se pudo identificar ciertas características y técnicas de obtención y tratamiento de los datos con el fin de obtener mejores resultados.



**Figura 84.** Resultado de los modelos CNN aplicados en los estudios primarios

Para llevar a cabo el proyecto de investigación orientado a la detección de la hoja enferma se ha creído conveniente filtrar solamente los modelos orientados a la detección, siendo el modelo con mayor precisión Faster RCNN seguido por Yolov7, Yolov5, SSD y CoffeNet, con una precisión de 99.16%, 96.70%, 95%, 90% y 87% respectivamente, como lo muestra Figura 85.

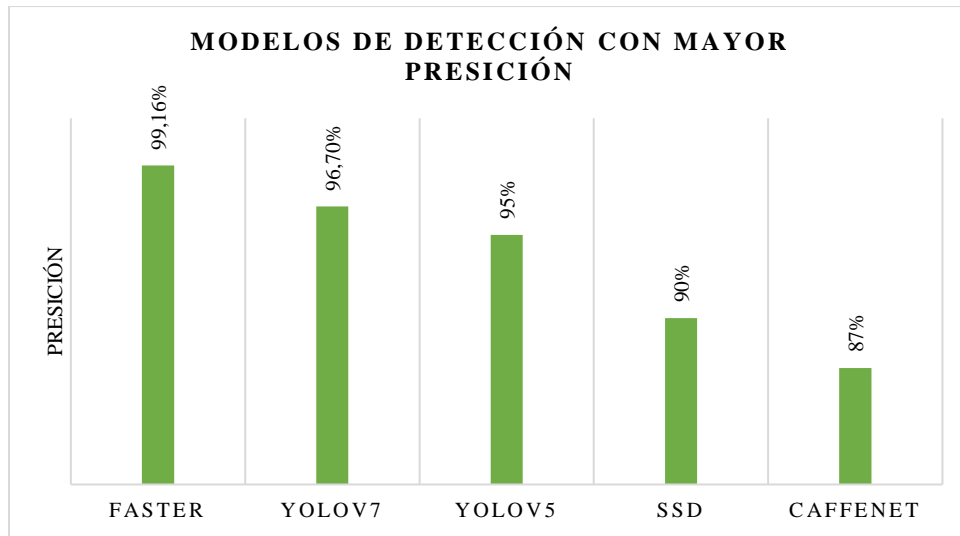


Figura 85. Modelos con mejores precisiones según la RSL

- **PI\_2: ¿Cuáles son las métricas usadas para evaluar el rendimiento de modelos o arquitecturas para detectar objetos?**

Tabla 47. Métricas para la evaluación de modelos CNN

Métricas	Precisión	Sensibilidad/ Recall	mAP	F1 Score	Exactitud/Accuracy	Especificado
Estudios	[2], [4], [5], [6], [9], [10], [12], [14], [15], [20], [21], [22], [23], [26], [29]	[5], [9], [10], [12], [21], [22], [23]	[8], [11], [23]	[9], [10], [11], [12], [21], [22]	[9], [10], [12], [21]	[5], [22]

De acuerdo al análisis realizado en la Tabla 47, se evidencia que la mayor parte de las investigaciones utilizan la métrica de *Precisión* para evaluar el modelo, pues esta muestra la proporción de hojas enfermas que fueron clasificadas correctamente entre todas las hojas realmente enfermas, esta medida es relevante para minimizar los falsos positivos (clasificar hojas sanas como enfermas). Seguidamente se encuentra la métrica de *Sensibilidad (recall)*, que indica el porcentaje de hojas enfermas que han sido identificadas correctamente con respecto al total de hojas enfermas reales, lo cual es crucial para saber si no se está perdiendo verdaderos positivos, a esta métrica le sigue el *Puntaje F1*, debido a que en varios trabajos se tiene un desfase en el número de imágenes

pertencientes a cada clase, por lo está métrica es aconsejable en esos casos, ya que se centra en el equilibrio de la precisión y la sensibilidad.

En cuanto a los trabajos orientado a la clasificación de enfermedades, en su mayor parte utilizan conjunto de datos libres que se encuentran en repositorios de forma gratuita y que sirven para fines investigativos como es PlantVillage y Plant Diseases, entre otros, tal es el caso del TA01, TA03, TA06, TA09, TA12, TA13, TA17, TA19, TA20,TA23, TA26, TA27, estos conjunto de datos contienen imágenes capturados en entornos preparados con fondos blancos e iluminación artificial para simular entornos reales.

Para llevar a cabo la tarea de detección de la enfermedad foliar, se identificaron diferentes técnicas de captura de imágenes, como por ejemplo en el TA02, las imágenes fueron tomadas por un dron a 2 metros de distancia del cultivo de soya y posteriormente etiquetadas con la ayuda del experto agrónomo para señalar las hojas sanas y enfermas, formando finalmente un conjunto de 500 imágenes. En el TA04, se capturaron imágenes de hojas de limón en una caja con fondo negro con iluminación en los extremos superiores para simular diferentes condiciones lumínicas. Se ajustó la intensidad de la luz para simular entornos reales y se utilizó una cámara de alta definición para fotografiar un conjunto de 450 hojas de mango. En el TRA04, TA13, TR14, TA22, TA24, TA28, se tomaron fotografías de las enfermedades foliares en entornos reales, permitiendo tener una visión más auténtica de las condiciones naturales de la planta, ya que algunos de estos trabajos lo capturan imágenes en invernaderos o cultivos al aire libre por lo que se obtiene una mayor diversidad de las imágenes de hojas y le permite al modelo adaptarse también a este tipo de entornos.

También se identificó algunas técnicas de pre procesamiento de datos utilizados para entrenar modelos CNN como el redimensionamiento de acuerdo al modelo aplicado y el aumento de datos cuando no se cuenta con gran cantidad de imágenes., entre estos métodos están la rotación, ampliación, cambio de color, recorte, cambio de tamaño aleatorio, y algoritmos de aumento automático, todo esto haciendo uso de scripts presentes en librerías de Python.

### **3. Discusión**

La presente RSL se indagó sobre la implementación de Redes Neuronales Convolucionales para la identificación de enfermedades foliares. La mayor parte de los trabajos analizados han optado por la implementación de modelos de clasificación y otra parte muestran la aplicación de

modelos de detección. En el caso de los trabajos orientados a la clasificación de objetos, tienen como finalidad determinar si una hoja está sana o enferma, mientras que los modelos de detección se encargan de identificar de manera precisa las áreas especificadas de la hoja que están afectadas por enfermedades.

En cuanto a los modelos de detección se identificaron modelos sobresalientes como Resnet, VGG16 y CGG19, que destacan de los demás por su precisión en la identificación de hojas enfermas. Por otra parte, en cuanto a los trabajos basados en la detección de enfermedades se obtuvo que el modelo Faster RCNN demostró una precisión sobresaliente del 99.16%, seguido por Yolov7 con un 96.07%, Yolov5 con un 95% y CofeeNet con un 85%. Sin embargo, de acuerdo al enfoque investigativo orientado a la hoja de fresa, se encontraron pocos documentos relacionados, a excepción del TA10, donde Faster RCNN obtuvo la precisión más baja (55.7%) mientras que Yolov5 obtuvo un mejor resultado con el 77.5% de precisión, lo que destaca la necesidad de una mayor investigación en esta área específica.

Mediante esta RSL, se identificó también las métricas utilizadas para evaluar el rendimiento del modelo en diferentes aspectos. La métrica más utilizada es la *precisión* que permite evaluar la capacidad del modelo para identificar correctamente los casos positivos; a continuación, se encuentra la *sensibilidad*, que se utilizó para comparar todas las predicciones que el modelo hizo con las etiquetas verdaderas y calcular cuántas de ellas fueron clasificadas correctamente, en el contexto de hojas enfermas, indica la proporción en el modelo clasifica correctamente hojas enfermas en base al número real de hojas enfermas que hay en el set de datos. Así mismo se utilizó el puntaje *F1*, que toma en cuenta tanto los falsos positivos como los falsos negativos, proporcionando un equilibrio entre la precisión y la exhaustividad del modelo, esta métrica es utilizada principalmente en aquellos casos donde existe un desequilibrio de clases.

Finalmente, durante una meticulosa revisión se apreció que el tratamiento de datos es una etapa importante y más laboriosa dentro de este proceso de desarrollo de modelos ya que de ello depende el correcto rendimiento del mismo, por lo cual se describen diversas técnicas como la limpieza, estandarización y aumento de datos de acuerdo a cada modelo implementado. Dentro del aumento de datos se aplican técnicas como rotación, corte, volteo como lo indican los TA01, TA03, TA05.

#### 4. Conclusiones

En base a los resultados obtenidos durante esta revisión sistemática de literatura se concluye que:

- La identificación de enfermedades foliares se pueden llevar a cabo mediante tarea de clasificación o detección de objetos, siendo esta última las más factible ya que permite seleccionar el área enferma de la hoja.
- Entre los modelos de detección los que demostraron mayor rendimiento en cuanto a precision son Faster RCNN, Yolov7, Yolov5 y CofeeNet.
- La métrica más utilizada para evaluar el modelo de detección es la Precisión.
- El modelo con mejores resultados en la detección de enfermedades en la hoja de fresa es Yolov5, ya que no se hallaron más trabajos relacionados en cuanto a este cultivo.
- La adquisición de imágenes en entornos reales permite un mejor rendimiento del modelo ya que muestran mayor diversidad visual de imágenes al ser capturadas en distintos escenarios ambientales y naturales.
- La técnica de aumento de datos es fundamental y útil, especialmente en aquellos casos donde se tiene un limitado número de datos.

#### 5. Referencias

- [1] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering—a systematic literature review," *Inf Softw Technol*, vol. 51, no. 1, pp. 7–15, 2009.
- [2] X. Zhang, Y. Qiao, F. Meng, C. Fan, and M. Zhang, "Identification of Maize Leaf Diseases Using Improved Deep Convolutional Neural Networks," *IEEE Access*, vol. 6, pp. 30370–30377, Mar. 2018, doi: 10.1109/ACCESS.2018.2844405.
- [3] E. C. Tetila *et al.*, "Automatic Recognition of Soybean Leaf Diseases Using UAV Images and Deep Convolutional Neural Networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 5, pp. 903–907, May 2020, doi: 10.1109/LGRS.2019.2932385.



- [4] E. Elfatimi, R. Eryigit, and L. Elfatimi, “Beans Leaf Diseases Classification Using MobileNet Models,” *IEEE Access*, vol. 10, pp. 9471–9482, Mar. 2022, doi: 10.1109/ACCESS.2022.3142817.
- [5] I. D. Carreño Chero and A. J. Obregon Mauricio, “Modelo YOLO (You Only Look Once) para el diagnóstico preliminar de la enfermedad de Huanglongbing en la planta de limón en la región Piura, 2023,” *Repositorio Institucional - UCV*, 2023, Accessed: May 03, 2024. [Online]. Available: <https://repositorio.ucv.edu.pe/handle/20.500.12692/135325>
- [6] S. Mousavi and G. Farahani, “A Novel Enhanced VGG16 Model to Tackle Grapevine Leaves Diseases With Automatic Method,” *IEEE Access*, vol. 10, pp. 111564–111578, Mar. 2022, doi: 10.1109/ACCESS.2022.3215639.
- [7] T. N. Pham, L. Van Tran, and S. V. T. Dao, “Early Disease Classification of Mango Leaves Using Feed-Forward Neural Network and Hybrid Metaheuristic Feature Selection,” *IEEE Access*, vol. 8, pp. 189960–189973, Mar. 2020, doi: 10.1109/ACCESS.2020.3031914.
- [8] J. and L. R. and L. Q. and Y. J. and M. Y. and L. Y. and W. Y. and Z. R. and M. H. and L. Y. and Z. L. and Z. L. and Z. X. and K. J. and Y. W. and M. L. Tan Zhihao and Shi, “Fast anther dehiscence status recognition system established by deep learning to screen heat tolerant cotton,” *Plant Methods*, vol. 18, no. 1, p. 53, Apr. 2022, doi: 10.1186/s13007-022-00884-0.
- [9] T. and J. A. and M. M. and R. J. and K. J. and H. A. Nawaz Marriam and Nazir, “A robust deep learning approach for tomato plant leaf disease localization and classification,” *Sci Rep*, vol. 12, no. 1, p. 18568, Nov. 2022, doi: 10.1038/s41598-022-21498-5.
- [10] E. Buyukarikan Birkan and Ulker, “Classification of physiological disorders in apples using deep convolutional neural network under different lighting conditions,” *Multimed Tools Appl*, Mar. 2023, doi: 10.1007/s11042-023-14766-7.

- [11] Y. Li, J. Wang, H. Wu, Y. Yu, H. Sun, and H. Zhang, “Detection of powdery mildew on strawberry leaves based on DAC-YOLOv4 model,” *Comput Electron Agric*, vol. 202, p. 107418, Nov. 2022, doi: 10.1016/J.COMPAG.2022.107418.
- [12] D. Kerre and H. Muchiri, “Detecting the Simultaneous Occurrence of Strawberry Fungal Leaf Diseases with a Deep Normalized CNN,” in *2022 7th International Conference on Machine Learning Technologies (ICMLT)*, in ICMLT 2022. New York, NY, USA: Association for Computing Machinery, 2022, pp. 147–154. doi: 10.1145/3529399.3529424.
- [13] R. Hu, S. Zhang, P. Wang, G. Xu, D. Wang, and Y. Qian, “The Identification of Corn Leaf Diseases Based on Transfer Learning and Data Augmentation,” in *Proceedings of the 3rd International Conference on Computer Science and Software Engineering*, in CSSE ’20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 58–65. doi: 10.1145/3403746.3403905.
- [14] F. Jakjoud, A. Hatim, and A. Bouaaddi, “Deep Learning Application for Plant Diseases Detection,” in *Proceedings of the 4th International Conference on Big Data and Internet of Things*, in BDIoT’19. New York, NY, USA: Association for Computing Machinery, 2020. doi: 10.1145/3372938.3372983.
- [15] A. A. Alfarisy, Q. Chen, and M. Guo, “Deep Learning Based Classification for Paddy Pests & Diseases Recognition,” in *Proceedings of 2018 International Conference on Mathematics and Artificial Intelligence*, in ICMAI ’18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 21–25. doi: 10.1145/3208788.3208795.
- [16] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, and N. Batra, “PlantDoc: A Dataset for Visual Plant Disease Detection,” in *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, in CoDS COMAD 2020. New York, NY, USA: Association for Computing Machinery, 2020, pp. 249–253. doi: 10.1145/3371158.3371196.
- [17] M. Timothy, O. John, A. Aibinu, and B. Adebisi, “Detection and Classification System for Cashew Plant Diseases Using Convolutional Neural Network,” in *The 5th International Conference on Future Networks & Distributed Systems*, in ICFNDS

2021. New York, NY, USA: Association for Computing Machinery, 2022, pp. 225–232. doi: 10.1145/3508072.3508107.
- [18] Y. Deng, Z. Chen, and S. Chen, “Fruit Leaf Pests and Diseases Identification Based on Data Enhancement and Transfer Learning,” in *Proceedings of the 4th International Conference on Intelligent Information Processing*, in ICIIP ’19. New York, NY, USA: Association for Computing Machinery, 2020, pp. 425–430. doi: 10.1145/3378065.3378146.
- [19] M. Ahmed and A. Ahmed, “Palm tree disease detection and classification using residual network and transfer learning of inception ResNet,” *PLoS One*, vol. 18, no. 3, p. e0282250, Mar. 2023, doi: 10.1371/JOURNAL.PONE.0282250.
- [20] V. E. Balas, H. Uluta, suluta, s, and V. A. Aslanta, s, “Design of Efficient Methods for the Detection of Tomato Leaf Disease Utilizing Proposed Ensemble CNN Model,” *Electronics 2023, Vol. 12, Page 827*, vol. 12, no. 4, p. 827, Feb. 2023, doi: 10.3390/ELECTRONICS12040827.
- [21] S. Albahli and M. Nawaz, “DCNet: DenseNet-77-based CornerNet model for the tomato plant leaf disease detection and classification,” *Front Plant Sci*, vol. 13, p. 957961, Sep. 2022, doi: 10.3389/FPLS.2022.957961/BIBTEX.
- [22] G. Latif, S. E. Abdelhamid, R. E. Mallouhy, J. Alghazo, and Z. A. Kazimi, “Deep Learning Utilization in Agriculture: Detection of Rice Plant Diseases Using an Improved CNN Model,” *Plants 2022, Vol. 11, Page 2230*, vol. 11, no. 17, p. 2230, Aug. 2022, doi: 10.3390/PLANTS11172230.
- [23] Z. Zhang, Y. Qiao, Y. Guo, and D. He, “Deep Learning Based Automatic Grape Downy Mildew Detection,” *Front Plant Sci*, vol. 13, p. 872107, Jun. 2022, doi: 10.3389/FPLS.2022.872107/BIBTEX.
- [24] J. Jepkoech, B. K. Kenduiywo, D. M. Mugo, and E. C. Tool, “A Backward Regressed Capsule Neural Network for Plant Leaf Disease Detection,” *Journal of Computer Science*, vol. 18, no. 9, pp. 821–831, Sep. 2022, doi: 10.3844/JCSSP.2022.821.831.
- [25] N. Amarasingam, F. Gonzalez, A. S. A. Salgadoe, J. Sandino, and K. Powell, “Detection of White Leaf Disease in Sugarcane Crops Using UAV-Derived RGB Imagery with

- Existing Deep Learning Models,” *Remote Sensing* 2022, Vol. 14, Page 6137, vol. 14, no. 23, p. 6137, Dec. 2022, doi: 10.3390/RS14236137.
- [26] J. Liu and X. Wang, “Tomato Diseases and Pests Detection Based on Improved Yolo V3 Convolutional Neural Network,” *Front Plant Sci*, vol. 11, p. 521544, Jun. 2020, doi: 10.3389/FPLS.2020.00898/BIBTEX.
- [27] S. Wagle and H. Ramachandran, “A Deep Learning-Based Approach in Classification and Validation of Tomato Leaf Disease,” *Traitement du Signal*, vol. 38, pp. 699–709, May 2021, doi: 10.18280/ts.380317.
- [28] S. Barburiceanu, S. Meza, B. Orza, R. Malutan, and R. Terebes, “Convolutional Neural Networks for Texture Feature Extraction. Applications to Leaf Disease Classification in Precision Agriculture,” *IEEE Access*, vol. 9, pp. 160085–160103, Mar. 2021, doi: 10.1109/ACCESS.2021.3131002.
- [29] Md. J. A. Soeb *et al.*, “Tea leaf disease detection and identification based on YOLOv7 (YOLO-T),” *Sci Rep*, vol. 13, no. 1, p. 6078, 2023, doi: 10.1038/s41598-023-33270-4

**Anexo 2.** Ensayo de adquisición del conjunto de datos mediante la captura de fotografía tomadas en huertos del cantón Saraguro.

### **1. Introducción**

En el presente ensayo se describe el desarrollo de la adquisición de imágenes la hoja de fresa del que presentan síntomas de mancha angular mediante la captura de fotografías en distintos huertos y con diferentes etapas de la misma, la cual fue dirigida por el Ingeniero Miguel Condolo, como experto en el tema quien determinaba si la planta presentaba la infección, como parte de este proyecto experimental.

### **2. Hipótesis**

El conjunto de datos requerido para el entrenamiento, validación y prueba de modelos de detección de objetos puede ser adquirido mediante fotografías de las diferentes características de la mancha angular en la hoja de fresa.

### **3. Materiales**

Cámara de teléfono móvil marca Samsung A23 con una apertura de F1.8, F2.2 y un tamaño de imagen de 50.0 MP (Megapíxeles).

### **4. Procedimiento**

- La investigadora determina con la ayuda del experto los lugares o huertos del cantón Saraguro donde exista plantaciones de fresa infectados con la enfermedad de la mancha angular.
- Se toma una muestra de hojas de fresa enferma en 4 huertos que son parte de la escuela agroecológica impulsada por el municipio del cantón Saraguro.
- La investigadora se ubica en las plantaciones de fresa, luego procede con ayuda del experto, a identificar las hojas de fresa que presentan características de mancha angular.
- Una vez identificadas las hojas enfermas, se procede a la captura de fotografías de las mismas, obteniendo las primeras fotografías en diferentes espacios, en caso de ser necesario se arranca la hoja para ubicarlo en un espacio con mejor visión.
- Este proceso se realiza hasta obtener una muestra considerable para el conjunto de datos, o dependiendo de la cantidad de hojas enfermas que haya en la localidad.

- Finalmente se inspecciona que las fotografías posean una buena resolución y nitidez, mediante la revisión de todo el conjunto de imágenes adquiridas.

## 5. Resultados

Luego de capturar fotografías de las hojas de fresa enfermas con una dimensión de 3000 x 4800 píxeles con una resolución horizontal y vertical de 72 PPP (puntos por pulgada) mediante la cámara de teléfono móvil Samsung A23 (En la Tabla 48 se detalla la cantidad de imágenes adquiridas), se transfirió dichas fotografías a un directorio local en el ordenador de la investigadora y un respaldo en la nube, obteniendo como resultado un conjunto de imágenes sólido donde se identifica de manera clara las hojas enfermas con mancha angular, mismas que con el respectivo procesamiento se aumentara el conjunto de datos.

**Tabla 48.** Número de imágenes de la hoja de fresa con mancha angular capturadas en áreas de estudio

<b>Lugar</b>	<b>Cantidad de imágenes</b>	<b>Fecha de recolección</b>
<b>El tablón( Invernadero)</b>	229	03/07/2023
<b>La Matara(Aire Libre)</b>	54	30/07/2023
<b>Puente Chico</b>	75	30/06/2023
<b>Cañicapac</b>	176	17/05/2023
<b>Total</b>	534	

## 6. Conclusión

Mediante el ensayo se pudo demostrar que la adquisición de imágenes a través de fotografías de la hoja de fresa con características de mancha angular, permitió establecer un conjunto de datos con 534 imágenes que se puede emplear durante las fases de entrenamiento, validación y prueba del modelo de detección de objetos, además de que las imágenes fueron tomadas en diferentes escenarios, tanto en invernadero y al aire libre en diferentes condiciones climáticas.

**Imágenes del ensayo de adquisición del conjunto de datos mediante la captura de fotografías de la hoja de fresa:**



**Figura 86.** Área de cultivo de fresa al aire libre en la comunidad de La Matara



**Figura 87.** Conjunto de datos adquirido en el cultivo de fresa de la comunidad La Matara





**Figura 88.** Invernadero de cultivo de fresa en la parroquia El Tablón



**Figura 89.** Captura de fotografías de las hojas enfermas en el cultivo de fresa de la parroquia El Tablón

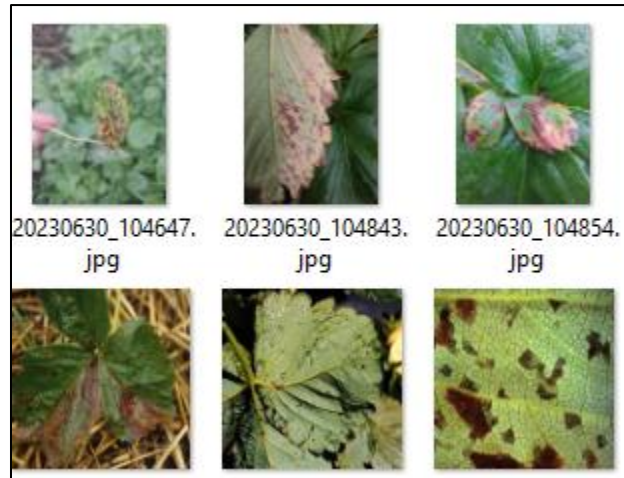




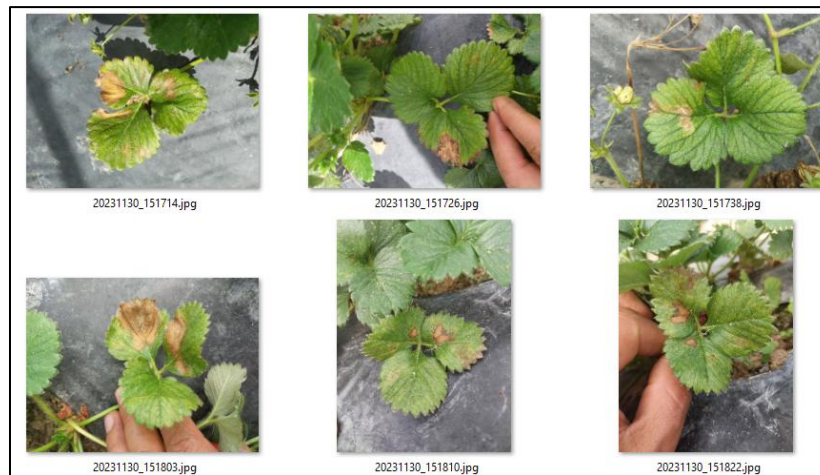
**Figura 90.** Captura de fotografías de las hojas enfermas en el cultivo de fresa Cañicapac



**Figura 91.** Conjunto de datos recopilado en el invernadero de la parroquia El Tablón



**Figura 92.** Conjunto de Datos adquirido en la comunidad de Puente Chico



**Figura 93.** Conjunto de Datos adquirido en la comunidad de Cañicapac

### Anexo 3. Pruebas de Aceptación de las interfaces del prototipo móvil



**UNL**  
Universidad Nacional de Loja



Carrera de Ingeniería en Sistemas / Computación



**saraguro**  
Gobierno Autónomo Descentralizado Municipal de Saraguro

Caso de prueba de aceptación (PA)	
<b>Código:</b> PA-001	<b>Historia de Usuario(nro.Nombre):</b> N°02 - Información Principal
<b>Nombre PA:</b> Interfaz que muestra información relevante a la enfermedad	
<b>Descripción:</b> En esta interfaz se muestra un concepto de la enfermedad de mancha angular y algunas recomendaciones de prevención de la misma.	
<b>Condiciones de Uso:</b>	
<ul style="list-style-type: none"> <li>- Se debe seleccionar la opción de “Información de enfermedad” de la pantalla principal.</li> </ul>	
<b>Entrada/Pasos de Ejecución:</b>	
<ul style="list-style-type: none"> <li>- Abrir la aplicación</li> <li>- Seleccionar “Información principal”</li> </ul>	
<b>Resultado esperado:</b>	
<ul style="list-style-type: none"> <li>- Interfaz con información relevante de la enfermedad.</li> </ul>	
<b>Evaluación de Prueba:</b>	
Aprobada <input checked="" type="checkbox"/>	Denegada <input type="checkbox"/>
<b>Comentarios y/o sugerencias:</b>	

Firma de aceptación:



**Ing. Miguel Angel Condoto**  
TECNICO DE APOYO AGROPECUARIO  
DIRECCIÓN DE GESTION AMBIENTAL Y PRODUCCIÓN DEL GADMIS



**Figura 94.** Prueba de aceptación Historia de Usuario N° 02





Universidad  
Nacional  
de Loja



Carrera de Ingeniería en  
Sistemas / Computación



Caso de prueba de aceptación (PA)	
<b>Código:</b> PA-002	<b>Historia de Usuario(nro.Nombre):</b> N°03 - Seleccionar foto de galería N°04 - Capturar foto con la cámara
<b>Nombre PA:</b> Interfaz con la opción de seleccionar galería o tomar foto	
<b>Descripción:</b> En esta interfaz se muestran dos opciones de entrada de imagen para su detección: utilizando la galería o utilizando la cámara del dispositivo móvil.	
<b>Condiciones de Uso:</b> <ul style="list-style-type: none"> <li>- La foto debe ser tomada de forma clara</li> <li>- La foto no debe ser tomada a mucha distancia para tener mejores características.</li> <li>- La imagen debe ser de hojas de fresa</li> </ul>	
<b>Entrada/Pasos de Ejecución:</b>	
<b>Uso de Galería:</b> <ul style="list-style-type: none"> <li>- Abrir la aplicación</li> <li>- Seleccionar el ícono de galería</li> <li>- Seleccionar la imagen que se quiera identificar</li> <li>- Seleccionar la opción de aceptación</li> </ul>	<b>Uso de Cámara:</b> <ul style="list-style-type: none"> <li>- Abrir la aplicación</li> <li>- Seleccionar el ícono de cámara</li> <li>- Tomar foto de la hoja de fresa</li> <li>- Seleccionar la opción de aceptación</li> </ul>
<b>Resultado esperado:</b> <ul style="list-style-type: none"> <li>- El modelo predictivo recibe la imagen de entrada y realiza las detecciones correspondientes.</li> </ul>	
<b>Evaluación de Prueba:</b>	
Aprobada <input checked="" type="checkbox"/>	Denegada <input type="checkbox"/>
<b>Comentarios y/o sugerencias:</b>	

Firma de aceptación:

**Ing. Miguel Ángel Condolo**  
TECNICO DE APOYO AGROPECUARIO  
DIRECCIÓN DE GESTION AMBIENTAL Y PRODUCCIÓN DEL GADMIS

**Figura 95.** Prueba de aceptación Historia de Usuario N° 03 y 04



UNL

Universidad Nacional de Loja



Carrera de Ingeniería en Sistemas / Computación



Gobierno Autónomo Descentralizado Municipal de Saraguro

Caso de prueba de aceptación (PA)	
<b>Código:</b> PA-003	<b>Historia de Usuario(nro.Nombre):</b> N°05 – Mostrar resultados
<b>Nombre PA:</b> Interfaz con los resultados obtenidos en la imagen	
<b>Descripción:</b> En esta interfaz se muestra la imagen de entrada con las detecciones realizadas por el modelo predictivo.	
<b>Condiciones de Uso:</b> - Ninguno	
<b>Entrada/Pasos de Ejecución:</b> - Cargar imagen de entrada utilizando cámara o galería - Esperar unos segundos para detección	
<b>Resultado esperado:</b> - Imágen con las respectivas detecciones, ya sea la hoja “sana” o con “m_angular”, encerrados en cuadros delimitadores con el porcentaje de precisión.	
<b>Evaluación de Prueba:</b>	
Aprobada <input checked="" type="checkbox"/>	Denegada <input type="checkbox"/>
<b>Comentarios y/o sugerencias:</b>	

Firma de aceptación:

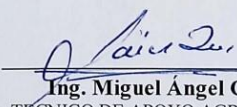
  
**Ing. Miguel Angel Condole**  
 TECNICO DE APOYO AGROPECUARIO  
 DIRECCIÓN DE GESTION AMBIENTAL Y PRODUCCION DEL GADMIS



Figura 96. Prueba de aceptación Historia de Usuario N° 05

**Anexo 4.** Resultados de la fase de evaluación de los modelos Yolov5, Yolov7, Yolov8 y Faster R-CNN con el conjunto de datos de pruebas

La siguiente tabla muestra un fragmento de los resultados de predicción obtenidos durante la fase de validación en los modelos implementados para la detección de mancha angular en la hoja de fresa. El informe completo en conjunto con la inferencia de cada modelo sobre el set se encuentra ubicados en el repositorio de Google Drive<sup>16</sup>.

Modelo CNN		Yolov5				Yolov7				Yolov8				Faster RCNN			
Datos Reales		Predicciones correctas	Predicciones Incorrectas	Objetos sin Identificar	% Promedio de predicciones correctas	Predicciones correctas	Predicciones Incorrectas	Objetos sin prediccion	% Promedio de predicciones correctas	Predicciones correctas	Predicciones Incorrectas	Objetos sin prediccion	% Promedio de predicciones correctas	Predicciones correctas	Predicciones Incorrectas	Objetos sin prediccion	% Promedio de predicciones correctas
Nº de Imagen	# etiquetas reales																
test_1	3	3	0	0	92,60%	2	0	1	36,60%	3	0	0	92,33%	3	0	0	98,00%
test_2	3	3	0	0	89,00%	0	0	3	0,00%	3	0	0	65,00%	3	1	0	97,00%
test_3	1	1	0	0	94,00%	0	0	1	0,00%	1	0	0	92,00%	1	2	0	100,00%
test_4	3	3	0	0	92,00%	1	0	2	18,00%	3	0	0	88,66%	3	0	0	97,00%
test_5	1	1	0	0	91,00%	0	1	1	0,00%	1	0	0	79,00%	1	3	0	99,00%
test_6	3	3	0	0	94,00%	3	0	0	59,60%	3	0	0	92,66%	3	0	0	99,00%
test_7	3	3	0	0	90,00%	3	0	0	62,00%	3	0	0	86,00%	3	1	0	98,00%
test_8	1	1	0	0	96,00%	0	0	1	0,00%	1	0	0	81,00%	1	0	0	99,00%
test_9	1	1	0	0	98,00%	0	0	1	0,00%	1	0	0	95,00%	1	0	0	99,00%
test_10	1	1	0	0	94,00%	1	0	0	74,00%	1	0	0	88,00%	1	2	0	98,00%
test_11	1	1	1	0	96,00%	1	0	0	75,00%	1	0	0	78,00%	1	1	0	99,00%
test_12	1	1	0	0	58,00%	0	0	1	0,00%	1	1	0	79,00%	1	0	0	99,00%
test_13	1	1	0	0	94,00%	0	0	1	0,00%	1	1	0	92,00%	1	2	0	99,00%
test_14	1	1	0	0	80,00%	0	0	1	0,00%	1	0	0	90,00%	1	0	0	98,00%
test_15	1	1	0	0	88,00%	0	0	1	0,00%	1	0	0	96,00%	1	0	0	98,00%
test_16	2	1	1	1	93,00%	0	0	2	0,00%	1	0	1	75,00%	2	1	0	91,00%
test_17	2	1	0	1	92,00%	0	0	2	0,00%	1	0	1	36,50%	2	1	0	97,00%
test_18	1	1	0	0	93,00%	1	0	0	64,00%	1	0	0	85,00%	1	1	0	99,00%
test_19	1	0	0	1	0,00%	0	0	1	0,00%	1	1	0	76,50%	1	2	0	98,00%
test_20	2	2	0	0	94,00%	0	0	2	0,00%	2	0	0	75,50%	2	0	0	95,00%
test_21	2	2	0	0	78,50%	0	0	2	0,00%	2	0	0	85,50%	2	2	0	98,00%
test_22	3	3	0	0	74,50%	2	1	1	44,00%	3	0	0	91,33%	3	2	0	97,00%
test_23	1	1	1	0	93,66%	0	0	1	0,00%	1	1	0	92,00%	1	2	0	99,00%
test_24	1	1	0	0	80,00%	0	0	1	0,00%	1	0	0	84,00%	1	0	0	98,00%
test_25	1	1	0	0	70,00%	0	0	1	0,00%	1	0	0	84,00%	1	0	0	100,00%

**Figura 97.** Comparación de resultados con el conjunto de prueba en la fase de validación

<sup>16</sup> Documento donde se visualiza las predicciones de cada modelo en el set de prueba y la tabla comparativa: <https://n9.cl/inferencias>

**Anexo 5.** Ensayo de pruebas de la aplicación móvil bajo supervisión del experto en un entorno real

## **1. Introducción**

Este ensayo describe la práctica de campo utilizando la técnica Zero-Shot learning para realizar las respectivas pruebas de la capacidad y eficiencia del modelo integrado en la aplicación móvil llamada “*ManchaAngularApp*” para identificar la mancha angular en la hoja de fresa.

Para llevar a cabo este proceso se contó con la presencia del experto, el Ing, Miguel Condolo, Técnico de Apoyo Agropecuario de la Dirección de Gestión Ambiental y Producción del GADMIS, y el Señor Carlos Poma, propietario del cultivo de Fresa ubicado en el barrio Tuchin localizada en la parroquia de San Antonio De Cumbe, mismo que se encuentra a una distancia de 24 km del cantón Saraguro.

En conjunto, se procedió a probar la aplicación móvil, por medio de la captura de imágenes directamente del cultivo, y con la ayuda del experto se pudo determinar y comparar los resultados emitidos por la app.

## **2. Objetivo**

El objetivo de esta práctica es evaluar la capacidad del modelo Yolov5 integrado en la aplicación móvil para detectar la mancha angular en hojas de fresa en un entorno de cultivo real.

## **3. Materiales**

Cámara de teléfono móvil marca Samsung A23 (puede ser otra marca)

## **4. Procedimiento**

4.1 *Selección del sitio de cultivo:* Se eligió un huerto seleccionado por el experto en el cantón Saraguro, que cuenta con cultivos de fresa al aire libre, esta selección permitió capturar variedad de características de la enfermedad ya que en estas condiciones las plantas están expuestas a los factores climáticos que causan la proliferación de esta bacteria.

4.2 *Recolección de imágenes:* Durante la visita al cultivo se recolectaron 50 imágenes de hoja de fresa, 20 sanas y 20 con síntomas sospechosos de la mancha angular observados por el experto, tales como manchas angulares con bordes oscuros y centros amarillentos con humedad o necrosis. Estas imágenes están ubicadas en el repositorio de Drive para el respectivo uso y pruebas.

4.3 *Pruebas y Evaluación*: la aplicación móvil se utilizó para realizar predicciones en tiempo real sobre las imágenes recolectadas, las predicciones realizadas por el modelo se compararon con las observaciones del ingeniero agrónomo para evaluar su precisión.

## 5. Resultados

La aplicación móvil integrada con Yolov5 demostró una alta precisión en la identificación de la mancha angular en las hojas de fresa. el modelo fue capaz de reconocer correctamente las hojas afectadas distinguiéndolas de las sanas, incluso teniendo variantes de la enfermedad no vistas por el modelo durante el entrenamiento.

A continuación, se puede visualizar la tabla de resultados obtenidas durante esta práctica de campo, en donde se contrastan los resultados obtenidos con la aplicación móvil y la observación del experto.

**Tabla 49.** Comparativa de resultados de prueba del modelo Yolov5 integrado en la App móvil y el experto

N° de Imágen	# Hojas enfermas de acuerdo al experto	# Hojas enfermas de acuerdo al modelo	N° de Imágen	# Hojas enfermas de acuerdo al experto	# Hojas enfermas de acuerdo al modelo
<b>ENFERMAS</b>			<b>SANAS</b>		
prueba_1	2	2	prueba_21	0	0
prueba_2	2	1	prueba_22	0	0
prueba_3	3	3	prueba_23	0	0
prueba_4	2	1	prueba_24	0	1
prueba_5	3	3	prueba_25	0	1
prueba_6	3	3	prueba_26	0	0
prueba_7	3	3	prueba_27	0	0
prueba_8	1	2	prueba_28	0	0
prueba_9	2	1	prueba_29	0	1
prueba_10	2	2	prueba_30	0	1
prueba_11	6	6	prueba_31	0	0
prueba_12	3	3	prueba_32	0	0
prueba_13	3	2	prueba_33	0	0
prueba_14	3	3	prueba_34	0	0
prueba_15	2	2	prueba_35	0	0
prueba_16	3	3	prueba_36	0	0
prueba_17	3	3	prueba_37	0	0
prueba_18	1	1	prueba_38	0	0
prueba_19	3	3	prueba_39	0	0
prueba_20	3	4	prueba_40	0	0
<b>TOTAL</b>	<b>53</b>	<b>51</b>	<b>TOTAL</b>	<b>0</b>	<b>4</b>



Mediante esta práctica se pudo verificar la eficiencia del modelo al momento de identificar hojas enfermas en el cultivo de fresa, en 20 imágenes de prueba con 53 objetos que hacen referencia a las hojas enfermas, la app móvil identificó correctamente 51 objetos. En cuanto a hojas sanas, se adquirieron 20 muestras, de las cuales el modelo identificó incorrectamente 4 objetos como mancha angular, lo cual corresponde a un resultado aceptable en cuanto a la capacidad del modelo para llevar a cabo el objetivo planteado.

## **6. Conclusión**

Finamente se puede concluir que la app “*ManchaAngularApp*” ha demostrado ser eficiente al momento de identificar hojas con mancha angular en un cultivo real de fresa, por lo cual se convierte en una herramienta útil, especialmente para aquellos agricultores que tienen sus cultivos en sectores rurales donde no tienen acceso a una red de internet, y equipos de cómputo, por lo cual una herramienta portable como lo es una aplicación móvil es de gran apoyo para los mismos.

## Fotografías de las prácticas realizadas en la comunidad de Tuchin

Se adjuntan las evidencias de la práctica de campo, en el repositorio<sup>17</sup> de Google Drive se encuentran ubicados todas las imágenes capturadas para las pruebas finales del prototipo.



**Figura 98.** Evidencia de la práctica de prueba de la aplicación móvil en la comunidad de Tuchin

<sup>17</sup> Evidencias de la práctica y conjunto de imágenes para la prueba final: <https://n9.cl/setpruebafinal>

## Anexo 6. Manual de Usuario



**unl**

Universidad  
Nacional  
de Loja



Carrera de Ingeniería en  
Sistemas / Computación

[MANUAL DE USUARIO]

# Manual para el manejo del prototipo de aplicación móvil de Detección de Mancha Angular en la hoja de Fresa: ManchaAngularAPP

Versión 1.0

*Elaborado por:*

*Diana Gabriela González Chillogalli*

Junio 2024

Loja – Ecuador

## HISTORIAL DE CAMBIOS

<b>Versión</b>	<b>Fecha</b>	<b>Responsable</b>	<b>Descripción de la modificación</b>
<b>1.0</b>	24/06/2024	Diana Gabriela Gonzalez Chillogalli	Documento inicial
<b>1.0</b>	26/07/2024	Diana Gabriela Gonzalez Chillogalli	Documento inicial

## Contenido

OBJETIVO.....	143
ALCANCE .....	143
APLICATIVO MÓVIL PARA LA DETECCIÓN DE MANCHA ANGULAR EN LA HOJA DE FRESA: ManchaAngularApp.....	143
<b>PREREQUISITOS:</b> .....	143
INSTRUCCIONES .....	143
<b>INSTALACIÓN DEL APLICATIVO MÓVIL</b> .....	143
<b>CARGAR IMAGENES DESDE GALERÍA</b> .....	145
<b>CAPTURAR IMAGEN DESDE LA APLICACIÓN</b> .....	147

## **OBJETIVO**

El presente documento tiene como objetivo proporcionar a los agricultores que poseen cultivo de fresa en el cantón Saraguro una guía para el correcto uso del prototipo de aplicación móvil orientado a detectar la enfermedad de mancha angular en la hoja de fresa, denominada ManchaAngularApp.

## **ALCANCE**

Las instrucciones que se presentan en este documento están dirigidas a los productores de fresa del cantón Saraguro, en especial a los pequeños agricultores que forman parte de la escuela agroecológica del GAD municipal, con la finalidad de proporcionar una herramienta que ayude a identificar la enfermedad de mancha angular en los cultivos de fresa, permitiendo una detección temprana para aplicar el tratamiento correcto, disminuyendo el daño en sus cultivos.

## **APLICATIVO MOVIL PARA LA DETECCION DE MANCHA ANGULAR EN LA HOJA DE FRESA: ManchaAngularApp**

Software de Inteligencia Artificial basado en Deep Learning que en su configuración contiene el modelo de detección de objetos Yolov5 entrenado para detectar hojas sanas y hojas con mancha angular en el cultivo de fresa.

## **PREREQUISITOS:**

- Conexión a Internet (únicamente para la descarga del aplicativo móvil)
- Cámara que permita capturar imágenes claras y con buena calidad
- Hojas de fresa sanas o con característica de enfermedad

## **INSTRUCCIONES**

El presente manual de usuario posee indicaciones específicas para el correcto uso del aplicativo móvil de detección de la mancha angular en la hoja de fresa por parte de los agricultores de la escuela agroecológica del cantón Saraguro. Esta aplicación funciona a través de un dispositivo móvil(celular).

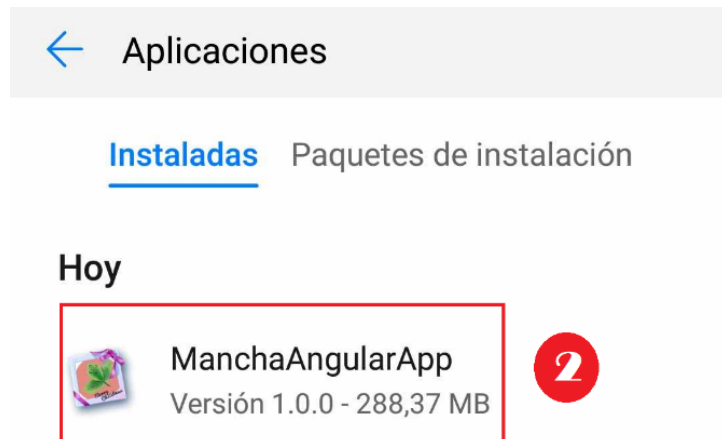
## **INSTALACIÓN DEL APLICATIVO MÓVIL**

1. En primer lugar, se procede a descargar el archivo ManchaAngularApp.apk del siguiente enlace:

- [https://drive.google.com/drive/folders/1-yZsHDrzgT4nDVjKZcfha0JiYFmsNgxk?usp=drive\\_link](https://drive.google.com/drive/folders/1-yZsHDrzgT4nDVjKZcfha0JiYFmsNgxk?usp=drive_link)

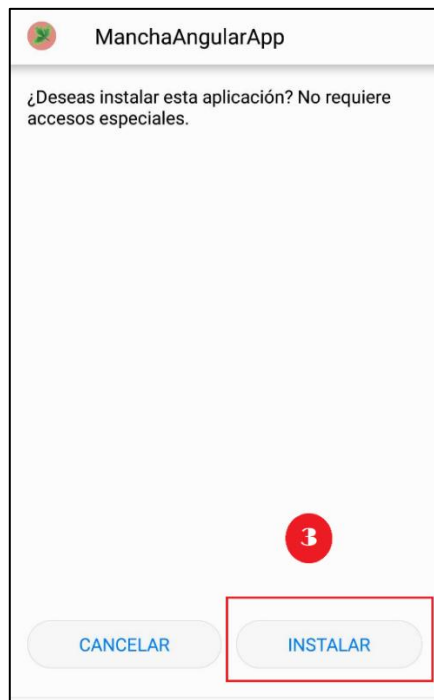
(Nota: En caso de aparecer la advertencia: “Es posible que el archivo sea dañino” → Pulsar: Descargar de todos modos)

2. Una vez descargado se procede a abrir el archivo .Apk

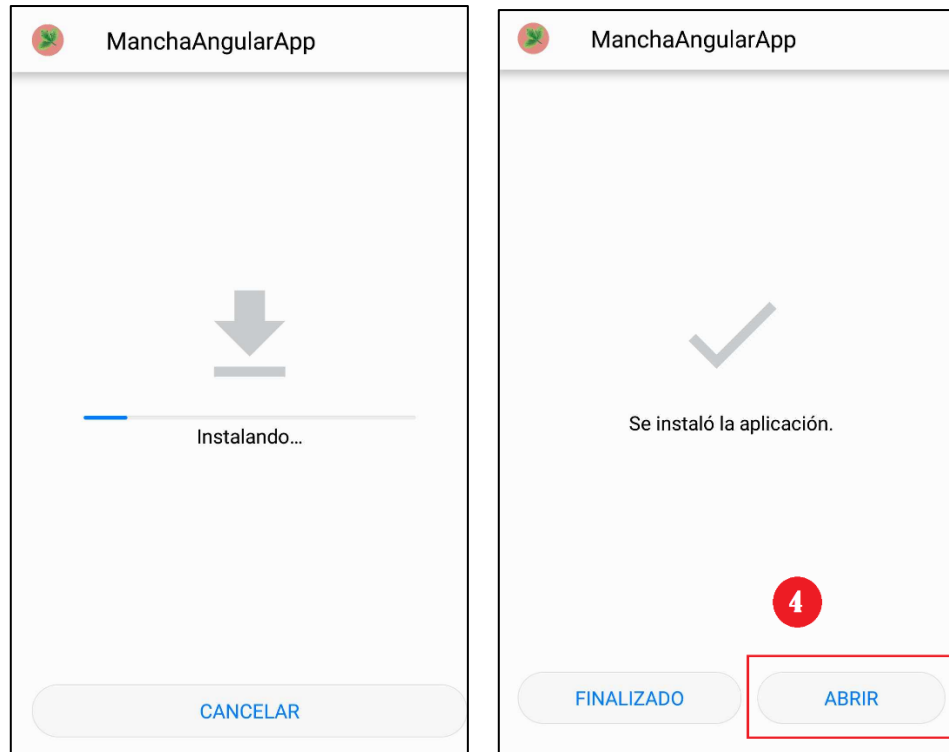


(Nota: En caso de no visualizar la pantalla anterior, diríjase a la sección de DESCARGAS de su dispositivo móvil, y pulse sobre el archivo para abrirlo)

3. A continuación, seleccionar la opción “instalar”

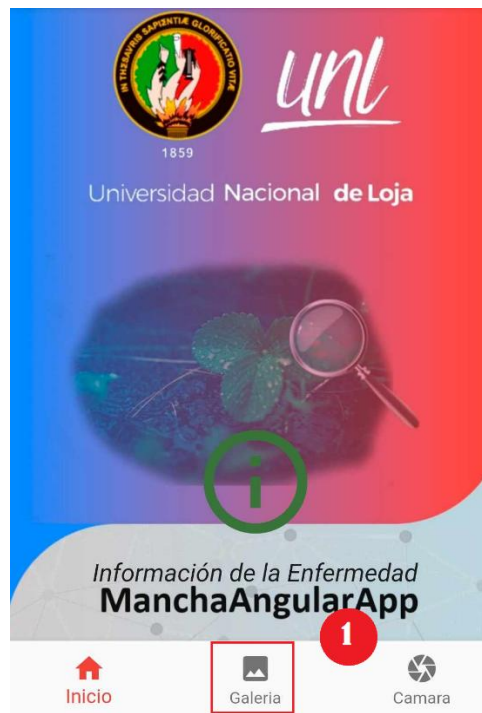


4. Espere el proceso de instalación, y una vez finalizado el mismo, procede a seleccionar el botón “ABRIR” para ir a la aplicación.



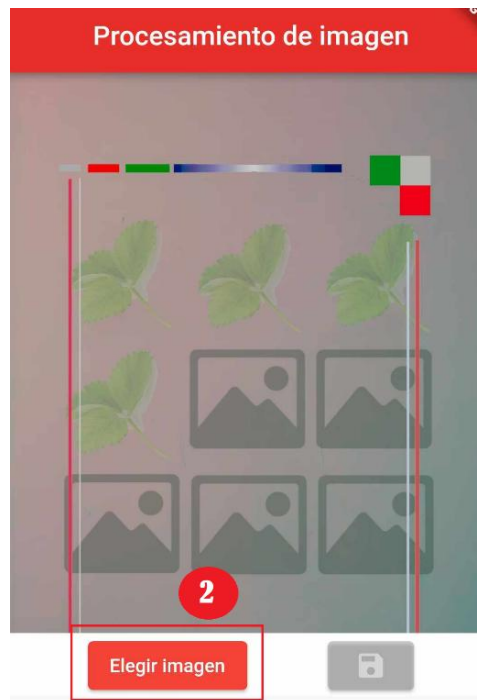
## CARGAR IMÁGENES DESDE GALERÍA

1. Abrir la aplicación, en la pantalla principal seleccionar el botón “Galería”

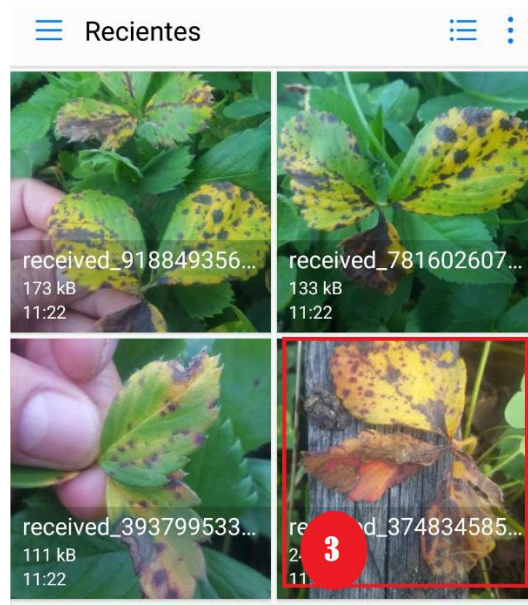




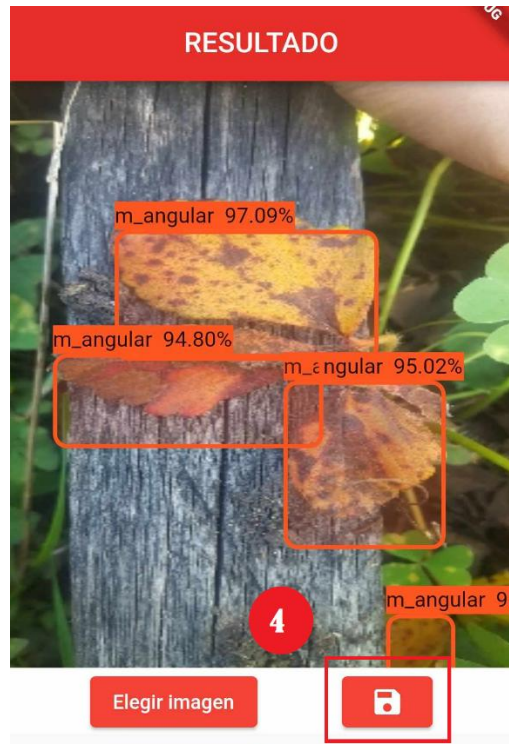
2. Se mostrará la pantalla adjunta, seleccione la opción “Elegir imagen”



3. A continuación, se mostrará una vista que contiene todas las fotografías almacenadas en la galería, seleccione una foto.



4. La imagen seleccionada se cargará en la vista principal de la aplicación, con los resultados de predicción (tiempo de espera 2 a 4 segundos). En caso de que se desee guardar el resultado, seleccionar el icono de “Guardar” para alojar esta imagen en su galería.

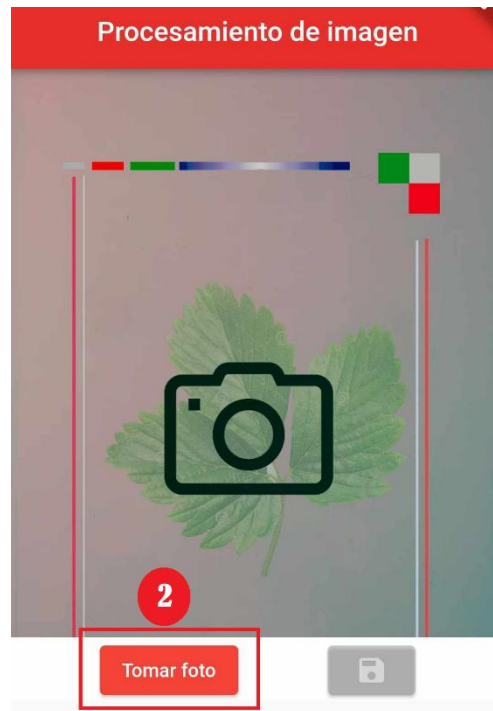


## CAPTURAR IMAGEN DESDE LA APLICACIÓN

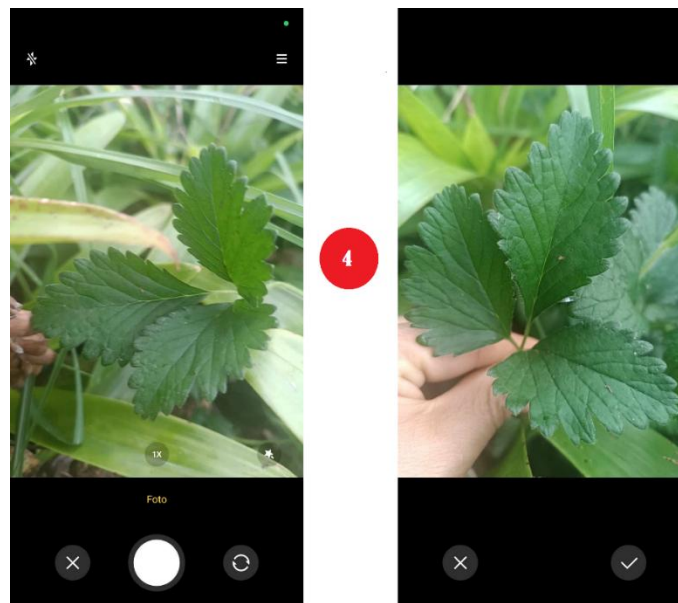
1. En primer se procede a abrir la aplicación, al observar la pantalla principal seleccionar el botón “Camara”



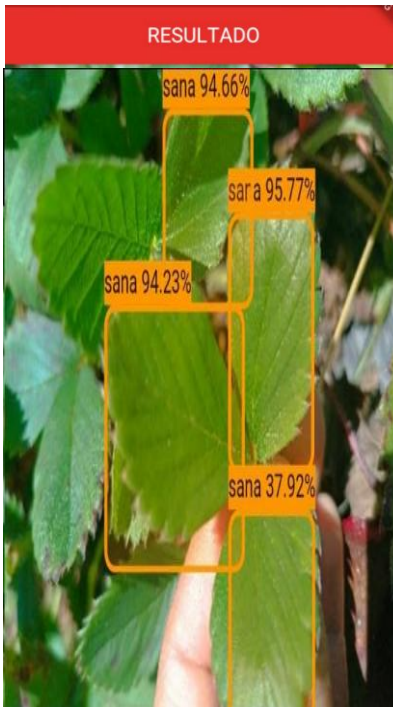
2. En la pantalla que se muestra seleccionar el botón “Tomar Foto”



3. La cámara del teléfono móvil se abrirá directamente desde la aplicación, se deberá enfocar las hojas de tal manera que la nitidez de la imagen sea la mejor, con el fin de obtener predicciones acertadas. Una vez capturada la imagen, seleccionar el ícono “✓” si esta de conforme con la imagen o ”X” si desea tomar una foto nueva.



4. La imagen capturada se cargará automáticamente en la vista principal con la respectiva predicción (tiempo de espera 3 a 4 segundos).



Acción	Estudiante	Firmas
Elaborado por:	Diana Gabriela González Chilloalli	

**Información de Contacto:**


Diana Gabriela González Chilloalli

Egresada de la Carrera de Ingeniería en Sistemas

**Celular:** 00999652485

**Correo:** [dggonzalesch@unl.edu.ec](mailto:dggonzalesch@unl.edu.ec)

## Anexo 7. Certificado de Data Set otorgado por el Técnico de Apoyo Agropecuario del GADMIS



Alcaldía  
**Saraguro**

**Dirección de Gestión Ambiental y Producción**


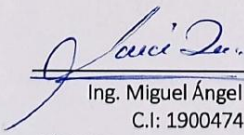
Saraguro, 15 de diciembre del 2023

Ing. Miguel Ángel Condolo Gualán TÉCNICO DE APOYO AGROPECUARIO DEL GADMIS con cédula de Identidad N.º 1900474188, en calidad de Coordinador de la Escuela Agroecológica del Pueblo Saraguro "SUMAK TARPUNA"





CERTIFICA:


Que el DATASET construido por la Srta. **Diana Gabriela González Chillogalli**, de nacionalidad ecuatoriana, con cédula Nro. **1150036984**, egresada de la Carrera de **Ingeniería en Sistemas** de la Universidad Nacional de Loja, posee un total de 500 fotografías capturadas mediante prácticas de campo en cuatro sectores de cultivo de fresa del cantón Saraguro (Cañicapac, La Matara, Puente Chico, El Tablón), en condiciones de invernadero y aire libre. La muestra recolectada está conformada por hojas de fresa enfermas con mancha angular.

Es todo lo que puedo certificar en honor a la verdad y faculto a la peticionaria hacer uso del presente documento para los fines pertinentes.



Ing. Miguel Ángel Condolo  
C.I: 1900474188  
TÉCNICO DE APOYO AGROPECUARIO  
DIRECCIÓN DE GESTIÓN AMBIENTAL Y PRODUCCIÓN DEL GADMIS

 José María Vilar y Sucre     [www.saraguro.gob.ec](http://www.saraguro.gob.ec)  
 (07) 2200-100     [municipio@saraguro.gob.ec](mailto:municipio@saraguro.gob.ec)





## Anexo 8. Certificado de pruebas de validación de la App Móvil

Dirección de Gestión Ambiental y Producción

Saraguro, 18 de julio del 2024

Ing. Miguel Ángel Condolo Gualán TÉCNICO DE APOYO AGROPECUARIO DEL GADMIS con cédula de Identidad N.º 1900474188, en calidad de Coordinador de la Escuela Agroecológica del Pueblo Saraguro "SUMAK TARPUNA"

CERTIFICA:

Haber realizado las pruebas de funcionamiento del prototipo de aplicación móvil denominado "ManchaAngularApp", resultado correspondiente al Trabajo de Titulación de la Srta. Diana Gabriela González Chilloalli, denominado: *Modelo predictivo basado en deep learning para identificar la Mancha Angular (Xanthomonas fragariae) en la hoja de Fresa para el Municipio del cantón Saraguro*, mismo que, en las pruebas experimentales llevadas a cabo en el cultivo de fresa ubicado en la comunidad de Tuchin, funcionó de manera correcta, realizando detecciones satisfactorias de hojas de fresa con la enfermedad de mancha angular, razón por la cual, considero que es una herramienta muy útil para los agricultores que forman parte del Gobierno Autónomo Descentralizado Intercultural del cantón Saraguro (GADMIS).

Certifico en honor a la verdad y faculto a la peticionaria hacer uso del presente documento para los fines pertinentes.

Ing. Miguel Ángel Condolo  
C.I.: 1900474188  
TÉCNICO DE APOYO AGROPECUARIO  
DIRECCIÓN DE GESTIÓN AMBIENTAL Y PRODUCCIÓN DEL GADMIS

José María Vivar y Sucre  
(07) 2200-100  
www.saraguro.gob.ec  
municipio@saraguro.gob.ec

**Anexo 9.** Certificado de traducción del resumen al idioma inglés

Saraguro, 23 de Julio del 2024

Gloria Mercedes Guailas Gualán  
Leda. en Ciencias de la Educación mención Idioma Inglés

A petición de la parte interesada y en forma legal.

**CERTIFICO**

Que, **Diana Gabriela González Chillogalli** con Cédula de Identidad Nro. 1150036984, estudiante de la carrera de **Ingeniería en Sistemas**, de la Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables de la Universidad Nacional de Loja, completó satisfactoriamente la presente traducción de español a inglés del resumen del Trabajo de Titulación denominado "**Modelo predictivo basado en deep learning para identificar la Mancha Angular (Xanthomonas fragariae) en la hoja de Fresa para el Municipio del cantón Saraguro**".

Traducción que fue guiada y revisada minuciosamente por mi persona. En consecuencia, se da validez a la presentación de la misma. Es todo cuanto puedo certificar en honor a la verdad, pudiendo la interesada hacer uso del presente documento en lo que estimare conveniente.

Atentamente,



Gloria Mercedes Guailas  
Leda. en Ciencias de la Educación mención Idioma Inglés  
Número de Registro Senescyt: 1008-15-1337588  
C.I: 1104980873  
E-mail: [gloria.guailas@educacion.gob.ec](mailto:gloria.guailas@educacion.gob.ec)  
Celular: 0939504329