



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 visión por computador para identificar la enfermedad del Tizón Tardío (*Phytophthora infestans*) en las hojas del tomate de riñón (*Solanum Lycopersicum*) del Cantón Saraguro.

Predictive model based on computer vision to identify late blight disease (*Phytophthora infestans*) on kidney tomato (*Solanum Lycopersicum*) leaves in Cantón Saraguro.

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

AUTORA:

Sisa Ñusta Parra Tene

DIRECTOR:

Ing. Roberth Gustavo Figueroa Díaz, Mg.Sc

Loja – Ecuador

2024

Certificación

Loja, 29 de noviembre de 2024

Ing. Roberth Gustavo Figueroa Díaz, Mg.Sc.

DIRECTOR 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 visión por computador para identificar la enfermedad del Tizón Tardío (*Phytophthora infestans*) en las hojas del tomate de riñón (*Solanum Lycopersicum*) del Cantón Saraguro**, previo a la obtención del título de Ingeniera en Sistemas, de autoría del estudiante **Sisa Ñusta Parra Tene** con **cédula de identidad Nro. 1105194367**, 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. Roberth Gustavo Figueroa Díaz, Mg.Sc.

DIRECTOR DEL TRABAJO DE TITULACIÓN

Autoría

Yo, **Sisa Ñusta Parra Tene**, 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 del Trabajo de Titulación en el Repositorio Digital Institucional – Biblioteca Virtual.

Firma:

Autora: Sisa Ñusta Parra Tene

Cédula: 1105194367

Fecha: 29-11-2024

Correo electrónico: sisa.parra@unl.edu.ec

Teléfono: 0980205467

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

Yo, **Sisa Ñusta Parra Tene**, declaro ser autora del Trabajo de Titulación denominado: “**Modelo predictivo basado en visión por computador para identificar la enfermedad del Tizón Tardío (*Phytophthora infestans*) en las hojas del tomate de riñón (*Solanum Lycopersicum*) del Cantón Saraguro**”; como requisito para optar al grado de: Ingeniera en Sistemas; autorizo al sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la publicación intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Digital 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 autorización, en la ciudad de Loja, a los veintinueve días del mes de noviembre del dos mil veinticuatro.

Firma:

Autor: Sisa Ñusta Parra Tene

Cédula: 1105194367

Dirección: San Lucas

Fecha: 29 de noviembre del 2024

Correo electrónico: sisaparra@unl.edu.ec

Celular: 0980205467

DATOS COMPLEMENTARIOS:

Director del Trabajo de Titulación: Ing. Roberth Gustavo Figueroa Díaz, Mg.Sc.

Dedicatoria

Con gratitud, dedico este trabajo de titulación a Dios, quien ha sido mi refugio y fortaleza en los momentos más desafiantes.

Dedico este trabajo de titulación con mucho amor a mis padres Margarita y Vicente. Ustedes han sido mi inspiración constante, brindándome un amor incondicional y un apoyo esencial en cada paso de mi carrera académica. Su aliento y familiaridad han sido la fuerza que me ha inspirado a alcanzar esta meta. Este logro es tan suyo como mío, ya que han estado a mi lado en cada momento.

A mis adorados hermanos, Yolanda y Pacha, cuya presencia ha llenado mi vida de cariño y alegría. Su amor incondicional ha sido mi refugio en los momentos difíciles, y su apoyo, una constante fuente de inspiración.

Gracias por caminar a mi lado y brindarme siempre esa luz que solo ustedes saben dar.

A mis abuelitas, Luz y Carmen, cuyo amor y sabiduría, su cariño y alegría han sido mi luz.

Que este trabajo sea un testimonio del amor y la gratitud que siento por cada uno de ustedes, que han sido una parte integral de mi viaje.

Sisa Ñusta Parra Tene

Agradecimiento

Primero, agradezco a Dios por ser mi guía y fuente de fortaleza, brindándome la sabiduría necesaria para llegar hasta aquí. A mis padres, les debo todo: su amor incondicional y apoyo moral fueron mi refugio en los momentos difíciles, su fe en mí, mi impulso constante. A mis hermanos, gracias por escucharme siempre, por estar ahí cuando más lo necesitaba. Y a mis queridas abuelitas, su presencia y amor me llenaron de fuerzas en los días más duros. Sin ustedes, este logro no habría sido posible. Su sacrificio y amor me han acompañado en cada paso de este camino académico.

Quiero expresar mis sinceros agradecimientos a mi compañera Diana Gabriela, Tu apoyo constante, amistad y colaboración han sido invaluable a lo largo de este proceso. Gracias por estar siempre dispuesta a ayudar, compartir tus conocimientos y ser una compañera tan solidaria.

Quisiera expresar mi profundo agradecimiento a mi director de trabajo de titulación, el ing. Roberth Gustavo Figueroa Díaz, su paciencia, comprensión y experiencia contribuyeron en el desarrollo de esta investigación, cuya orientación, sabiduría y guía han sido fundamentales en el proceso de exploración y escritura.

A mis amigos y compañeros, quienes han compartido conmigo las alegrías y desafíos de este camino. Sus palabras de aliento y su compañía han hecho más llevaderas las horas de estudio.

Finalmente, agradezco a la Universidad Nacional de Loja por proporcionar el entorno y los recursos necesarios para mi formación académica.

Con profundo agradecimiento,

Sisa Ñusta Parra Tene

Índice de contenido

Portada	i
Certificación	ii
Autoría	iii
Carta de autorización	iv
Dedicatoria	v
Agradecimiento	vi
Índice de contenido	vii
Índice de tablas:.....	xi
Índice de figuras:.....	xiii
Índice de anexos:.....	xvi
1. Título	1
2. Resumen	2
Abstract	3
3. Introducción	4
4. Marco teórico	6
4.1. Antecedentes.....	6
4.1.1. Industria y Agricultura 4.0.....	6
4.2. Cultivo del tomate de riñón (<i>Solanum lycopersicum</i>).....	6
4.2.1. Origen del tomate riñón	6
4.2.2. Cultivo de tomate en Ecuador.....	7
4.2.3. Tizón tardío (<i>Phytophthora infestans</i>) en el tomate de riñón	7
4.3. Fundamentos metodológicos	8
4.3.1. Metodología Design Thinking	8
4.4. Fundamentos teóricos	10

4.4.1.	Inteligencia Artificial (IA)	10
4.4.5.	Aprendizaje automático en visión por computador	14
4.5.	Bibliotecas o Librerías	23
4.5.1.	Tensorflow	23
4.5.2.	Keras	23
4.5.3.	Sklearn	24
4.6.	Arquitecturas de redes convolucionales	25
4.6.1.	VGG16.....	25
4.6.2.	ResNet50.....	26
4.6.3.	InceptionV3.....	27
5.	Metodología.....	32
5.1.	Área de estudio	32
5.2.	Procedimiento.....	32
5.2.1.	Objetivo 1: Identificar métodos, técnicas basadas en inteligencia artificial y recopilar datasets con imágenes de hojas sanas y enfermas del tomate de riñón.....	33
5.2.2.	Objetivo 2: Desarrollar un modelo de visión por computador para la detección automática de la enfermedad Tizón Tardío (<i>Phytophthora infestans</i>) en las hojas del tomate de riñón. 33	
5.2.3.	Objetivo 3: Validar el modelo de visión por computador en un escenario de prueba aplicada a los invernaderos del cantón Saraguro.	33
5.3.	Recursos.....	34
5.3.1.	Recursos científicos	34
5.3.2.	Recursos Técnicos	35
5.4.	Participantes.....	36
6.	Resultados	37

6.1. Objetivo 1: Identificar métodos, técnicas basadas en inteligencia artificial y recopilar Datasets con imágenes de hojas sanas y enfermas del tomate de riñón.	37
Fase 1: Definir	37
6.1.1 Tarea 1: Recolectar imágenes de la hoja de tomate de riñón en estado sano y con tizón tardío.....	37
6.1.2 Tarea 2: Realizar el pre-procesamiento de las imágenes adquiridas para la creación del conjunto de datos que permitirá el entrenamiento del modelo de visión por computador.....	42
6.1.3 Tarea 3: Dividir los datos en entrenamiento, evaluación y prueba.	45
6.2. Objetivo 2: Desarrollar un modelo de visión por computador para la detección automática de la enfermedad Tizón Tardío (<i>Phytophthora infestans</i>) en las hojas del tomate de riñón.	46
Fase 2: Idear	46
6.2.1 Tarea 4: Investigar los modelos Deep learning más utilizados y sus características para el reconocimiento de enfermedades foliares.	46
Fase 3: Prototipar	53
6.2.2 Tarea 5: Entrenar el modelo de visión por computador utilizando los datos de entrenamiento	54
6.3. Objetivo 3: Validar el modelo de visión por computador en un escenario de prueba aplicada a los invernaderos del cantón Saraguro.	68
Fase 4: Evaluar	68
6.3.1 Tarea 6: Validar y contrastar los resultados obtenidos en un escenario de prueba con la opinión de un experto.....	69
7. Discusión	76
7.1 Objetivo 1: Identificar métodos, técnicas basadas en inteligencia artificial y recopilar dataset con imágenes de hojas sanas y enfermas del tomate de riñón.	76
7.2 Objetivo 2: Desarrollar un modelo de visión por computador para la detección automática de la enfermedad Tizón Tardío (<i>Phytophthora infestans</i>) en las hojas del tomate de riñón.	76

7.3 Objetivo 3: Validar el modelo de visión por computador en un escenario de prueba aplicada a los invernaderos del cantón Saraguro.....	78
8. Conclusiones	79
9. Recomendaciones	80
10. Bibliografía	81
11. Anexos	88

Índice de tablas:

Tabla 1. Trabajos relacionados	28
Tabla 2. Herramientas utilizadas para la recolección de datos (imágenes)	38
Tabla 3. Repositorios de hojas de tomate de riñón.....	39
Tabla 4. Descripción de los conjuntos de datos sin procesar	41
Tabla 5. Ejemplo de uso de la aumentación de datos.	44
Tabla 6. Descripción del conjunto de datos propio	44
Tabla 7. Número total del conjunto de datos	45
Tabla 8. Cadenas de búsqueda para la revisión de trabajos relacionados.....	47
Tabla 9. Preguntas de investigación para RSL	48
Tabla 10. Métodos y técnicas basadas en visión por computador	49
Tabla 11. Modelos de aprendizaje automático más efectivos	51
Tabla 12. Especificaciones de entrenamiento	55
Tabla 13. Aplicación de los Modelos.....	61
Tabla 14. Comparación de rendimiento de modelos con él: Dataset1, Dataset2 y Dataset3	62
Tabla 15. Resultados del modelo	65
Tabla 16. Resultados de clasificación: etiquetas reales vs. predicciones de ResNet50	67
Tabla 17. Diagnóstico del Modelo	71
Tabla 18. Resultados de la fase de pruebas en un entorno real.....	75
Tabla 19. Metodología de RSL adaptadas de Barbara Kitchenham	90
Tabla 20. Preguntas de investigación el contexto de la RSL	90
Tabla 21. Criterios de búsqueda para las bibliotecas virtuales	92
Tabla 22. Estudios encontrados en la base de datos científicos	94
Tabla 23. Cantidad de artículos aceptados, rechazados y duplicados del primer análisis	94

Tabla 24. Cantidad de artículos aceptados, rechazados y duplicados del segundo análisis	95
Tabla 25. Cantidad de artículos aceptados, rechazados y duplicados del tercer análisis	95
Tabla 26. Trabajos aprobados (TA) durante la selección de estudios	96
Tabla 27. Métodos, técnicas basadas en Visión por Computador	97
Tabla 28. Modelos de aprendizaje automático son más efectivos para la detección temprana ...	99
Tabla 29. Muestra precisión de diversos modelos de aprendizaje automático	101
Tabla 30. Dataset utilizado para llevar a cabo el entrenamiento del modelo.....	102
Tabla 31. Proceso de adquisición y depuración de imágenes de hojas de tomate de riñón.....	108

Índice de figuras:

Figura 1. Progresión de los síntomas causados por <i>Phytophthora infestans</i>	8
Figura 2. Fases de Design Thinking	9
Figura 3. Arquitectura de una red neurona.....	12
Figura 4. Fases del procesamiento de visión por computador.....	14
Figura 5. Aprendizaje no Supervisado	15
Figura 6. Aprendizaje Supervisado	15
Figura 7. Aprendizaje no supervisado	16
Figura 8. Ejemplo del esquema de la Red Neuronal.....	17
Figura 9. Estructura de la neurona	18
Figura 10. Estructura neurona artificial	18
Figura 11. Demostración de un filtro convolucional	19
Figura 12. Demostración de aplicación de max-pooling.....	20
Figura 13. Demostración de la capa completamente conectada	20
Figura 14. Matriz de confusión.....	21
Figura 15. Arquitectura del modelo VGG16	26
Figura 16. Arquitectura del modelo ResNet50.....	27
Figura 17. Arquitectura del modelo InceptionV3.....	28
Figura 18. Comunidades de Saraguro donde se recolectaron las hojas de tomate.....	32
Figura 19. Fases de la metodología Design Thinking implementados en la investigación.....	33
Figura 20. Cultivos de tomate en el cantón Saraguro	38
Figura 21. Conjunto de datos propio de la hoja de tomate de riñón sana y con tizón tardío.....	39
Figura 22. Conjunto de imágenes de la hoja de tomate de riñón con tizón tardío y sana - Plant Village.....	41

Figura 23. Organización de imágenes privadas en carpetas	42
Figura 24. Script para el aumento de datos en Google Colab	43
Figura 25. Estructura del dataset de hojas enfermas y sanas de tomate de riñón	46
Figura 26. Gráfica de métodos utilizados para la identificación de enfermedades en hojas	49
Figura 27. Modelos seleccionados para ser entrenados y evaluados	53
Figura 28. Hiperparámetros de entrenamiento	56
Figura 29. Especificación de rutas al conjunto de datos	56
Figura 30. Generador de imágenes (entrenamiento y validación)	57
Figura 31. Construcción y Configuración del Modelo VGG16.....	57
Figura 32. Estructura del modelo	58
Figura 33. Construcción y Configuración del Modelo InceptionV3	59
Figura 34. Construcción y Configuración del Modelo ResNet50	59
Figura 35. Método Transfer Learning	60
Figura 36. Técnica Feature Extraction	60
Figura 37. Gráfica que ilustra los resultados de presión	63
Figura 38. Gráfica que ilustra los resultados de pérdida	63
Figura 39. Arquitectura de Clasificación de Hojas de Tomate.....	64
Figura 40. Comparación visual de imágenes del modelo ResNet50	66
Figura 41. Curvas de precisión y pérdida durante el entrenamiento del modelo	66
Figura 42. Matriz con los datos de test	68
Figura 43. Pantalla acceso a la aplicación	69
Figura 44. Visualización de diagnósticos desde la aplicación	70
Figura 45. Enfoque de RSL de Kitchenham	88
Figura 46. Zona de Estudio 1- de la Parroquia Urdaneta	109

Figura 47. Zona de Estudio 2- Parroquia de San Pablo de Tenta	110
Figura 48. Zona de Estudio 3- comunidad de Tuncarta	110
Figura 49. Recolección de hojas de tomate de riñón en Urdaneta	110
Figura 50. Recolección de hojas de tomate de riñón Tenta	111
Figura 51. Recolección de hojas de tomate de riñón Tuncarta	111
Figura 52. Organización para la Captura de Imágenes	112
Figura 53. Clasificación de imagen con la ayuda del experto	112

Índice de anexos:

Anexo 1. Revisión Sistemática de Literatura	88
Anexo 2. Recolección de datos para la clasificación de tizón tardío en hojas de tomate de riñón	107
Anexo 3. Manual de Usuario.....	113
Anexo 4. Certificado de validación del conjunto de datos por el Técnico de Apoyo Agropecuario del Gobierno Autónomo Descentralizado del Cantón Saraguro	118
Anexo 5. Certificación de Evaluación del prototipo Web.....	119
Anexo 6. Certificado de traducción del resumen al inglés	120

1. Título

Modelo predictivo basado en visión por computador para identificar la enfermedad del Tizón Tardío (*Phytophthora infestans*) en las hojas del tomate de riñón (*Solanum Lycopersicum*) del Cantón Saraguro.

Predictive model based on computer vision to identify late blight disease (*Phytophthora infestans*) on kidney tomato (*Solanum Lycopersicum*) leaves in Cantón Saraguro.

2. Resumen

En el cantón Saraguro, muchos agricultores se dedican a la producción de tomate de riñón; sin embargo, esta actividad se ha visto afectada por enfermedades como el tizón tardío, que ocasiona pérdidas económicas significativas. Este trabajo de titulación tiene como objetivo aplicar tecnologías basadas en inteligencia artificial, específicamente redes neuronales convolucionales, para identificar dicha enfermedad en las hojas de tomate de riñón, facilitando así una detección oportuna y una gestión más efectiva del cultivo.

La metodología empleada se basó en el enfoque de Design Thinking, implementando las fases de Definir, Idear, Prototipar y Evaluar. En la primera fase, se recolectaron fotografías de hojas de tomate de riñón sanas y con tizón tardío en distintos invernaderos del cantón Saraguro, logrando un total de 1,447 imágenes, las cuales se combinaron con 3,351 imágenes del set de datos de PlantVillage. En la segunda fase, se investigaron los modelos más efectivos para la clasificación de enfermedades en hojas de plantas, destacándose los modelos ResNet50, VGG16 e InceptionV3. Esto permitió avanzar a la tercera fase, donde se entrenaron estos modelos, obteniendo precisiones del 99.8%, 99.3% y 98.5%, respectivamente.

En la fase final, se evaluó el modelo ResNet50 mediante una aplicación web que integró el modelo entrenado. Se recolectaron 100 imágenes de hojas de tomate de riñón, sanas y enfermas, en un invernadero, las cuales se subieron a la aplicación para su diagnóstico. Los resultados se compararon con la evaluación de un agrónomo del cantón Saraguro, logrando una precisión del 91% en la clasificación de la enfermedad.

Palabras: CNN, VGG16, ResNet50, InceptionV3, Deep Learning.

Abstract

In Saraguro canton, many farmers are engaged in the production of kidney tomato; however, this activity has been affected by diseases such as late blight, which causes significant economic losses. This degree work aims to apply technologies based on artificial intelligence, specifically convolutional neural networks, to identify this disease in kidney tomato leaves, thus facilitating timely detection and more effective crop management.

The methodology used was based on the Design Thinking approach, implementing the Define, Ideate, Prototype and Evaluate phases. In the first phase, photographs of healthy and late blight kidney tomato leaves were collected in different greenhouses in the Saraguro canton, achieving a total of 1,447 images, which were combined with 3,351 images from the PlantVillage dataset. In the second phase, the most effective models for the classification of plant leaf diseases were investigated, highlighting the ResNet50, VGG16 and InceptionV3 models. This allowed advancing to the third phase, where these models were trained, obtaining accuracies of 99.8%, 99.3% and 98.5%, respectively.

In the final phase, the ResNet50 model was evaluated using a web application that integrated the trained model. One hundred images of healthy and diseased kidney tomato leaves were collected in a greenhouse and uploaded to the application for diagnosis. The results were compared with the evaluation of an agronomist from Saraguro canton, achieving an accuracy of 91% in the classification of the disease.

Keywords: CNN, VGG16, ResNet50, InceptionV3, Deep Learning.

3. Introducción

El cultivo de tomate riñón es una actividad crucial en el Cantón Saraguro, especialmente en comunidades rurales, donde la producción se realiza mayoritariamente en invernaderos. Reconociendo la importancia de este sector y los desafíos que enfrentan los pequeños agricultores, el Gobierno Municipal ha puesto en marcha diversos proyectos de apoyo encaminados al beneficio del agricultor [1]. Sin embargo, persiste una problemática importante: la falta de conocimientos para identificar y controlar enfermedades que afectan a los cultivos, siendo el tizón tardío una de las más devastadoras.

El tizón tardío, causado por el hongo *Phytophthora infestans*, es reconocido como una de las enfermedades más destructivas para el cultivo de tomate [2]. Sus síntomas incluyen manchas verde-negras y grasosas en el envés de las hojas, que pueden propagarse rápidamente, causando estragos en los cultivos en cuestión de días. La capacidad de esta enfermedad para propagarse rápidamente a hojas y frutos, subraya la importancia de su detección temprana como un factor crucial para la aplicación de medidas correctivas que minimicen las pérdidas en la producción.

En este contexto, el proyecto busca implementar un modelo predictivo que permita detectar la enfermedad a tiempo, utilizando tecnologías avanzadas como la inteligencia artificial. Este enfoque innovador busca proporcionar a los agricultores de Saraguro una herramienta fácil de usar, que les ayude a identificar el tizón tardío en sus cultivos, permitiéndoles tomar medidas inmediatas para frenar su propagación y minimizar los daños.

Para el desarrollo del modelo, se crearon tres conjuntos de datos. El primero consistió en 3351 imágenes de hojas de tomate riñón sanas y afectadas por el tizón tardío, obtenidas del repositorio en línea PlantVillage. El segundo conjunto incluyó 4094 imágenes capturadas en invernaderos del cantón Saraguro, recolectadas durante trabajos de campo realizados por la autora. Finalmente, se formó un tercer conjunto combinando ambos sets de imágenes, con el objetivo de enriquecer la diversidad de muestras.

Se entrenaron y validaron tres modelos de redes neuronales convolucionales (CNN): ResNet50, InceptionV3 y VGG16, seleccionados tras una revisión exhaustiva de la literatura. Todos los modelos fueron entrenados bajo las mismas condiciones para permitir una comparación equitativa de su rendimiento en los distintos conjuntos de datos.

Luego de la evaluación, se seleccionó el modelo con mejor desempeño, que fue sometido a un nuevo ciclo de entrenamiento ajustando el tamaño del batch size para optimizar su rendimiento.

Para validar el modelo en condiciones reales, se desarrolló un prototipo web que facilita la interacción con el sistema de manera accesible. En la fase de pruebas, se recolectaron 100 imágenes de hojas en un invernadero del cantón Saraguro, las cuales fueron procesadas por el modelo. Los diagnósticos generados se compararon con los emitidos por un experto en el campo, obteniéndose resultados positivos que confirmaron la efectividad del modelo en un entorno real.

Este enfoque no solo permitió validar el sistema en un entorno controlado, sino que también facilitó su aplicación en escenarios prácticos, ofreciendo una solución tecnológica útil para los agricultores de la región.

4. Marco teórico

En este apartado se presentan los fundamentos teóricos que respaldan el Trabajo de Titulación (TT), basados en una revisión exhaustiva de la literatura existente.

4.1. Antecedentes

4.1.1. Industria y Agricultura 4.0

La Industria 4.0, asimismo conocida como la cuarta revolución industrial, es un concepto que surgió en Alemania en 2011 como parte del plan estratégico "High Technology Strategy for Germany 2020". La Industria 4.0 se basa en la integración de diversas tecnologías digitales, entre ellas el Internet de las Cosas (IoT). La Industria 4.0 tiene como objetivo de integrar varias tecnologías digitales, incluyendo el Internet de las Cosas (IoT), para mejorar la eficiencia en la producción, personalizar productos, optimizar la cadena de suministro y responder de manera rápidamente [3]. Esta iniciativa, que comenzó en Alemania, se ha convertido en un movimiento global que está transformando la forma en que operan las industrias en todo el mundo [4].

Por otro lado, la Agricultura 4.0 se busca desarrollar herramientas para optimizar los procesos agrícolas. Actualmente, el concepto incluye el uso de tecnologías digitales e inteligencia artificial, donde representa una revolución tecnológica en el sector agrícola [5]. Se basa en la aplicación de tecnologías digitales y de información, como sensores, drones, GPS y análisis de datos, para optimizar todos los aspectos de la producción agrícola [6]. Esta transformación busca aumentar la productividad, reducir el consumo de recursos como el agua y los fertilizantes, y mejorar la sostenibilidad ambiental. Esta nueva era agrícola no solo se enfoca en la eficiencia, sino también en la calidad de los cultivos. La Agricultura 4.0 permite un monitoreo preciso de los cultivos y el uso de sistemas de riego inteligentes, lo que se traduce en cosechas de mayor calidad [7]. Tanto la Industria 4.0 como la Agricultura 4.0 se centran en la implementación de tecnologías avanzadas para cambiar y optimar los sectores industriales y agrícolas tradicionales, con el objetivo de lograr una mayor eficiencia, sostenibilidad y competitividad [8].

4.2. Cultivo del tomate de riñón (*Solanum lycopersicum*)

4.2.1. Origen del tomate riñón

El tomate, de nombre científico conocido como *Solanum lycopersicum*, se cree que el origen del género está en la región de los Andes, que se extiende desde el sur de Colombia hasta el

norte de Chile, incluidas las Islas Galápagos y la región de los Andes orientales. Esta hipótesis se basa en la presencia de varias especies silvestres endémicas en estas zonas de América del Sur, incluyendo Colombia, Ecuador, Perú, Bolivia y Chile [9]. Es un cultivo de gran relevancia. Además, el tomate es una parte esencial de la dieta y la gastronomía, utilizado en una variedad de platos tradicionales y contemporáneos.

4.2.2. Cultivo de tomate en Ecuador

En el contexto agrícola de Ecuador, el cultivo de tomate en el sector agrícola no solo ayuda a alimentar a la población, sino que además es una fuente clave de ingresos para los agricultores en diversas regiones [10]. Además, la producción de tomate es una actividad económica significativa en la cadena de suministro de alimentos y tiene un impacto en la economía local y nacional [11]. En Ecuador, el tomate es un cultivo importante y de gran demanda en la industria alimentaria. Cada año, se producen aproximadamente 62 mil toneladas de tomates, la mayoría se localiza en la provincia de Santa Elena y Azuay, Imbabura y Carchi, donde las condiciones son ideales para su cultivo [12].

El desarrollo productivo en Saraguro se ha visto favorecido gracias al establecimiento de Escuelas de Campo (ECAS) en el Municipio de Loja. Estas iniciativas se centran en capacitar y brindar asesoría técnica a los productores locales, especialmente a aquellos que cultivan tomate riñón en invernadero. En varias comunidades de Saraguro, incluidas Lluzhapa, Tuncarta y Tambopamba, se están estableciendo ECAS para mejorar las prácticas agrícolas y optimizar los resultados de la producción de tomate riñón [13].

4.2.3. Tizón tardío (*Phytophthora infestans*) en el tomate de riñón

Phytophthora infestans, el agente causal del tizón tardío, es un patógeno extremadamente destructivo perteneciente a la clase Oomycetes [14]. Esta enfermedad puede provocar pérdidas significativas e incluso la destrucción total de cultivos de tomate de riñón. Esta enfermedad puede atacar a la planta de tomate en cualquier etapa de su crecimiento. Los síntomas comienzan manifestándose como áreas necrosadas en las hojas, rodeadas de un fieltro blanco, a la par de la aparición de micelio. Las hojas terminan muriendo rápidamente, pero la enfermedad termina extendiéndose a demás partes de la planta. En los frutos se observa zonas color marrón, pero firmes [15].

4.2.3.1. Síntomas

Como se puede observar en la Figura 1 tomadas de [16], el *Phytophthora infestans*, al principio inicia apareciendo pequeñas manchas de color verde oscuras en las hojas y se desarrollan de manera rápida. A medida que la enfermedad va avanzando provoca el marchitamiento, consecutivamente cambia de color de forma amarillento, esta enfermedad tiende a propagarse de manera rápida en el ambiente con alta humedad superior al 90 % acompañada de temperaturas de 10 a 25 °C, noches frías y días moderadamente cálidos, a través de la lluvia, el viento, goteras en los invernaderos, el hongo se propaga de manera más rápida. En etapas más severas, el tizón tardío impacta el tallo, provocando la caída de hojas y frutos, acompañado de la formación de manchas marrones o negras, reblandecimiento y descomposición. Además, la enfermedad tiene el potencial de extenderse a las raíces, provocando pudrición y comprometiendo la resistencia del sistema radicular la propagación del hongo se multiplica rápidamente y es capaz de atacar a numerosas plantas [17].



Figura 1. Progresión de los síntomas causados por *Phytophthora infestans*

4.3. Fundamentos metodológicos

En esta sección, se comparten los principales fundamentos metodológicos para el desarrollo de este trabajo de titulación.

4.3.1. Metodología Design Thinking

Es un guía funcional que utiliza herramientas del diseño y la creatividad generando ideas innovadoras permitiendo resolver los requerimientos del cliente. En [18] se describe su naturaleza

centrada en el usuario y su enfoque iterativo ayudando en cada etapa del proceso. En este proceso, las necesidades humanas siempre ocupan un lugar central. Identificar y analizar necesidades y tendencias, proponer soluciones, probar prototipos e integrar conocimientos de diversas disciplinas como psicología, sociología, marketing e ingeniería. En términos generales, el diseño consta de cinco etapas principales, como se puede observar en la Figura 2, tomada de [18].

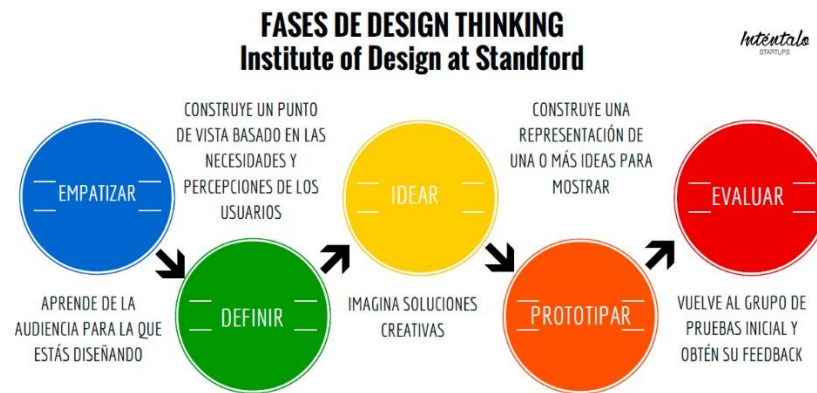


Figura 2. Fases de Design Thinking

A continuación, se describen los pasos clave en cada etapa de Design Thinking según [19]:

1. **Empatizar:** En esta etapa, busca comprender entender al usuario y sus necesidades a través de la observación y la interacción directa. La observación activa consiste en identificar lo que verdaderamente es importante para el cliente, desplegando una empatía que resulta fundamental para captar sus verdaderas necesidades y expectativas.
2. **Definir:** Una vez obtenida la información en la etapa de la empatía, se pasa a clara y concretar el problema a resolver. Esto implica sintetizar la información para que el problema tenga sentido para el cliente, para de esta manera facilitar una solución factible.
3. **Idear:** Con el problema claramente definido, se pasa a la fase de generación de ideas. Aquí se fomenta la creatividad mediante sesiones de lluvia de ideas, donde se generan múltiples soluciones potenciales, sin importar si las ideas están relacionadas entre sí o si difieren de la idea inicial. En esta fase, se analizaron diversos documentos relacionados con el tema de estudio, lo que permitió recopilar

ideas y posibles soluciones para orientar el desarrollo del proyecto en este documento.

4. **Prototipar:** En esta fase consiste en convertir la idea en un modelo tangible. Se crea un prototipo del producto o servicio diseñado para que la solución pueda desarrollarse y probarse con el público. La creación de prototipos es crucial porque permite identificar y corregir errores antes de la implementación final, ahorrando tiempo y recursos.
5. **Evaluar/Probar:** En la fase final, también llamada prueba, el prototipo es evaluado con usuarios existentes. En este período, observa cómo los interesados interactúan con el producto o servicio, de esta manera recibiendo comentarios directos. Esta información es invaluable para iterar y mejorar la solución antes de su implementación final, de esta manera asegurando que los resultados finales satisfagan las expectativas y necesidades del usuario.

4.4. Fundamentos teóricos

En esta sección se muestran las principales bases teóricas para el desarrollo de este trabajo de titulación.

4.4.1. Inteligencia Artificial (IA)

La inteligencia artificial (IA) es un campo interdisciplinario de la informática que se enfoca en crear sistemas que normalmente es realizado por un humano, se caracterizan principalmente por desarrollar algoritmos capaces de llevar a cabo tareas que al ser realizados por el ser humano necesite de inteligencia. Es decir que busca desvalorar máquinas y programas que permitan imitar procesos cognitivos humanos, en este caso el aprendizaje, percepción, razonamiento y la toma de decisiones [20].

La IA se basa en la idea de que las máquinas pueden ser programadas para realizar tareas complejas mediante el uso de algoritmos y modelos matemáticos. Estos modelos pueden ser entrenados para aprender de datos y mejorar su desempeño a lo largo del tiempo, lo que se conoce como aprendizaje automático [21]. Los sistemas de IA pueden clasificarse en dos categorías principales: IA débil y IA fuerte. La IA débil se refiere a sistemas diseñados para realizar tareas específicas, como reconocimiento de voz o diagnóstico médico, sin poseer conciencia o

comprensión. La IA fuerte, por otro lado, implica la creación de sistemas que tienen la capacidad de razonar y comprender de manera similar a un ser humano [22].

En el contexto de este trabajo de titulación, la inteligencia artificial desempeña un papel crucial al permitir la detección automática de enfermedades en las hojas de tomate de riñón mediante modelos de visión por computado basados en redes neuronales convolucionales (CNN). Estos modelos utilizan algoritmos de aprendizaje profundo para analizar imágenes y realizar diagnósticos precisos de las condiciones de las hojas.

4.4.2. Machine Learning

El aprendizaje automático, por su significado en inglés *Machine learning*, es una rama de la Inteligencia Artificial (IA) que se encuadra en el aprendizaje de la computadora que permite realizar acciones si la participación humana [23], este aprendizaje automático permite a los ordenadores la posibilidad de aprender sin ser explícitamente programados. Este aprendizaje se caracteriza por desarrollar programas informáticos que están en constantes cambios cuando ingresan nuevos datos, ya que los datos son dinámicos, están generando actualizaciones cada cierto tiempo, lo que se quiere es tener un valor muy cercano a la realidad, puesto que estos datos se usan para realizar el análisis respectivo dependiendo el área que se quiera usar.

Al hablar de aprendizaje automático va de la mano con la minería de datos. Estos sistemas buscan entre los datos encontrar patrones de comportamiento. La información para la minería de datos consiste en extraer datos para la interpretación humana, mientras que el aprendizaje automático utiliza los datos para identificar patrones de comportamiento, es decir, realiza un seguimiento de estos para ajustar las acciones de cierto programa en consecuencia. Existen dos tipos de algoritmos aprendizaje automático, los supervisados y los no supervisados, los algoritmos supervisados pueden emplear los datos aprendidos y pasarlo a nuevos datos, por otro lado, los algoritmos no supervisados tienen como finalidad en un conjunto eliminar distintas inferencias.

Hoy en día las aplicaciones utilizan el aprendizaje automático para la resolución de problemas. En [24], como es el caso de aplicaciones de la salud, en donde el aprendizaje automático se comporta como un estudiante que inicia su movimiento hospitalario, donde la atención para un nuevo paciente le permite conocer las características y la combinación para pronosticar un resultado de manera confiable.

4.4.3. Deep Learning (aprendizaje profundo)

El aprendizaje automático profundo, la cual utiliza redes neuronales artificiales que sustenta la gran mayoría de los modelos de aprendizaje profundo. El resultado que se obtiene del aprendizaje profundo es denominado a veces aprendizaje neuronal profundo o redes neuronales profundas [25].

Las redes neuronales constituyen un paradigma informático de procesamiento de información, el cual está inspirado en los sistemas neuronales biológicos, donde la neurona biológica es considerada como un elemento de almacenamiento de datos y forma un conjunto de elementos específicos de procesamiento que son denominados nodos o neuronas que están conectadas entre sí, como se observa en Figura 3 tomadas de [26].

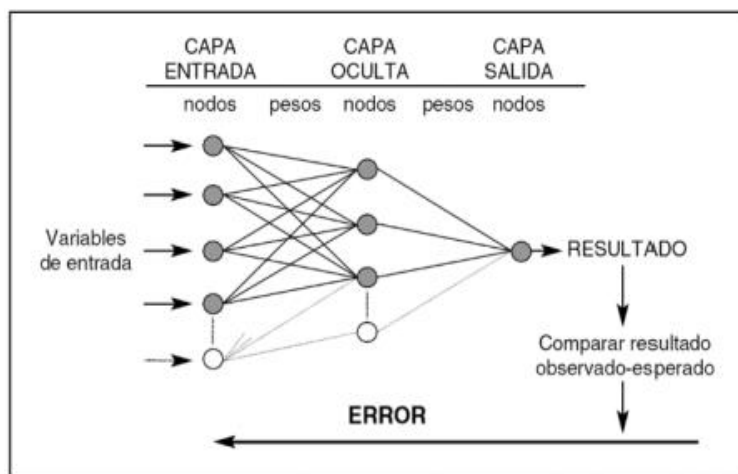


Figura 3. Arquitectura de una red neuronal

Las redes neuronales imitan el funcionamiento del cerebro humano, compuesto por millones de células llamadas neuronas [25]. Estas neuronas procesan la información mediante un conducto de entrada de información, un procesamiento y un canal de salida de información. La red neuronal es un algoritmo que calcula y se basa en una analogía del sistema nervioso, cuyo proceso consiste en simular la capacidad de aprendizaje de un sistema nervioso, de modo que la red neuronal (RN) asimile la identificación de un patrón de agrupación entre los datos de variables predictoras conocidas entradas y los estados dependientes de los valores conocidos como salidas [27].

Las redes neuronales artificiales muestran características parecidas a las del cerebro, por lo que aprenden la rutina, obtienen casos anteriores y trasladan a nuevos casos, además extrae las características específicas partiendo de las entradas que representan información irrelevante, que

ofrece ventajas y además este tipo de tecnología se aplica en múltiples áreas de predicción de información. Las redes neuronales se han ido perfeccionando con el paso de tiempo hoy en día, en los sistemas inteligentes que usan machine learning, implementan técnicas muy avanzadas para la resolución de problemas, que están basados en modelos predictivos.

4.4.4. Visión por computador

La visión por computador es una rama de la inteligencia artificial, pero también de la informática, que se centra en dotar a las máquinas de la capacidad de interpretar de manera de forma semejante a como lo realiza una persona. Esta disciplina abarca de algoritmos y técnicas que permiten a las computadoras realizar análisis, procesar y extraer información significativa de datos visuales. La visión por computador encuentra aplicaciones en una amplia gama de campos, desde el reconocimiento facial hasta la detección de objetos en imágenes médicas [28].

- Visión por computador en la agricultura

La visión por computador en inteligencia artificial (IA) implica el uso de algoritmos y herramientas que permiten a las máquinas comprender y analizar imágenes o videos, recopilando funciones visuales humanas. En el ámbito agrícola, esta tecnología ha cobrado especial importancia, ya que ofrece soluciones para mejorar la eficiencia y calidad en la producción de alimentos. Al analizar imágenes capturadas por drones, satélites o cámaras en los campos, es posible identificar patrones, detectar problemas y tomar decisiones fundamentadas sobre el estado de los cultivos, ayudando a los agricultores a optimizar sus prácticas [29].

Las principales aplicaciones de la visión por computador en la agricultura incluyen:

- **Reconocimiento de patrones:** permite identificar características únicas de imágenes que ayudan a diferenciar entre plantas sanas y aquellas afectadas por enfermedades. Esto es fundamental para la detección de problemas de cultivos y mejores decisiones de gestión agrícola [30].
- **Seguimiento de objetos:** se utiliza para monitorear el crecimiento y desarrollo de los cultivos en tiempo real, ayudando a detectar cambios en las condiciones de las plantas, como la salud o la tasa de crecimiento, para optimizar el rendimiento agrícola [30].
- **Segmentación de Imágenes:** Proceso de dividir una imagen en componentes u objetos de interés, permitiendo un análisis más preciso del estado de cada planta o fruto y priorizando áreas que requieren atención inmediata [30].

- Fases del procedimiento de visión artificial.

El proceso de visión por computador está estructurado por tres partes principales que se destacan a continuación: primero es la adquisición de la imagen, en esta parte las fotos son tomadas mediante camaras o sensores, segundo es procesamiento de imagen, que incluye la segmentación, que se encarga de dividir la imagen capturada en regiones o partes significativas, con este proceso facilita la extracción de las características más relevantes, permitiendo que la imagen se analice de manera más detallada, en esta última fase de interpretación, se trata de extraer las características relevantes y la toma de decisiones basada en el análisis previo, para de esta manera busca imitar la capacidad humana de comprender el entorno visual, los pasos que llevan se puede visualizar en la Figura 4 la cual fue tomada de [31].

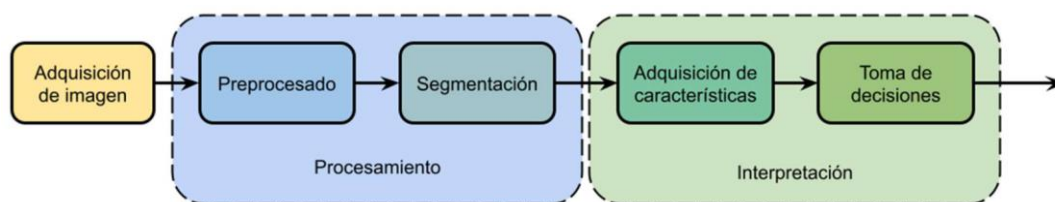


Figura 4. Fases del procesamiento de visión por computador

4.4.5. Aprendizaje automático en visión por computador

El aprendizaje automático ha transformado la visión por computador al proporcionar herramientas para analizar e interpretar datos visuales sin programación específica. En este campo destacan dos métodos principales: el aprendizaje supervisado y el aprendizaje no supervisado [32]. El aprendizaje supervisado utiliza datos etiquetados para entrenar un modelo que puede predecir o clasificar nuevas instancias, mientras que el aprendizaje no supervisado utiliza datos no etiquetados para identificar patrones y estructuras subyacentes por sí solo. Ambos métodos son fundamentales para la clasificación y detección de objetos en imágenes [33].

4.4.5.1. Aprendizaje Supervisado

Los algoritmos que manejan aprendizaje supervisado tienden a trabajar con información, la etiqueta es una de las salidas que se muestra de inicio en el conjunto de datos ya conocidos. Cuando se usan los datos que se tiene, se diseña un modelo de predicción que permitirá saber la clase con la que se puede crear nuevos objetos que aún no se han visualizado. Entre los algoritmos más

utilizados se encuentran Naive Bayes, Vecinos más cercanos y Máquinas de vectores de soporte [34].

En el aprendizaje supervisado se tiene la siguiente clasificación:

- a. Aprendizaje Supervisado por regresión, como se puede ver en la Figura 5 fue tomada de [35] [36]; en la que se muestra la regresión es un aprendizaje, que va a predecir un valor casi cercano a la realidad basado en las entradas; es así como también se muestran algoritmos que tienen para la clasificación por regresión.

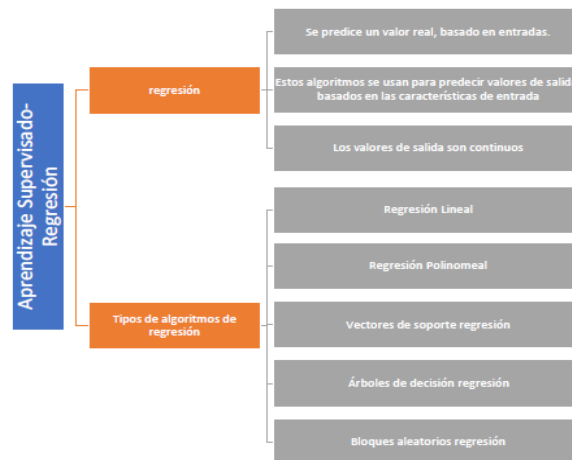


Figura 5. Aprendizaje no Supervisado

- b. Aprendizaje por clasificación, como se muestra en la Figura 5 y Figura 6 basadas en [36] sostiene que es un algoritmo que intenta etiquetar cada uno de los ejemplos eligiendo entre dos o más clases diferentes.

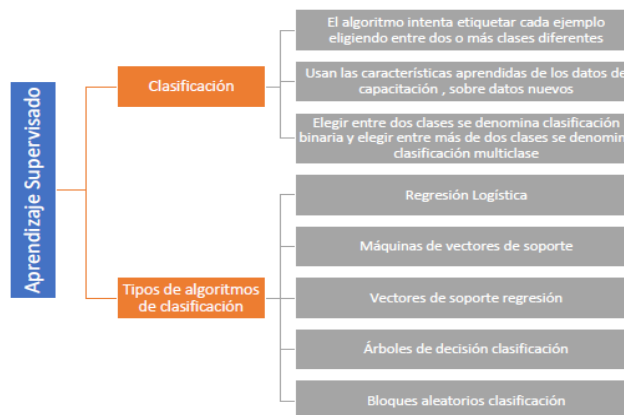


Figura 6. Aprendizaje Supervisado

4.4.5.2. Aprendizaje No Supervisado

Utiliza un conjunto de datos, los cuales no disponen de etiqueta alguna. La idea es que el algoritmo pueda encontrar patrones por sí solo para poder entender el conjunto de datos. Este aprendizaje se utiliza, generalmente, en problemas de *clustering* agrupamientos y de concurrencia.

Las características más habituales de algoritmos son:

- a) Algoritmos de *clustering*
- b) Análisis de componentes principales
- c) Descomposición en valores singulares
- d) Análisis de componentes principales

Se puede hablar del análisis de aprendizaje no supervisado, que menciona [37], tiene que ser un proceso, el cual se logra llevar a cabo con los pasos expuestos en la Figura 7 tomada de [38], donde se puede ver, que establecen pasos ordenadamente, esto se inicializa por los datos, que se realiza una selección de sus principales características para pasar al modelamiento de un algoritmo de agrupación, una vez que se ha realizado este proceso para hacer la validación de agrupación para continuar con la interpretación de resultado y por último la verificación del algoritmo de agrupación.

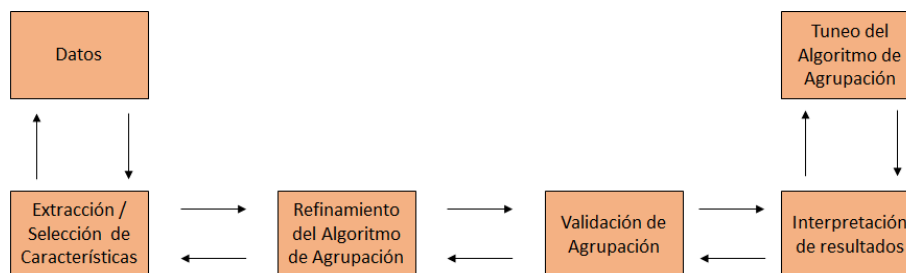


Figura 7. Aprendizaje no supervisado

4.4.6. Redes Neuronales

Las redes neuronales son sistemas computacionales diseñados tomando como referencia la estructura y el funcionamiento del cerebro humano. Están formadas por nodos interconectados, las cuales están compuestas por nodos conectados entre sí, conocidos como neuronas artificiales, que se encargan de procesar y enviar información. Estas redes pueden aprender patrones y relaciones complejas en los datos mediante el ajuste de los pesos en sus conexiones.

Las redes neuronales son capaces de aprender patrones y relaciones complejas a partir de datos a través del ajuste de sus conexiones ponderadas. En el contexto de la visión por computador, las redes neuronales juegan un papel crucial en la detección y clasificación de objetos en imágenes. La temática consultada señala que "las redes neuronales son la columna vertebral de muchos algoritmos de visión por computador y aprendizaje profundo" [39].

Por otra parte, la red neuronal se compone de neuronas conectadas, tiende a estar organizadas en múltiples capas, los datos que ingresan a través de la capa de ingreso, posterior a esto pasan por una o varias ocultas y finalmente se comunican con la capa de salida. Teniendo en cuenta que las capas ocultas están formadas por varias subcapas, cada capa está encargada de enviar y transformar, para refinar la información antes de que tenga la salida, en la Figura 8 extraída de [40] se puede visualizar como es la estructura.

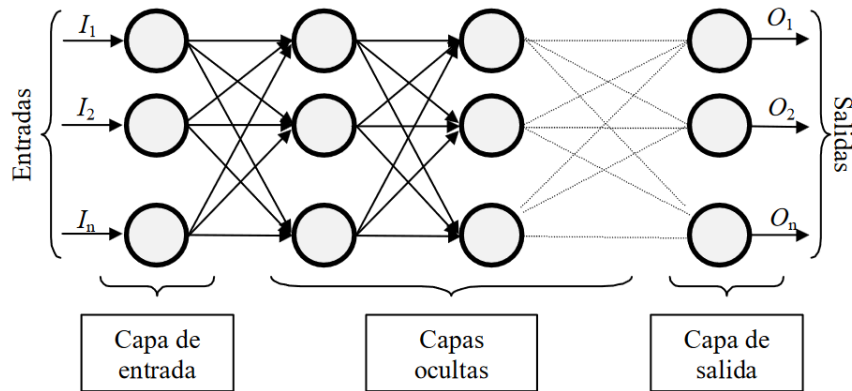


Figura 8. Ejemplo del esquema de la Red Neuronal

4.4.7. Redes Neuronales Convolucionales

Las Redes Neuronales Convolucionales (CNN) son una clase de redes neuronales especialmente diseñadas para el procesamiento de imágenes. Estas redes imitan el proceso visual en los seres humanos al aplicar capas de convolución y pooling para extraer características de las imágenes, como bordes, texturas y formas. Las CNN son conocidas por su capacidad para aprender jerarquías de características, lo que les permite identificar patrones complejos en imágenes. Las Redes Neuronales Convolucionales han corroborado lograr un gran éxito en operaciones de clasificación y detección de objetos en visión por computador. Por otra parte, también las redes neuronales están basadas en la inteligencia artificial (IA), es desarrollado con el objetivo de lograr imitar las funciones elementales que tiene el cerebro humano, para lograr la adquisición del

conocimiento. Esta arquitectura es en capas, las cuales están unidas en gran parte de números por unidades que se llaman neuronas, siendo una estructura básica [41].

Un proceso similar ocurre en nuestro cerebro, ya que cada una de las neuronas están interrelacionadas una con otra, con el fin de enviar y recibir conocimiento, para ellos tiene tres partes fundamentales que son:

- **Dendritas:** es el encargado de emitir los impulsos nerviosos por las neuronas.
- **Soma:** es el encargado de procesar los impulsos eléctricos
- **Axón:** envía impulsos nerviosos hacia neuronas contiguas

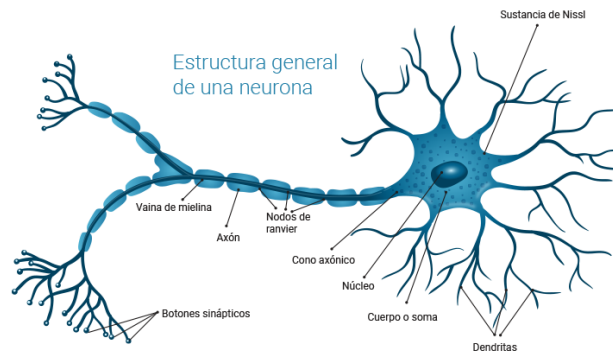


Figura 9. Estructura de la neurona

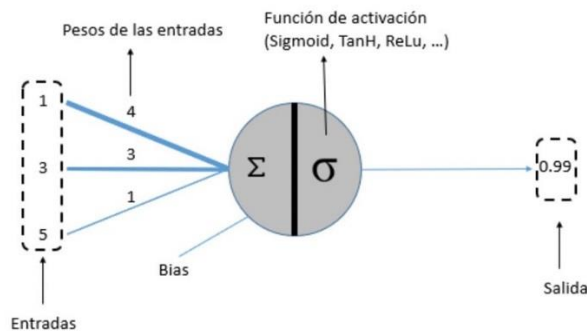


Figura 10. Estructura neurona artificial

Figura 9 y Figura 10 tomados de [42] se observa los enlaces son producidos inicialmente en la entrada que es donde se reciben los valores que son resultado de una suma o multiplicación, el cual se transmite hacia dentro de la red neuronal, donde está una estructura compleja en forma de red, cuyos enlaces son los pesos de las conexiones.

4.4.7.1. Capa convolucional

La capa convolucional en una red neuronal convolucional (CNN) es una estructura que realiza la convolución entre los datos de entrada y un conjunto de filtros de convolución. Esta capa es fundamental para la extracción de características relevantes en imágenes y datos multidimensionales. La convolución implica la aplicación de estos filtros en regiones locales de los datos de entrada, lo que permite detectar patrones y características específicas en la información. También es importante la capa, ya que emplea un filtro o kernel a la imagen de entrada para de esta manera generar un mapeo, el filtro lo realiza en la imagen, calculando efectos y sumas para destacar patrones importantes de la imagen original, diferenciando así a las CNN de otras redes neuronales [43], en la Figura 11 se puede apreciar un ejemplo de cómo es un filtro convolucional, tomada de [43].

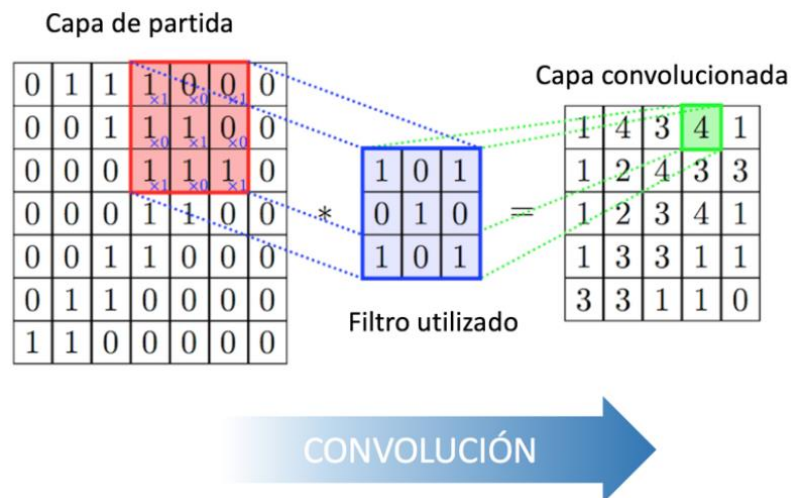


Figura 11. Demostración de un filtro convolucional

4.4.7.2. Capa de pooling

La capa de pooling en una CNN tiene como objetivo reducir la dimensionalidad de los mapas de características generados por las capas convolucionales previas. Esta capa logra esto al tomar regiones de los mapas de características y resumirlas en un solo valor, generalmente utilizando operaciones como el máximo o el promedio. El pooling disminuye la cantidad de parámetros en la red, lo que ayuda a controlar el sobreajuste y a mejorar la eficiencia computacional. En esta capa existen tres formas para la aplicación de pooling: primero está el max-pooling, en esta etapa devuelve el máximo valor de entre todos los píxeles cubiertos por el kernel, segundo está el min-pooling: en esta fase devuelve el valor mínimo de entre todos los píxeles cubiertos por el kernel, finalmente está el average pooling: se calcula el valor medio de entre todos

los píxeles cubiertos por el kernel [43], en la Figura 12, extraída de [43] se puede apreciar un ejemplo de max-pooling

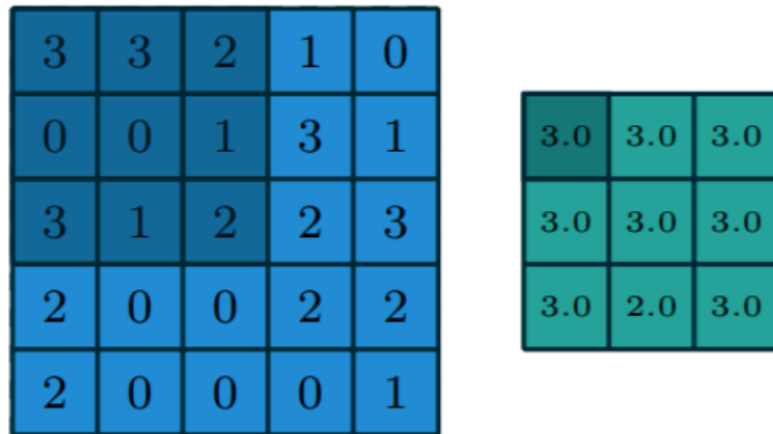


Figura 12. Demostración de aplicación de max-pooling

4.4.7.3. Capa de clasificación (capa completamente conectada)

La capa de clasificación, también conocida como capa completamente conectada, es la parte final de una CNN y se encarga de tomar las características extraídas por las capas anteriores y realizar la clasificación final de los objetos en las imágenes. Esta capa consiste en neuronas conectadas a todas las neuronas de la capa anterior, lo que permite considerar las relaciones entre las características extraídas para tomar decisiones de clasificación [44], en la Figura 13, derivada de [44] se puede visualizar un ejemplo de cómo están conectadas.

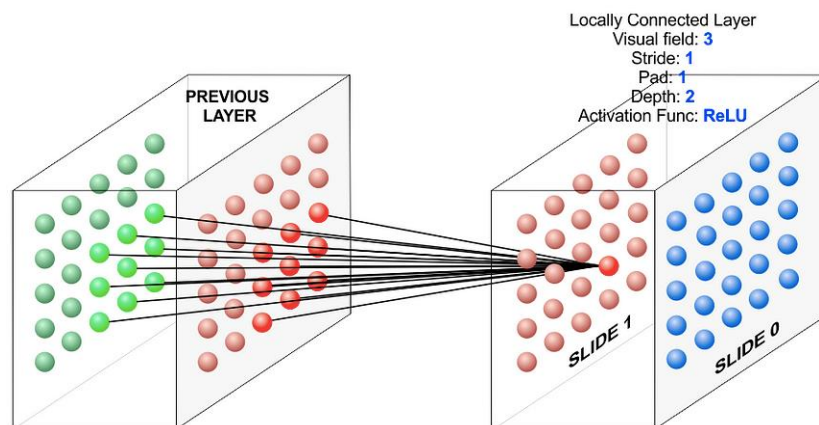


Figura 13. Demostración de la capa completamente conectada

4.4.8. Matriz de confusión

La matriz de confusión es una herramienta fundamental en el campo del aprendizaje automático y la visión por computador. La misma es utilizada para evaluar el rendimiento de un modelo clasificador, como las redes neuronales convolucionales (CNN). La matriz de confusión es una herramienta que no solo sirve para evaluar y ajustar modelos de visión por computador como las CNN en la detección de enfermedades en hojas de tomate de riñón. También ayuda a los investigadores y científicos a comprender cómo el modelo está funcionando y a identificar áreas de mejora para lograr un rendimiento óptimo en la detección de enfermedades, como se puede visualizar en la Figura 14 tomada de [45].

Matriz de confusión		Estimado por el modelo			
		Negativo (N)	Positivo (P)		
Real	Negativo	a: (TN)	b: (FP)	Precisión ("precision") Porcentaje predicciones positivas correctas:	$d/(b+d)$
	Positivo	c: (FN)	d: (TP)		
		Sensibilidad, exhaustividad ("Recall") Porcentaje casos positivos detectados	Especificidad ("Specificity") Porcentaje casos negativos detectados	Exactitud ("accuracy") Porcentaje de predicciones correctas (No sirve en datasets poco equilibrados)	
		$d/(d+c)$	$a/(a+b)$	$(a+d)/(a+b+c+d)$	

Figura 14. Matriz de confusión

En la matriz proporciona una descripción detallada de la calidad de las predicciones del modelo y se compone de cuatro componentes: verdaderos positivos (VP), verdaderos negativos (VN), falsos positivos (FP) y falsos negativos (FN). Los verdaderos positivos (VP) representan las instancias en las que el modelo clasifica correctamente una hoja enferma como enferma. Los verdaderos negativos (VN) indican que el modelo identifica correctamente una hoja sana como sana. Los falsos positivos (FP) son casos en los que el modelo clasifica incorrectamente una hoja sana como enferma, y los falsos negativos (FN) reflejan las situaciones en las que el modelo no puede detectar una hoja enferma que estaba presente [46].

- **VP (verdaderos positivos):** representa la cantidad de positivos que fueron clasificados correctamente como positivos por el modelo.

- **VN (verdaderos negativos)**: representa la cantidad de negativos que fueron clasificados correctamente como negativos por el modelo.
- **FN (falsos negativos)**: representa la cantidad de positivos que fueron clasificados incorrectamente como negativos.
- **FP (falsos positivos)**: representa la de negativos que fueron clasificados incorrectamente como positivos [46].

Permite calcular métricas como la precisión, la sensibilidad, la especificidad y la puntuación F1, que proporcionan una visión completa de la eficacia del modelo en la tarea de detección de enfermedades. Las fórmulas para calcular estas métricas son las siguientes [47]:

- **Exactitud (Accuracy)**: Representa el porcentaje de datos clasificados correctamente. Se calcula dividiendo la suma de VP y VN por el total de observaciones [47].

$$\text{Exactitud} = \frac{VP + VN}{\text{Total}}$$

- **Precisión**: Indica la proporción entre el número de predicciones correctas de la clase positiva y el total de predicciones positivas realizadas [47].

$$\text{Precisión} = \frac{VP}{\text{Total clasificados positivos}}$$

- **Tasa de error**: Refleja el porcentaje de datos clasificados incorrectamente, calculado como la suma de falsos positivos (FP) y falsos negativos (FN) dividida por el total de observaciones [47].

$$\text{Tasa de error} = \frac{FP + FN}{\text{Total}}$$

- **Sensibilidad**: Representa la proporción de casos positivos que el algoritmo identificó correctamente. Se calcula dividiendo los verdaderos positivos (VP) por el total de casos positivos [47].

$$\text{Sensibilidad} = \frac{VP}{\text{Total Positivos}}$$

- **Especificidad:** Indica la proporción de casos negativos que fueron correctamente identificados por el algoritmo. Se calcula dividiendo los verdaderos negativos (VN) por el total de casos negativos [47].

$$\text{Especificidad} = \frac{\text{VN}}{\text{Total Negativos}}$$

4.5. Bibliotecas o Librerías

4.5.1. Tensorflow

La librería *TensorFlow* está dedicada al desarrollo de diversas maneras de aprendizaje automático, permitiendo en primera instancia de la ejecución en equipos sofisticados en hardware de manera eficiente [48]. Esta librería fue creada bajo la licencia Apache, desarrollado en 2011 para productos internos de Google.

La API *Tensorflow* es una librería estable que fue desarrollada en 2017 por el lenguaje de programación Python, la cual se ha convertido en una librería muy importante para comprobar estadísticas existen utilizadas para el desarrollo de aplicaciones como Deep Learning, Caffe o Theano, en [49]esto es por su fiabilidad al momento de presentar resultados, que ratifique la fiabilidad del modelo predictivo.

4.5.2. Keras

Keras es una librería del lenguaje de programación Python que permitirá desarrollar y evaluar modelos de desarrollo de aplicaciones Deep Learning. Keras soporta librerías de computación numérica como Theano y TensorFlow, la cual permite definir y entrenar modelos de redes neuronales en pocas y sencillas líneas de código [50]. Dicho de otra manera, la librería Keras es una API para Redes Neuronales que puede ser ejecutada sobre TensorFlow, es un Kit de Herramientas Cognitivas de Microsoft (CNTK) o Theano [51]. En el código abierto, escrita en el lenguaje Python altamente dinámica, que posee implementaciones de los bloques de las Redes Neuronales como funciones objetivo, activación, capaz y optimizadores. Keras es ampliamente utilizada en aplicaciones que requieren bibliotecas de aprendizaje automático debido a sus características destacadas [51]. Entre sus principales funcionalidades, se encuentran:

- **Permite el uso de redes neuronales convolucionales (CNN):** Keras permite la implementación de redes neuronales convolucionales, las cuales son fundamentales para tareas de procesamiento de imágenes y reconocimiento de patrones.
- **Permite crear prototipos fáciles y rápidos por trabajar por módulos:** debido a su enfoque modular, esta biblioteca ayuda a crear prototipos de modelos de forma rápida y sencilla.
- **Se puede ejecutar sobre el procesador y en una Unidad de Procesamiento de Gráficos (GPU):** Keras proporciona la flexibilidad de ejecutar modelos en unidades centrales de procesamiento (CPU) y unidades de procesamiento de gráficos (GPU), acelerando así el entrenamiento y la evaluación del modelo [50].

4.5.3. Sklearn

Es una biblioteca de código abierto en Python que se utiliza para el aprendizaje automático y la minería de datos. Esta biblioteca proporciona una amplia gama de herramientas y algoritmos que son esenciales para tareas relacionadas con la inteligencia artificial, como la clasificación, la regresión, el agrupamiento [52]. Sklearn se destaca por su facilidad de uso y su capacidad para integrarse eficazmente con otras bibliotecas de Python. La biblioteca *scikit-learn* [53] se utiliza para llevar a cabo análisis predictivos, abarcando tareas como la clasificación y algoritmos de agrupamiento. Esta biblioteca se basa en NumPy, SciPy y matplotlib, lo que facilita la reutilización de código que emplee estas librerías. Adicionalmente, scikit-learn incorpora diversas funciones para el pre-procesamiento de datos [53]:

- **Normalización:** Ajusta las variables numéricas de manera que tengan una media de 0 y una varianza de 1 o que se encuentren dentro de un rango específico, como [0,1]. También permite normalizar vectores para que tengan una norma de 1.
- **Transformaciones no lineales:** Basadas en cuantiles y exponentes, estas transformaciones se utilizan para modificar variables con distribuciones extremadamente sesgadas. Incluyen opciones como la transformada de Yeo-Johnson y la de Box-Cox.
- **Discretización:** Consiste en convertir una variable numérica en un conjunto de valores posibles siguiendo ciertos criterios. En casos extremos, una variable puede transformarse en solo dos valores posibles, lo que se denomina binarización.

- Manejo de valores faltantes: Cuando faltan datos de alguna variable en ciertos registros (por ejemplo, cuando un usuario no responde a una pregunta en una encuesta), es posible imputar un valor en función de algún criterio automatizable, como sustituirlo por la mediana.
- Generación de interacciones entre variables a través del empleo de polinomios.

4.6. Arquitecturas de redes convolucionales

Las arquitecturas de redes convolucionales son un tipo de red neuronal artificial que se utiliza principalmente en tareas de visión por computador, como el reconocimiento de objetos en imágenes. Las redes neuronales convolucionales (CNN) se caracterizan por tener capas de convolución que permiten aprender características específicas de las imágenes [54]. Estas arquitecturas han demostrado un alto rendimiento en una variedad de aplicaciones de procesamiento de imágenes.

4.6.1. VGG16

VGG16 es una arquitectura de red convolucional profunda que se destaca por su profundidad y simplicidad en su diseño. Fue desarrollada por el Grupo de Visualización y Geometría de la Universidad de Oxford. Esta arquitectura se compone de 16 capas convolucionales y es conocida por su capacidad para extraer características complejas de las imágenes, lo que la hace efectiva en tareas de clasificación y detección de objetos.

VGG16 se caracteriza por el uso de filtros convolucionales de tamaño 3x3 con un stride de 1 y capas de pooling de tamaño 2x2 con un stride de 2, lo que contribuye a su capacidad para capturar características jerárquicas en imágenes. La red incluye 13 capas convolucionales y 3 capas totalmente conectadas, lo que facilita el aprendizaje profundo y la clasificación de imágenes en múltiples categorías. Su arquitectura ha sido adaptada y optimizada en numerosos estudios para mejorar la precisión en tareas específicas, como la detección de objetos en tiempo real y el reconocimiento de patrones complejos [55].

El modelo VGG16 permite la detección de patrones y características locales en imágenes mediante el uso de capas convolucionales profundas, que están diseñadas para extraer y aprender características de bajo a alto nivel. Las primeras capas convolucionales de VGG16 se enfocan en detectar características básicas como bordes y texturas mediante filtros de convolución pequeños (3x3) que operan sobre pequeñas regiones de la imagen. A medida que se avanza en la red, las

capas convolucionales sucesivas combinan estas características básicas para identificar formas y patrones más complejos, gracias a la profundidad y la estructura jerárquica del modelo [56]. Este proceso se ve complementado por la utilización de funciones de activación como ReLU y la aplicación de capas de agrupamiento (pooling) que reducen la dimensionalidad mientras retienen las características más relevantes, facilitando así la detección precisa de patrones complejos en la imagen. En la Figura 15 tomadas de [56] se puede visualizar la arquitectura del modelo VGG16.

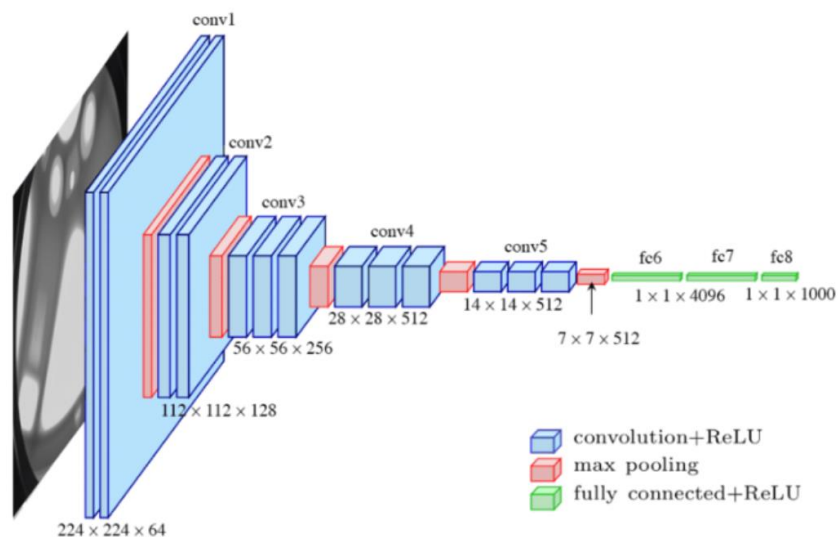


Figura 15. Arquitectura del modelo VGG16

4.6.2. ResNet50

ResNet50, o Residual Network 50, es una arquitectura de red neuronal profunda desarrollada por Kaiming. Se caracteriza por la introducción de conexiones residuales que permiten el flujo directo de información a través de las capas de la red, lo que facilita el entrenamiento de redes profundas. Esta arquitectura es ampliamente utilizada en tareas de clasificación de imágenes y detección de objetos debido a su eficiencia y precisión [57].

El modelo ResNet50 de arquitectura basada en bloques de conexión residual permite la construcción de redes extremadamente profundas sin enfrentar problemas de degradación de rendimiento. Este modelo consta de 50 capas y emplea "residual blocks" que incluyen conexiones de atajo que bypassan una o más capas, facilitando el flujo de información y gradientes a través de la red. Las conexiones residuales ayudan a mitigar el problema del desvanecimiento del gradiente, permitiendo que el modelo aprenda características complejas de las imágenes de manera más efectiva. Cada residual block se compone de dos capas convolucionales de 3×3 con Batch

Normalization y funciones de activación ReLU, lo que contribuye a la detección precisa de patrones de alto nivel como texturas y formas complejas, mejorando el rendimiento en tareas de clasificación y detección de objetos [58], en la Figura 16, tomadas de [58] se puede visualizar la arquitectura del modelo ResNet50.

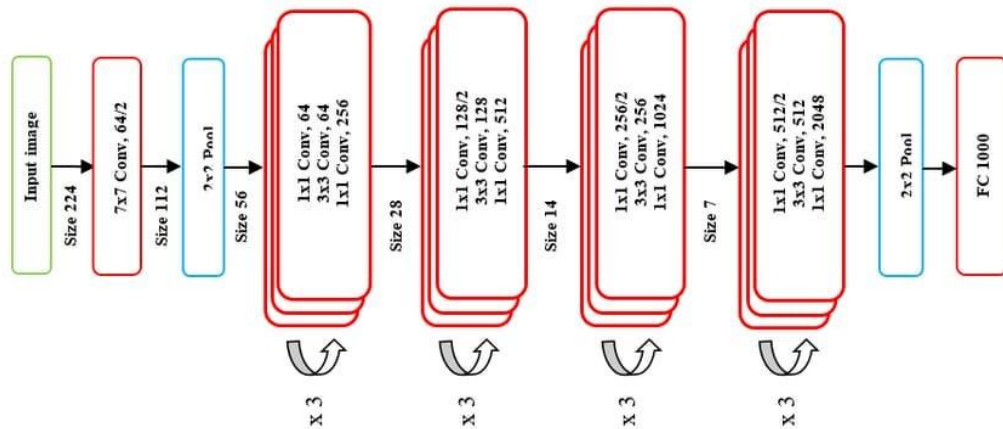


Figura 16. Arquitectura del modelo ResNet50

4.6.3. InceptionV3

InceptionV3, es una arquitectura de red convolucional que se destaca por su capacidad para extraer características de diferentes escalas y niveles de abstracción en las imágenes. Utiliza módulos de convolución de múltiples tamaños de filtro en paralelo para capturar una amplia gama de características. InceptionV3 es conocida por su rendimiento en tareas de clasificación de imágenes y detección de objetos en conjuntos de datos desafiantes [59].

La arquitectura conocida como "Inception modules," integra múltiples tipos de convoluciones en paralelo dentro de un solo bloque. Específicamente, el módulo InceptionV3 incorpora convoluciones de diferentes tamaños (1x1, 3x3 y 5x5) y una capa de agrupamiento (pooling) para capturar características de diversos tamaños y formas en las imágenes. Estas operaciones son combinadas mediante concatenación, permitiendo al modelo aprender representaciones más ricas y variadas [60]. Además, InceptionV3 incluye técnicas como la normalización de lotes (Batch Normalization) y la reducción de la dimensionalidad a través de convoluciones 1x1, lo que optimiza la eficiencia computacional y mejora la precisión en la detección y clasificación de patrones complejos, la arquitectura del modelo se puede visualizar en la Figura 17, extraída de [60].

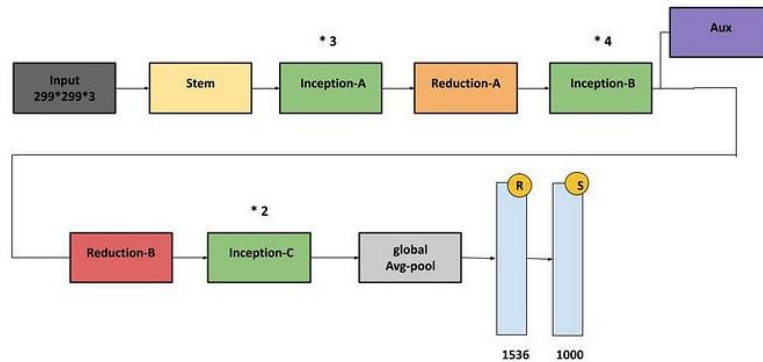


Figura 17. Arquitectura del modelo InceptionV3

4.7. Trabajos relacionados

En la Tabla 1 se muestra todos los trabajos con el título y descripción.

Tabla 1. Trabajos relacionados

Código	Título	Descripción	Referencia Bibliográfica
TA1	Convolutional Neural Networks for Texture Feature Extraction. Applications to Leaf Disease Classification in Precision Agriculture	Este trabajo de titulación se centra en el uso de Redes Neuronales Convolucionales (CNN) para extraer características de textura de hojas, con el propósito de clasificar enfermedades en agricultura de precisión.	[61]
TA2	Leaf Disease Detection Based on Lightweight Deep Residual Network and Attention Mechanism	Este trabajo propone un enfoque para la detección de enfermedades en hojas utilizando una red neuronal residual profunda ligera y un mecanismo de atención para mejorar la precisión en la detección.	[62]
TA3	Cardamom Plant Disease Detection Approach Using EfficientNetV2	En este trabajo de titulación, se utiliza EfficientNetV2 para detectar enfermedades en plantas de cardamomo, destacando su eficiencia y	[63]

		rendimiento en la detección de enfermedades.	
TA4	FieldPlant: A Dataset of Field Plant Images for Plant Disease Detection and Classification with Deep Learning	FieldPlant es un conjunto de datos de imágenes de plantas de campo creado para la detección y clasificación de enfermedades de plantas utilizando técnicas de aprendizaje profundo.	[64]
TA5	Plant Disease Detection and Classification by Deep Learning—A Review	Este trabajo proporciona una revisión del uso de técnicas de aprendizaje profundo para la detección y clasificación de enfermedades en plantas, abordando su eficacia y aplicaciones en la agricultura.	[65]
TA6	Plant Disease Detection Using Generated Leaves Based on DoubleGAN	Se propone un enfoque innovador para la detección de enfermedades en plantas utilizando hojas generadas artificialmente mediante DoubleGAN, lo que amplía las posibilidades de entrenamiento de modelos.	[66]
TA7	Deep Learning-Based Object Detection Improvement for Tomato Disease	Este trabajo de titulación se centra en mejorar la detección de enfermedades en tomates mediante el uso de técnicas avanzadas de aprendizaje profundo para la detección de objetos.	[67]
TA8	PlantDet: A Robust Multi-Model Ensemble Method Based on Deep Learning for Plant Disease Detection	PlantDet es un método robusto que combina múltiples modelos de aprendizaje profundo para la detección de enfermedades en plantas, mejorando la precisión y la generalización del modelo.	[68]

TA9	Machine Learning and Deep Learning Based Computational Techniques in Automatic Agricultural Diseases Detection: Methodologies, Applications, and Challenges	Este trabajo de titulación revisa las técnicas computacionales basadas en aprendizaje automático y profundo para la detección automática de enfermedades agrícolas, analizando metodologías, aplicaciones y desafíos asociados.	[69]
TA10	A novel deep learning method for detection and classification of plant diseases	Se presenta un método novedoso de aprendizaje profundo para la detección y clasificación de enfermedades en plantas, destacando su eficacia y rendimiento en la detección temprana de enfermedades.	[70]
TA11	Improved crossover-based monarch butterfly optimization for tomato leaf disease classification using convolutional neural network	Este trabajo de titulación propone una mejora en el algoritmo de optimización de mariposa monarca basada en cruzamiento para la clasificación de enfermedades en hojas de tomate utilizando redes neuronales convolucionales, mejorando la precisión y la eficiencia del modelo.	[71]
TA12	A robust deep learning approach for tomato plant leaf disease localization and classification	Se presenta un enfoque robusto de aprendizaje profundo para la localización y clasificación de enfermedades en hojas de plantas de tomate, destacando su capacidad para identificar y clasificar múltiples enfermedades con alta precisión.	[72]
TA13	Design of efficient techniques for tomato leaf	Este trabajo propone técnicas eficientes para la detección de	[73]

	disease detection using genetic algorithm-based and deep neural networks	enfermedades en hojas de tomate utilizando algoritmos genéticos y redes neuronales profundas, mejorando la precisión y la eficiencia del proceso de detección.	
TA14	Detecting the Simultaneous Occurrence of Strawberry Fungal Leaf Diseases with a Deep Normalized CNN	Se presenta un método para detectar la ocurrencia simultánea de enfermedades fúngicas en hojas de fresa utilizando una CNN normalizada profunda, destacando su capacidad para identificar múltiples enfermedades con alta precisión.	[74]
TA15	Plant Disease Detection in Imbalanced Datasets Using Efficient Convolutional Neural Networks with Stepwise Transfer Learning	Este trabajo propone un enfoque para la detección de enfermedades en plantas en conjuntos de datos desbalanceados utilizando redes neuronales convolucionales eficientes con aprendizaje de transferencia escalonado, mejorando la precisión y la generalización del modelo en condiciones desafiantes.	[75]
TA16	Early real-time detection algorithm of tomato diseases and pests in the natural environment	Se presenta un algoritmo de detección temprana en tiempo real de enfermedades y plagas de tomate en el entorno natural, destacando su capacidad para identificar y clasificar rápidamente amenazas para los cultivos de tomate.	[76]

5. Metodología

5.1. Área de estudio

El presente Trabajo de Titulación (TT) se llevó a cabo en invernaderos pertenecientes a grupos de agricultores apoyados por el Municipio de Saraguro. Estos se encuentran ubicados en comunidades como Tuncarta, Tenta y Urdaneta. La selección de estos sitios se realizó con el apoyo del Técnico de Fomento agropecuario del Municipio de Saraguro, debido a que son las áreas con mayor producción de tomate de riñón y se han convertido en las principales abastecedoras de este producto en los mercados locales.

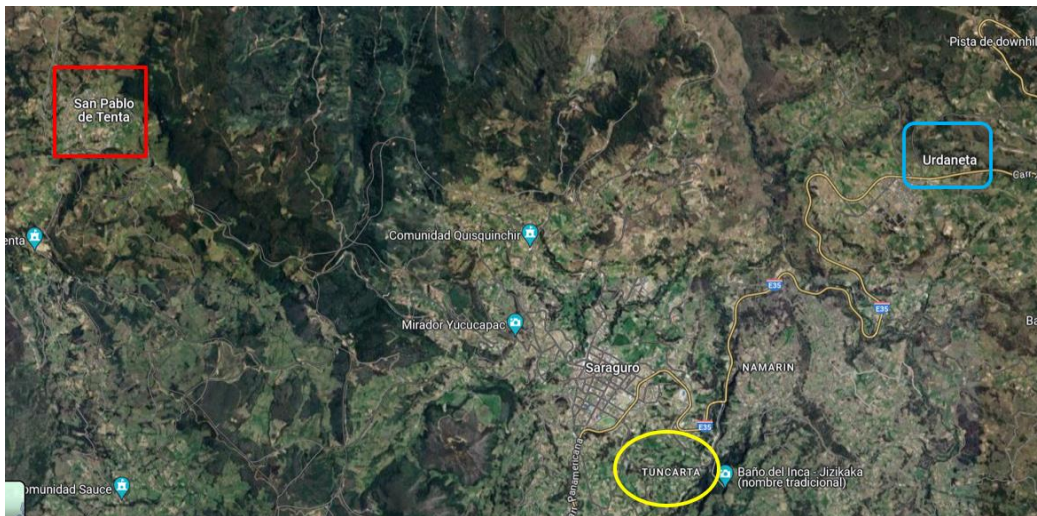


Figura 18. Comunidades de Saraguro donde se recolectaron las hojas de tomate

5.2. Procedimiento

Para llevar a cabo el presente TT, se utilizó la metodología *Design Thinking*, ya que, gracias a su enfoque, permitió un desarrollo estructurado para cada etapa del proceso. Para la realización de este trabajo, se adaptaron cuatro fases principales de la metodología: Definir, Idear, Prototipar y Evaluar. Las fases de esta metodología se presentan en la Figura 19.

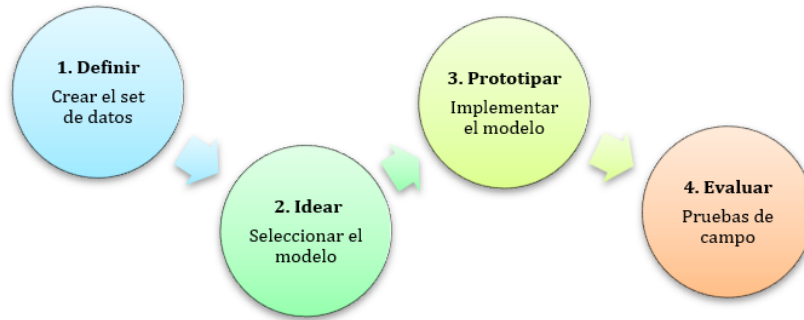


Figura 19. Fases de la metodología Design Thinking implementados en la investigación

5.2.1. Objetivo 1: Identificar métodos, técnicas basadas en inteligencia artificial y recopilar datasets con imágenes de hojas sanas y enfermas del tomate de riñón.

Fase 1 - Definir:

- Recolectar imágenes de la hoja de tomate de riñón en estado sano y afectadas por tizón tardío.
- Realizar el pre-procesamiento de las imágenes adquiridas para la creación del conjunto de datos que permitirá el entrenamiento del modelo de visión por computador.
- Dividir los datos en conjunto de entrenamiento, evaluación y prueba.

5.2.2. Objetivo 2: Desarrollar un modelo de visión por computador para la detección automática de la enfermedad Tizón Tardío (*Phytophthora infestans*) en las hojas del tomate de riñón.

Fase 2 - Idear:

- Investigar los modelos *Deep learning* más utilizados y sus características para el reconocimiento de enfermedades foliares.

Fase 3 - Prototipar:

- Entrenar el modelo de visión por computador utilizando los datos de entrenamiento

5.2.3. Objetivo 3: Validar el modelo de visión por computador en un escenario de prueba aplicada a los invernaderos del cantón Saraguro.

Fase 4 - Evaluar

- Validar y contrastar los resultados obtenidos en un escenario de prueba con la opinión de un experto.

5.3. Recursos

5.3.1. Recursos científicos

- ***Método científico***

El método científico es un proceso sistemático que permite a los investigadores obtener conocimiento objetivo sobre la realidad. El método emplea principalmente el enfoque hipotético-inductivo, que implica observar sucesos naturales y establecer un marco interpretativo [77]. Este método se aplicó en el desarrollo del presente Trabajo de Titulación (TT), ya que se aplicó para constatar los resultados obtenidos en cada uno de los objetivos (véase Sección Resultados), así como la Discusión (véase Sección Discusión) y Conclusiones (véase Sección Conclusiones).

- ***Método Analítico***

El método analítico es importante en el ámbito academia, ya que facilita el proceso de la investigación, al descomponer al problema complejo en los componentes básicos de lo general a lo específico, permitiendo analizar cada elemento de manera individual, para integrarlos en una visión global [78]. El método analítico se aplicó a esta investigación para desglosar el problema complejo de la detección de la enfermedad del tizón tardío en hojas de tomate de riñón en componentes más manejables y específicos. Este enfoque permitió una comprensión de cada uno de los aspectos involucrados en el proceso de detección. Inicialmente, se identificaron los métodos y técnicas basadas en inteligencia artificial más adecuados para este tipo de análisis. Posteriormente, se desarrolló un modelo de visión por computador enfocado en la detección de la enfermedad, y finalmente, se validó este modelo en un entorno de prueba aplicado a los invernaderos del cantón Saraguro. Se hizo la descomposición, lo que facilitó el tratamiento de cada etapa del trabajo de titulación, desde la recolección de datos hasta la validación práctica y las decisiones que se realizaron basadas en datos y evidencia.

- ***Método de Barbara Kitchenham***

Este método es una técnica de revisión sistemática de literatura, la cual fue desarrollada por Bárbara Kitchenham, una destacada investigadora en el campo de la ingeniería de software [79]. En este trabajo de TT, se aplicó el método para la en la revisión sistemática de literatura, se aplicó dicho método para la recolección y evaluación de información relevante, con el fin de identificar las mejores técnicas basadas en *deep learning* para el reconocimiento de enfermedades en las hojas de tomate de riñón. A través de esta metodología, permitió extraer y analizar datos de diversas

fuentes, facilitando la adquisición de imágenes provenientes de diferentes bases de datos, esenciales para la construcción del conjunto de datos utilizado en el entrenamiento, validación y prueba a través del modelo. Además, esta revisión sistemática proporcionó una base documental clave para el desarrollo del modelo de visión por computador. Esta recopilación de estudios permitió identificar enfoques probados y validados en el ámbito del *deep learning* aplicados a la detección de enfermedades en plantas. Gracias a este análisis, fue posible seleccionar las arquitecturas y técnicas más adecuadas, optimizando así la precisión y eficiencia del modelo. Para más detalles sobre los estudios revisados y su aplicación en el desarrollo del modelo, ver el **Anexo 1**.

5.3.2. Recursos Técnicos

Durante el desarrollo de esta investigación, se hicieron uso de herramientas colaborativas esenciales para facilitar la comunicación y la colaboración efectiva entre los miembros del equipo de investigación.

- ***Entorno de desarrollo en línea***

Google Colab fue una herramienta fundamental en este trabajo de titulación. Como entorno de desarrollo en línea basado en la plataforma, proporcionó acceso gratuito a GPUs (Unidades de Procesamiento Gráfico), lo cual permitió el entrenamiento de modelos de *deep learning*.

- ***Plataformas colaborativas***

Se utilizó Google Drive, para organizar y gestionar la información, incluyendo los datos para el entrenamiento del modelo, esta herramienta permitió el almacenamiento, optimizando el proceso de carga y manejo de datos durante el entrenamiento del modelo.

- ***Streamlit***

En el trabajo de titulación, se utilizó Streamlit para la implementación de la página web desarrollada. Por su capacidad de crear las aplicaciones ya que el framework tiene las librerías de Python facilitó la creación de la aplicación de manera rápida y eficiente, Streamlit permitió integrar de forma efectiva las visualizaciones de la presentación de los resultados obtenidos. La facilidad de uso y la integración con herramientas de visualización fueron factores clave para el éxito en la implementación del prototipo web.

5.4. Participantes

Para el desarrollo del presente TT enfocado en la línea de investigación de Inteligencia Artificial, se contó con los siguientes participantes:

- La estudiante Sisa Ñusta Parra Tene como autora del presente TT, el conjunto de actividades inició desde el planteamiento de la propuesta de del TT, desarrollo y finalización de cada uno de los objetivos planteados.
- Ing. Roberth Figueroa, Mg. cumpliendo el papel de tutor académico y director del TT, quien supervisó los avances académicos y técnicos desarrollados por la autora.
- El personal de apoyo del GAD municipal de Cantón Saraguro, el Ing. Miguel Condolo, Técnico del departamento de Gestión Ambiental y Producción fomento agropecuario, quien contribuyó a la validación del funcionamiento del modelo.

6. Resultados

En esta sección, se presentan los resultados obtenidos en cada uno de los objetivos específicos del presente TT que han contribuido al cumplimiento del objetivo principal. Se detallan las actividades y tareas correspondientes que se llevaron a cabo para alcanzar con éxito dichos objetivos.

6.1. Objetivo 1: Identificar métodos, técnicas basadas en inteligencia artificial y recopilar Datasets con imágenes de hojas sanas y enfermas del tomate de riñón.

En este primer objetivo, se aborda la fase de "Definir" dentro de la metodología Design Thinking. El enfoque principal consiste en identificar las técnicas de inteligencia artificial más adecuadas para la detección de enfermedades en las hojas del tomate de riñón. Además, se procedió a la recopilación de un conjunto de datos compuesto por imágenes de hojas sanas y enfermas, las cuales serán fundamentales para entrenar los modelos. Esta fase es crucial, ya que establece las bases necesarias para avanzar en el desarrollo del proyecto, asegurando que se cuenten con los datos y métodos apropiados desde el inicio.

Fase 1: Definir

6.1.1 Tarea 1: Recolectar imágenes de la hoja de tomate de riñón en estado sano y con tizón tardío.

6.1.1.1 Conjunto de datos Propio

La obtención de imágenes de la hoja de tomate de riñón con tizón tardío y sanas, se llevó a cabo mediante visitas en las parroquias de Urdaneta, Tenta, y en la comunidad de Tuncarta del cantón Saraguro, provincia de Loja, que son sectores en donde existe mayor cultivo de este producto, un ejemplo de estos cultivos se muestra en la Figura 20.

Se realizó la recolección y clasificación de hojas de tomate de riñón en dos categorías: "hojas sanas" y "hojas enfermas", centrándose en las afectadas por el tizón tardío. Las hojas fueron transportadas para su captura fotográfica de manera individual sobre un fondo blanco, una técnica esencial para estandarizar las imágenes. El uso de este fondo permitió resaltar las características visuales de las hojas, como el color y los signos de enfermedad, sin interferencias externas, facilitando un análisis preciso durante el procesamiento de los datos, más detalles sobre este proceso se encuentran en el **Anexo 2**.

Los recursos utilizados para el desarrollo de esta tarea son: cámara de dispositivo móvil, tablero de madera con fondo blanco, el detalle de estos se ven reflejados en la Tabla 2.



Figura 20. Cultivos de tomate en el cantón Saraguro

Tabla 2. Herramientas utilizadas para la recolección de datos (imágenes)

Recursos 1			
Cámara de teléfono móvil	Apertura de cámara	Tamaño (Megapíxeles)	Procesador de móvil
Xiaomi Redmi Note 10	f/4.5	50 MP	MediaTek Helio G88 GPU ARM Mali-G52
Recurso 2			
Tablero de madera	Dimensiones(cm)		Color Fondo
	50x50 cm		Blanco

Este proceso de adquisición de imágenes resultó en un total de 730 imágenes de hojas enfermas con tizón tardío y 717 imágenes de hojas sanas de tomate de riñón, conformando así el primer conjunto de datos adquiridos por la investigadora, al cual se denominó “*Dataset1*”

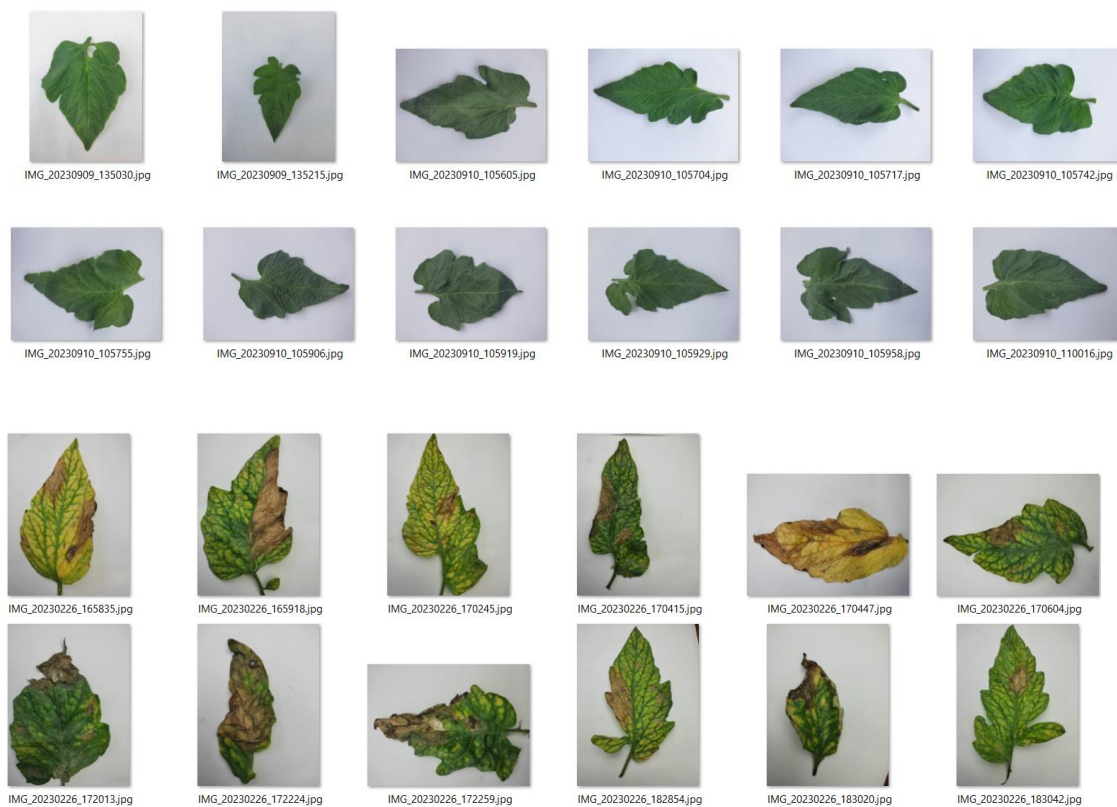


Figura 21. Conjunto de datos propio de la hoja de tomate de riñón sana y con tizón tardío

6.1.1.2 Conjunto de datos en la web

Durante la recolección de datos para este trabajo de titulación, se llevó a cabo una revisión sistemática de literatura. Durante esta etapa, se realizó una búsqueda meticulosa de trabajos previos y estudios relacionados que se centraran en la detección del tizón tardío en las hojas del tomate de riñón. Esta indagación permitió el acceso a bases de datos previamente utilizadas en investigaciones relacionadas, lo que posibilitó la formación de un repositorio de datos sólido y altamente relevante. Como resultado de este análisis, se identificaron tres repositorios de imágenes de hojas de tomate, los cuales se detallan en la Tabla 3.

Tabla 3. Repositorios de hojas de tomate de riñón

Dataset	Repositorio
PlantVillage	https://www.kaggle.com/datasets/emmarex/plantdisease

PlantDoc <https://github.com/pratikkayal/PlantDoc-Dataset/tree/master/train/Tomato%20leaf%20late%20blight>

Imagenet <http://www.image-net.org/>

Tras un análisis exhaustivo de los conjuntos de datos, se determinó que PlantVillage es el más adecuado, con 54,000 imágenes que abarcan diversas categorías de plantas y enfermedades, incluyendo el tomate de riñón. En comparación, aunque ImageNet posee más de 14 millones de imágenes, estas abarcan una amplia gama de categorías no etiquetadas, lo cual habría requerido un esfuerzo considerable para seleccionar y filtrar las imágenes necesarias para esta investigación. Adicionalmente, se consideró el conjunto de datos PlantDoc, que contiene imágenes tomadas en invernaderos; sin embargo, presenta un alto nivel de ruido que dificulta la extracción de características de las enfermedades, lo que complica la clasificación entre hojas sanas y enfermas.



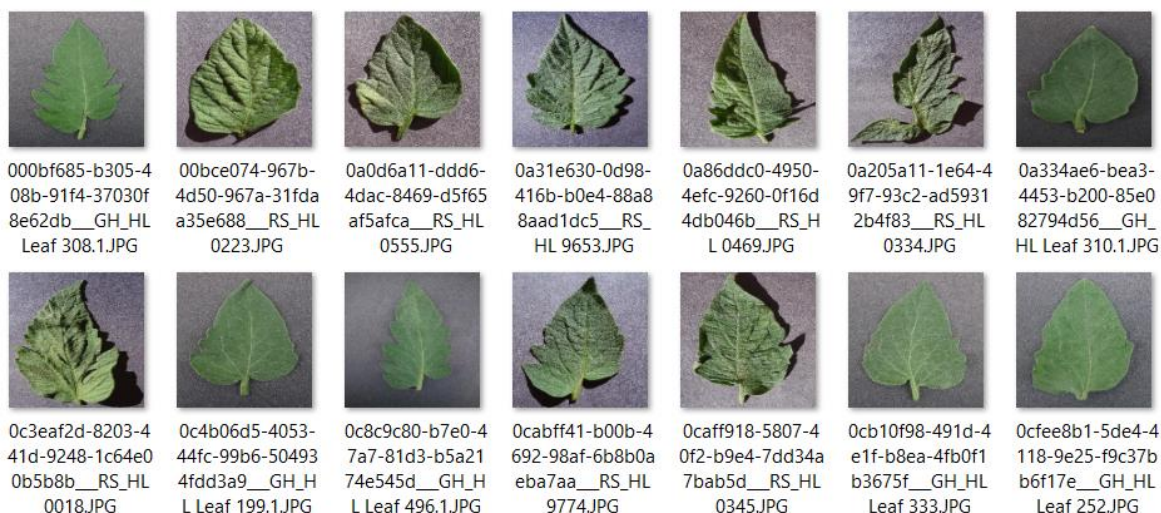


Figura 22. Conjunto de imágenes de la hoja de tomate de riñón con tizón tardío y sana - Plant Village

Como se muestra en la Figura 22, las imágenes obtenidas de PlantVillage muestran hojas tanto sanas como enfermas con un fondo de color gris. Esta característica del conjunto de dataset es especialmente beneficiosa, ya que permitió que el modelo adquiriera la capacidad de identificar no solo la forma y los bordes de las hojas, sino también los patrones internos que son cruciales para la detección. Dadas las consideraciones sobre el tamaño del conjunto de dataset y la capacidad de las clases, se tomó la decisión de seleccionar el conjunto de imágenes de PlantVillage. Este conjunto de dataset no solo ofrece una amplia variedad de imágenes clasificadas, sino que también cumple con las características esenciales requeridas para su uso en un modelo de clasificación, como el tamaño y el fondo de cada imagen. Este conjunto fue adquirido desde el repositorio de Kaggle, se extrajo solamente la categoría de hojas de tomate sanas y con tizón tardío, conformando un segundo set de datos denominado “**Dataset2**”, mismo que se detalla en la Tabla 4, donde se presenta el número total de imágenes recuperadas.

Tabla 4. Descripción de los conjuntos de datos sin procesar

Conjunto de datos	Hojas Sanas	Tizón Tardío	Total
Dataset1	717	730	1447
Dataset2	1591	1760	3351
Total	2308	2490	4798

6.1.2 Tarea 2: Realizar el pre-procesamiento de las imágenes adquiridas para la creación del conjunto de datos que permitirá el entrenamiento del modelo de visión por computador.

Antes de proceder al entrenamiento de los modelos de clasificación, es necesario realizar el pre-procesamiento de las imágenes adquiridas. Este paso es crucial para asegurar que el conjunto de datos esté en óptimas condiciones y que el modelo de visión por computador pueda aprender de manera eficiente. El pre-procesamiento incluye diversas técnicas como la categorización, ajuste de tamaño y aumento de datos, garantizando que las imágenes sean consistentes y representativas de las categorías "hojas sanas" y "hojas enfermas". En este caso, el conjunto de datos se utilizó para entrenar tres modelos diferentes de redes neuronales convolucionales (CNN): VGG16, ResNet50 e InceptionV3. Cada uno de estos modelos tiene diferentes arquitecturas y profundidades, lo que permitió evaluar cuál es más efectivo en función del rendimiento y precisión en la tarea de clasificación de imágenes.

6.1.1.3 Categorización de las imágenes

Cada imagen recopilada se almacenó en una carpeta, lo que permitió clasificar cada dato de manera correcta y necesaria para los siguientes pasos. Las imágenes fueron categorizadas en dos grupos principales: "hojas sanas" y "hojas con tizón tardío". Este proceso se realizó en colaboración con un experto en la materia, quien garantizó la correcta asignación de cada imagen a su categoría correspondiente, en donde se colocó cuidadosamente en la carpeta correspondiente según su estado, es decir, si la hoja se encontraba en buen estado, se ubicaba en la carpeta "Sanas", mientras que, si presentaba signos de la enfermedad del "Tizón Tardío". Esta clasificación en carpetas puede visualizarse en la Figura 23.

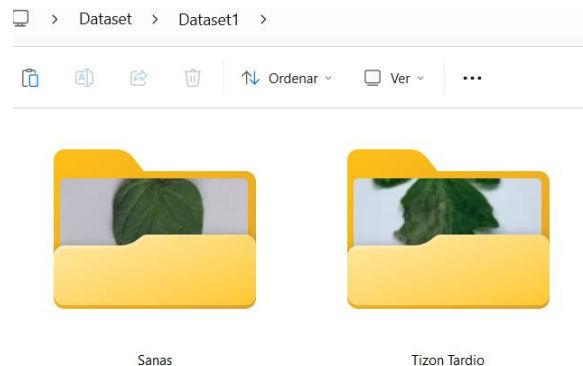


Figura 23. Organización de imágenes privadas en carpetas

6.1.1.4 Aumento de datos (Data Augmentation)

Con el fin de enriquecer el conjunto de datos propio y aumentar el número de imágenes disponibles para el entrenamiento del modelo, se aplicó las siguientes técnicas de aumento de datos:

- Rotación en incrementos de 90 grados en sentido de las manecillas del reloj.
- Rotación de -15° girando a lo contrario de las agujas del reloj y $+15^\circ$ en sentido a las agujas del reloj.
- Aplicación de brillo con un ajuste de -25% a $+25\%$ incrementando la iluminación.

En la Figura 24 se puede visualizar el código empleado para la aplicación de estas técnicas en el conjunto de datos.

```
# Abrir la imagen
with Image.open(ruta_imagen) as img:
    os.makedirs(directorio_destino, exist_ok=True)

    # Definir las rotaciones de 90° y aleatorias
    rotaciones_90 = [90]
    rotaciones_aleatorias = [-15, 15]
    nombres_transformaciones = []

    # Aplicar rotaciones de 90° en sentido horario, antihorario, y volteada
    for rotacion in rotaciones_90:
        img_rotada = img.rotate(rotacion, expand=True)
        nombre_archivo = f"rotacion_{rotacion}.jpg"
        ruta_completa_destino = os.path.join(directorio_destino, nombre_archivo)
        img_rotada.save(ruta_completa_destino)
        nombres_transformaciones.append(ruta_completa_destino)

    # Aplicar rotaciones aleatorias entre -15° y +15°
    angulo_aleatorio = random.uniform(-15, 15)
    img_rotada_aleatoria = img.rotate(angulo_aleatorio, expand=True)






    # Ajustar el brillo entre -25% y +25%
    factor_brillo = random.uniform(0.75, 1.25) # 0.75 representa -25% y 1.25 representa +25%
    enhancer = ImageEnhance.Brightness(img_rotada_aleatoria)
    img_brillo_ajustado = enhancer.enhance(factor_brillo)

    # Guardar imagen con rotación aleatoria y ajuste de brillo
    nombre_archivo = f"rotacion_aleatoria_{angulo_aleatorio:.2f}_brillo_{factor_brillo:.2f}.jpg"
    ruta_completa_destino = os.path.join(directorio_destino, nombre_archivo)
    img_brillo_ajustado.save(ruta_completa_destino)
    nombres_transformaciones.append(ruta_completa_destino)
```

Figura 24. Script para el aumento de datos en Google Colab

En la Tabla 5 se visualiza brevemente un ejemplo de la imagen original y sus variantes de acuerdo a la técnica aplicada.

Tabla 5. Ejemplo de uso de la aumentación de datos.

Imagen original en estado tizón tardío	Primer dato con generando aumentación de dato	Segundo dato con generando aumentación de dato
		
		

En la Tabla 6 se detalla el total de imágenes que conforman el conjunto de dato propio luego de haber aplicado técnicas de aumento de datos.

Tabla 6. Descripción del conjunto de datos propio

Categoría	N.º Imágenes	Con Aumento de datos
Hojas Sanas	717	2021
Tizón Tardío	730	2028
Total	1447	4049

Finalizado este proceso de aplicación de técnicas de aumento de datos, se logró ampliar el set de datos a un total de 4049 imágenes, abarcando hojas de tomate tanto sanas como afectadas por el tizón tardío. A este set con aumento de datos se añadió el conjunto de datos de PlantVillage formando un tercer set de datos denominado “**Dataset3**”, en la Tabla 7 se muestran los conjuntos de datos que serán utilizados para la evaluación de los modelos a implementarse.

Tabla 7. Número total del conjunto de datos

Conjunto de datos	Total
Dataset1	4049
Dataset2	3351
Dataset3	7400
Total	14800

6.1.3 Tarea 3: Dividir los datos en entrenamiento, evaluación y prueba.

La división del conjunto de datos, es esencial para evaluar y validar el rendimiento de los modelos de clasificación. Como se menciona en [70] se divide el conjunto de datos en tres grupos mutuamente excluyentes: entrenamiento, validación y prueba, lo cual permite que el modelo aprenda patrones de los datos de entrenamiento, se ajuste y optimice durante la validación, y finalmente, se evalúe de manera justa y no sesgada en el conjunto de prueba. Esto se hace para proporcionar una evaluación imparcial del modelo después del entrenamiento.

El proceso de división de datos se llevó a cabo de manera rigurosa, y su implementación se basó en prácticas fundamentales de aprendizaje automático. Para lograrlo, se utilizó un script en un entorno Jupyter Notebook, cuyos parámetros de entrada fueron cuidadosamente configurados para garantizar una separación efectiva.

Se realizó la división de cada uno de los sets obtenidos en etapas anteriores, los cuales son: "Dataset1", "Dataset2" y "Dataset3", especificando los porcentajes esenciales: el porcentaje de muestra de entrenamiento, que representaba el 70% de las imágenes, y el porcentaje de muestra de validación, que constituía el 20%, el 10% restante destinado al porcentaje de pruebas. Finalmente se obtuvo la estructura que se indica en la Figura 25, tanto en términos de división en datos de entrenamiento, validación y prueba, como en la categorización en "sana" y "con tizón tardío".



Figura 25. Estructura del dataset de hojas enfermas y sanas de tomate de riñón

6.2. Objetivo 2: Desarrollar un modelo de visión por computador para la detección automática de la enfermedad Tizon Tardío (*Phytophthora infestans*) en las hojas del tomate de riñón.

Fase 2: Idear

En esta fase, se identificaron y exploraron las soluciones tecnológicas más prometedoras para alcanzar el objetivo de detección automática. Se investigaron los modelos de *deep learning* más utilizados en el reconocimiento de enfermedades foliares, evaluando sus características, ventajas y limitaciones. Esta investigación permitió seleccionar las arquitecturas más adecuadas y adaptar las técnicas a las necesidades específicas del proyecto.

6.2.1 Tarea 4: Investigar los modelos *Deep learning* más utilizados y sus características para el reconocimiento de enfermedades foliares.

En el contexto de esta investigación, se llevó a cabo una revisión literaria. El objetivo principal de esta revisión fue identificar y analizar los modelos *deep learning* empleados en la identificación y clasificación de enfermedades en hojas de plantas vegetales, especialmente, el tizon tardío en la hoja de tomate de riñón. La revisión literaria abarcó diversas fuentes académicas e informes relacionados con el tema, que se encuentren publicados a partir del año 2019 en adelante.

El desarrollo del protocolo de revisión tuvo un enfoque metodológico de Bárbara Kitchenham, comenzando con la formulación de preguntas que dirigieron la revisión de manera focalizada. Para planificar la búsqueda, se utilizó el método PICOC, adaptado a la investigación al excluir el criterio de "Comparación". Esta estrategia permitió estructurar la revisión de manera

eficiente y asegurar que el estudio abordara los aspectos más relevantes del tema en cuestión. Las fuentes seleccionadas para la búsqueda de trabajos e investigaciones relacionadas incluyeron:

- IEEE Digital Library
- ACM Digital Library
- Springer Link
- SciELO

Las palabras clave definidas y utilizadas para formar la cadena de búsqueda para cada repositorio de búsqueda fueron:

<ul style="list-style-type: none"> - Compute vision - Deep learning - Leaf diseases - Methods - Models 	<ul style="list-style-type: none"> - Techniques - Classification - Dataset - Detection.
---	---

A partir de estas palabras clave, se definieron cadenas de búsqueda específicas para cada biblioteca virtual, como lo muestra en la Tabla 8.

Tabla 8. Cadenas de búsqueda para la revisión de trabajos relacionados

Bibliotecas Virtuales	Cadenas de búsqueda
ACM Digital Library	("leaf diseases") AND ("computer vision" OR "machine learning") AND ("methods" OR "models" OR "techniques")
IEEE Digital Library	("leaf diseases") AND ("computer vision" OR "machine learning") AND ("methods" OR "models" OR "techniques")
Scielo	("Algorithms" OR "Computer Vision" OR "Image Processing" OR "Machine Learning" OR "Convolutional Neural Networks" OR "digital Image Classification") AND ("Disease Detection" OR "Plant Disease classification" OR "Leaf Disease Detection")

	date: 2018-2023
	WoS Áreas Temáticas: Horticultura
Science direct	("Algorithms" OR "Computer Vision" OR "Image Processing" OR "Machine Learning" OR "Convolutional Neural Networks" OR "Digital Image Classification") AND ("Disease Detection" OR "Plant Disease clasification" OR "Leaf Disease Detection")

Como resultado de la búsqueda se obtuvo 240 trabajos relacionados, de los cuales, luego de distintos análisis, se aceptaron 16 estudios que cumplían con los criterios de selección y que permitían dar respuesta a las preguntas de investigación que se muestran en la Tabla 9.

Tabla 9. Preguntas de investigación para RSL

ID	Preguntas de Investigación
PI_1	¿Cuáles son los enfoques existentes basados en Visión por Computador para identificar las enfermedades en hoja?
PI_2	¿Qué modelos de aprendizaje automático son más efectivos para la detección temprana de enfermedades en hojas de tomate de riñón?
PI_3	¿Cuál es la influencia del tamaño y calidad del dataset en el rendimiento de los modelos de inteligencia artificial en la identificación de hojas sanas y enfermas del tomate de riñón?

- PI_1: ¿Cuáles son los enfoques existentes basados en Visión por Computador para identificar las enfermedades en hoja?

En respuesta a esta pregunta, se encontró que, las técnicas implementadas para identificar enfermedades en hojas incluyen: Support Vector Machine (SVM), Region-Based Convolutional Neural Network (RCNN), y Redes Neuronales Convolucionales (CNN), como se puede observar en la Figura 26.

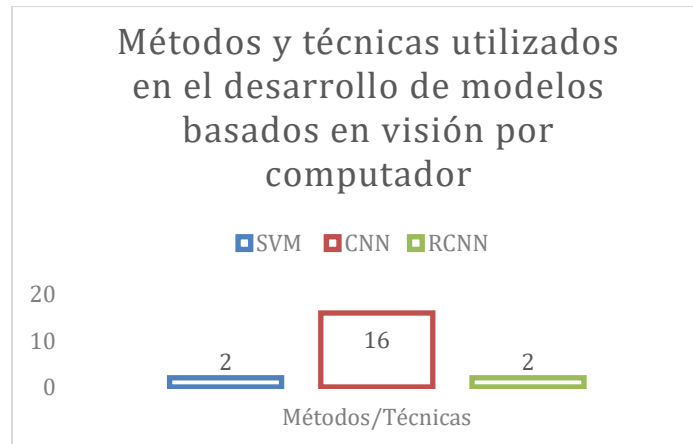


Figura 26. Gráfica de métodos utilizados para la identificación de enfermedades en hojas

Como indica la Figura 26, de los 16 artículos sometidos a análisis, todos ellos hacen hincapié en el empleo de Redes Neuronales Convolucionales (CNN) como la técnica predominante para llevar a cabo la clasificación de enfermedades en plantas. En la Tabla 10 se observa que las investigaciones realizadas en los distintos estudios, incorporan técnicas de aprendizaje automático como Support Vector Machine (SVM) o Redes Neuronales Recurrentes (RNN). Estas metodologías y técnicas se aplican en diversos contextos y con enfoques específicos para abordar la problemática de la detección de enfermedades en plantas mediante la visión por computador.

Tabla 10. Métodos y técnicas basadas en visión por computador

Ref.	Título	Métodos/Técnicas	Año	Ref.
TA01	Neural Networks for Texture Feature Extraction. Applications to Leaf Disease Classification in Precision Agriculture	CNN, SVM	2021	[61]
TA02	Leaf Disease Detection Based on Lightweight Deep Residual Network and Attention Mechanism	CNN	2023	[62]
TA03	Cardamom Plant Disease Detection Approach Using EfficientNetV2	CNN	2022	[63]
TA04	FieldPlant: A Dataset of Field Plant Images for Plant Disease Detection and Classification with Deep Learning	CNN	2023	[64]

TA05	Plant Disease Detection and Classification by Deep Learning—A Review	CNN	2021	[65]
TA06	Plant Disease Detection Using Generated Leaves Based on DoubleGAN	CNN, RCNN	2022	[66]
TA07	Deep Learning-Based Object Detection Improvement for Tomato Disease	CNN	2020	[67]
TA08	PlantDet: A Robust Multi-Model Ensemble Method Based on Deep Learning for Plant Disease Detection	CNN	2023	[68]
TA09	Machine Learning and Deep Learning Based Computational Techniques in Automatic Agricultural Diseases Detection: Methodologies, Applications, and Challenges	CNN, RNN, SVM	2022	[69]
TA10	A novel deep learning method for detection and classification of plant diseases	CNN	2022	[70]
TA11	Improved crossover-based monarch butterfly optimization for tomato leaf disease classification using convolutional neural network	CNN	2021	[71]
TA12	A robust deep learning approach for tomato plant leaf disease localization and classification	CNN	2022	[72]
TA13	Design of efficient techniques for tomato leaf disease detection using genetic algorithm-based and deep neural networks	CNN	2022	[73]
TA14	Detecting the Simultaneous Occurrence of Strawberry Fungal Leaf Diseases with a Deep Normalized CNN	CNN	2022	[74]
TA15	Plant Disease Detection in Imbalanced Datasets Using Efficient Convolutional Neural Networks with Stepwise Transfer Learning	CNN	2021	[75]
TA16	Early real-time detection algorithm of tomato diseases and pests in the natural environment	CNN	2021	[76]

Para el desarrollo del presente TT, centrado en la identificación del tizón tardío en las hojas de tomate, se tomó la decisión de emplear modelos de Redes Neuronales Convolucionales (CNN). Esta elección se fundamenta en una revisión de literatura, que ha corroborado que los modelos CNN son actualmente los más predominantes y efectivos en la clasificación de imágenes, especialmente en aplicaciones relacionadas con la visión por computador. Los modelos CNN destacan por su capacidad para extraer características de relevancia en las imágenes y aprender patrones complejos, características que los hacen sumamente apropiados para la tarea de identificar de las enfermedades en las hojas vegetales.

- PI_2: ¿Qué modelos de aprendizaje automático son más efectivos para la detección temprana de enfermedades en hojas de tomate de riñón?

Tabla 11. Modelos de aprendizaje automático más efectivos

Ref.	Modelos	Precisión
[61]	AlexNet	89,25%
	ResNet50	89,79%
	VGG16	93,94%
[62]	GoogLeNet+InceptionV3	91,19%
	AlexNet	95,64%
	VGG16	98,74%
	Xception145	98,68%
	ResNet18	98,02%
[63]	EfficientNetV2-S	98,28%
	EfficientNetV2-L	98,26%
[64]	MobileNet	82,9%
	VGG16	80,54%
	InceptionResNetV2	81,81%
	InceptionV3	82,54%
[65]	AlexNet	96,53%
	VGG16	95,33%
	ResNet50	99,95%
[66]	VGG16	99,06%

	ResNet50	99,00%
	DenseNet121	99,22%
	Faster RCNN	97,01%
[67]	Faster RCNN-mobile	97,37%
	Faster RCNN-res101	98,54%
	InceptionResNetV2	79,39%
	EfficientNetV2L	93,44%
[68]	Xception	91,54%
	VGG16	81,22%
	VGG19	85,72%
	SVM	99,33%
	AlexNet	97,22%
[69]	VGG16	99,25%
	GoogleNet	99,69%
	ResNet50	96,27%
	GoogleNet+ InceptionV3	99,35%
[70]	MobileNet-Beta	99,85%
	EfficientNet	99,94%
	VGG16	99,98%
[71]	InceptionV3	99,97%
	GoogleNet	87,16%
	ResNet-101	89,95%
[72]	Xcepción	88,25%
	VGG-19	90,39%
	ResNet50	99,97%
	VGG16	97,1%
[73]	ResNet50	96,7%
	GoogleNet	88,50%
[74]	VGG 16	94,69%
	Resnet-50	88,94%

	MobileNetV3	98,39%
[75]	VGG-16	71,40%
	InceptionV3	99.91%
[76]	YOLOv3	91,81%

De acuerdo con el análisis de los resultados presentados en la Tabla 11, los modelos CNN predominantes han demostrado un desempeño óptimo en términos de precisión, estableciendo un umbral deseado igual o superior al 90%. En base a este análisis, se han destacado tres modelos por su alto rendimiento en cuanto a precisión, entre los cuales se encuentran los modelos VGG16, ResNet50 e InceptionV3, que presentan tasas de precisión del 99.98%, 99.97% y 99.97%, respectivamente Figura 25. Por esto, se seleccionaron estas redes para ser implementados en este Trabajo de Titulación, mismo que serán entrenados y evaluados con un conjunto específico de imágenes posteriormente se compararán en función de sus resultados finales para determinar cuál de ellos es el más adecuado para detectar el tizón tardío en la hoja de tomate.

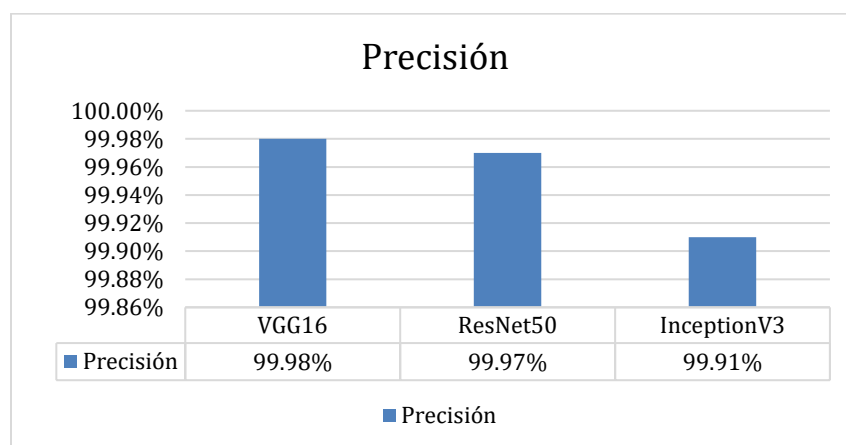


Figura 27. Modelos seleccionados para ser entrenados y evaluados

Fase 3: Prototipar

Con la información obtenida en la fase de ideación, se procedió a entrenar el modelo de visión por computador utilizando los datos de entrenamiento recopilados. Este proceso incluyó la implementación y ajuste del modelo para que pudiera identificar y clasificar correctamente los síntomas de la enfermedad en las imágenes de las hojas.

6.2.2 Tarea 5: Entrenar el modelo de visión por computador utilizando los datos de entrenamiento

En este apartado se realizó el entrenamiento de los modelos seleccionados en la fase anterior, los cuales son: VGG16, InceptionV3, ResNet50. Para llevar a cabo esta tarea se desarrolló el código en Python, utilizando el entorno de ejecución de Google Colaboraty, debido a los recursos computacionales que permite su GPU. De la misma manera se importaron las librerías correspondientes para efectuar el entrenamiento. A continuación, se describen algunas de las librerías más importantes para construir y entrenar el modelo como Keras, Torch, Torchvison y las librerías que permiten el análisis y visualización de datos como Numpy, Pandas, Seaborn, Matplotlib.

- **keras.models.Sequential:** Define un modelo secuencial, que es una pila lineal de capas.
- **keras.models.Model:** Proporciona una forma más flexible de definir modelos con entradas y salidas arbitrarias.
- **keras.layers.Conv2D:** Añade una capa de convolución bidimensional.
- **keras.layers.MaxPool2D:** Añade una capa de pooling (submuestreo) máximo bidimensional.
- **keras.layers.Dense:** Añade una capa densamente conectada (fully connected layer).
- **keras.layers.Flatten:** Aplana la entrada para que pueda ser alimentada a una capa densa.
- **keras.layers.Dropout:** Añade una capa de Dropout para prevenir el sobreajuste.
- **keras.layers.Input:** Define una entrada de datos para un modelo Keras.
- **keras.optimizers.Adam:** Define el optimizador Adam, que es un método de optimización estocástica adaptativa.
- **keras.callbacks.EarlyStopping:** Callback que detiene el entrenamiento cuando una métrica monitorizada ha dejado de mejorar.
- **keras.callbacks.ModelCheckpoint:** Callback que guarda el modelo después de cada epoch si la métrica monitorizada ha mejorado.
- **keras.preprocessing.image.ImageDataGenerator:** Generador de datos que permite la augmentación de imágenes en tiempo real.
- **keras.applications.imagenet_utils.preprocess_input:** Pre-procesa los datos de entrada de imágenes para que sean compatibles con los modelos pre-entrenados en ImageNet.

- **matplotlib.pyplot:** Librería para crear gráficos estáticos, animados e interactivos en Python.
- **Pandas:** Facilita estructuras de datos y herramientas intuitivas para realizar análisis de datos de manera rápida y sencilla.
- **numpy:** Librería fundamental para computación científica en Python, proporciona soporte para grandes matrices y matrices multidimensionales.
- **seaborn:** Es una biblioteca de visualización de datos construida sobre matplotlib, que ofrece una interfaz de alto nivel para crear gráficos estadísticos visualmente atractivos e informativos.
- **os:** Proporciona una manera de usar funcionalidades dependientes del sistema operativo como leer o escribir en el sistema de archivos.

Por otra parte, se establecieron los hiperparámetros de entrenamiento, que fueron aplicados a todos los modelos, mismos que se visualizan en la Tabla 12. En la Figura 28 se muestra el código donde se configuran estos valores. Fue necesario redimensionar el tamaño de cada imagen (`width_shape`, `height_shape`) a 224 x 224 píxeles, se estableció dos clases, ya que en este trabajo de titulación se tiene la clase “sana” y “tizon_tardío”, cada modelo se entrenó durante 100 épocas con un `batch_size` de 32 y para evitar el sobreajuste y ahorrar tiempo de computación, se utilizó la técnica de `early stopping`, deteniendo el entrenamiento cuando el modelo no muestra mejoras durante 30 épocas consecutivas.

Tabla 12. Especificaciones de entrenamiento

Parámetros	VGG16	ResNet50	InceptionV3
N.º de épocas	100	100	100
Batch size	32	32	32
Número Clases	2	2	2
Tamaño de imagen	224 x 224	224 x 224	224 x 224
Paciencia	30	30	30
Optimizador	adam	adadelta	adam, sgd

```
width_shape = 224
height_shape = 224
num_classes = 2
epochs = 100
batch_size = 32
patience_early_stopping = 30
```

Figura 28. Hiperparámetros de entrenamiento

A continuación, como se observa en la Figura 29, se definen las rutas a los conjuntos de datos de entrenamiento, validación y prueba que se encuentran almacenados en Google Drive. La variable “train_path” apunta a la ubicación del conjunto de datos de entrenamiento, que es usado para ajustar los parámetros del modelo. La variable “valid_path” indica la ubicación del conjunto de datos de validación, que se utiliza para evaluar el rendimiento del modelo durante el entrenamiento y ajustar hiperparámetros. Finalmente, la variable “test_path” señala la ubicación del conjunto de datos de prueba, que se emplea para evaluar el rendimiento final del modelo después del entrenamiento.

```
train_path = "/content/drive/MyDrive/Dataset1/train"
valid_path = "/content/drive/MyDrive/Dataset1/valid"
test_path = "/content/drive/MyDrive/Dataset1/test"
```

Figura 29. Especificación de rutas al conjunto de datos

Posteriormente, se configuró el generador de imágenes para los conjuntos de datos de entrenamiento y validación utilizando “ImageDataGenerator” de Keras, como se indica en la Figura 30. Los generadores aplican la función “preprocess_input” y cargan las imágenes desde los directorios especificados (train_data_dir y validation_data_dir), redimensionándolas a 224 x 224 píxeles, con un batch size de 32 y en modo categórico, lo cual permitió alimentar los datos al modelo de manera eficiente durante el entrenamiento y la validación.


```

train_datagen = ImageDataGenerator(
    preprocessing_function=preprocess_input
)

train_generator = train_datagen.flow_from_directory(
    train_data_dir,
    target_size=(width_shape, height_shape),
    batch_size=batch_size,
    class_mode='categorical'
)

validation_datagen = ImageDataGenerator(
    preprocessing_function=preprocess_input
)

validation_generator = validation_datagen.flow_from_directory(
    validation_data_dir,
    target_size=(width_shape, height_shape),
    batch_size=batch_size,
    class_mode='categorical'
)

```

Figura 30. Generador de imágenes (entrenamiento y validación)

- Configuración de los modelos CNN

- **Modelo VGG16**

Se construyó un modelo basado en la arquitectura VGG16, utilizando una entrada de imágenes con un tamaño específico y cargando pesos pre-entrenados en ImageNet.

```

1 from keras.applications.vgg16 import VGG16
2
3 # Construir el modelo
4 image_input = Input(shape=(width_shape, height_shape, 3))
5 model = VGG16(input_tensor=image_input, include_top=True, weights='imagenet')
6
7 last_layer = model.get_layer('fc2').output
8 out = Dense(num_classes, activation='softmax', name='output')(last_layer)
9 custom_model = Model(image_input, out)
10
11 # Configurar capas para entrenamiento
12 for layer in custom_model.layers[:-1]:
13     layer.trainable = False
14
15 custom_model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

```

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels.h5
553467096/553467096 [=====] - 3s 0us/step

Figura 31. Construcción y Configuración del Modelo VGG16

Se realizó la configuración del modelo utilizando el código como se puede ver en la Figura 31 en donde primeramente se definió la entrada del modelo y carga VGG16 con sus pesos pre-entrenados, incluyendo las capas superiores originales. Luego, se reemplazó la última capa completamente conectada con una nueva capa densa con activación softmax, adecuada para clasificaciones multiclase. Todas las capas preexistentes del modelo se congelan para mantener los pesos pre-entrenados, entrenando solo la nueva capa de salida. Finalmente, se compiló el modelo con la pérdida de entropía cruzada categórica y el optimizador Adam, configurando la métrica de precisión (accuracy) para evaluar el rendimiento durante el entrenamiento y la validación.

Para implementar el modelo VGG16, se comenzó descargando el modelo pre-entrenado. A continuación, en la Figura 32 se proporciona una visión general de la estructura del modelo, al igual que su configuración y conexión de las capas, luego de la ejecución del código antes mencionado.

```

Model: "model"
-----
Layer (type)                Output Shape                Param #
-----
input_1 (InputLayer)        [(None, 224, 224, 3)]      0
block1_conv1 (Conv2D)       (None, 224, 224, 64)       1792
block1_conv2 (Conv2D)       (None, 224, 224, 64)       36928
block1_pool (MaxPooling2D)  (None, 112, 112, 64)       0
block2_conv1 (Conv2D)       (None, 112, 112, 128)      73856
block2_conv2 (Conv2D)       (None, 112, 112, 128)      147584
block2_pool (MaxPooling2D)  (None, 56, 56, 128)        0
block3_conv1 (Conv2D)       (None, 56, 56, 256)        295168
block3_conv2 (Conv2D)       (None, 56, 56, 256)        590880
block3_conv3 (Conv2D)       (None, 56, 56, 256)        590880
block3_pool (MaxPooling2D)  (None, 28, 28, 256)        0
block4_conv1 (Conv2D)       (None, 28, 28, 512)        1180160
block4_conv2 (Conv2D)       (None, 28, 28, 512)        2359808
block4_conv3 (Conv2D)       (None, 28, 28, 512)        2359808
block4_pool (MaxPooling2D)  (None, 14, 14, 512)        0
block5_conv1 (Conv2D)       (None, 14, 14, 512)        2359808
block5_conv2 (Conv2D)       (None, 14, 14, 512)        2359808
block5_conv3 (Conv2D)       (None, 14, 14, 512)        2359808
block5_pool (MaxPooling2D)  (None, 7, 7, 512)          0
flatten (Flatten)           (None, 25088)              0
fc1 (Dense)                 (None, 4096)               102764544
fc2 (Dense)                 (None, 4096)               16781312
output (Dense)              (None, 2)                  8194
-----
Total params: 134268738 (512.19 MB)
Trainable params: 8194 (32.01 KB)
Non-trainable params: 134260544 (512.16 MB)

```

Figura 32. Estructura del modelo

- **Modelo InceptionV3**

Se creó un modelo utilizando la arquitectura InceptionV3 con pesos pre-entrenados en ImageNet, excluyendo la capa superior. Se configuraron todas las capas del modelo base como no entrenables para preservar los pesos pre-entrenados. Luego, se añadieron capas adicionales al modelo: una capa de Flatten para aplanar la salida del modelo base, seguida de una capa densa con 512 neuronas y activación 'relu', y una capa densa final con una activación 'softmax' para la clasificación en un número de categorías igual al número de clases en train_ds.class_indices.

Finalmente, se construyó el modelo completo combinando las entradas del modelo base con estas nuevas capas. El código utilizado para esta configuración se observa en la Figura 33.

```

def get_model():
    base_model = InceptionV3(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
    for layer in base_model.layers:
        layer.trainable = False

    x = Flatten()(base_model.output)
    x = Dense(512, activation='relu')(x)
    x = Dense(len(train_ds.class_indices), activation='softmax')(x)
    model = Model(inputs=base_model.input, outputs=x)
    return model

model = get_model()

```

Figura 33. Construcción y Configuración del Modelo InceptionV3

- **Modelo ResNet50**

Como se indica en la Figura 34, se creó un modelo basado en la arquitectura ResNet50, cargando pesos pre-entrenados en ImageNet y excluyendo la capa superior. Se añadieron capas adicionales: GlobalAveragePooling2D para reducir la dimensionalidad, seguida de una capa densa con 1024 neuronas y activación 'relu', y una capa de salida con softmax para la clasificación en num_classes categorías. Todas las capas del modelo base fueron congeladas para preservar los pesos pre-entrenados, y el modelo se compiló utilizando el optimizador Adam, la función de pérdida 'categorical_crossentropy' y la métrica de precisión.

```

base_model = ResNet50(weights='imagenet', include_top=False, input_shape=(width_shape, height_shape, 3))
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation='relu')(x)
predictions = Dense(num_classes, activation='softmax')(x)

model = Model(inputs=base_model.input, outputs=predictions)

for layer in base_model.layers:
    layer.trainable = False

model.compile(optimizer=Adam(), loss='categorical_crossentropy', metrics=['accuracy'])

```

Figura 34. Construcción y Configuración del Modelo ResNet50

Para este modelo el método de entrenamiento es transfer learning (aprendizaje por transferencia), y la técnica utilizada es feature extraction (extracción de características).

Método de entrenamiento: Transfer Learning: Permitió aprovechar el modelo pre-entrenado con el conjunto de datos de ImageNet para resolver una nueva tarea de clasificación con la hoja de tomate de riñón, evitando entrenar el modelo desde cero reduciendo así, la cantidad de tiempo de entrenamiento y el número de imágenes.

```

base_model = ResNet50(weights='imagenet', include_top=False, input_shape=(width_shape, height_shape, 3))
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation='relu')(x)
predictions = Dense(num_classes, activation='softmax')(x)

```

Figura 35. Método Transfer Learning

Técnica utilizada: Feature Extraction: Con esta técnica se congelaron todas las capas de ResNet50 y solo se entrenó las capas añadidas en la parte superior del modelo (GlobalAveragePooling2D y las capas densas), sin ajustar los pesos de las capas pre-entrenadas, lo que permite al modelo pre-entrenado actuar como un extractor de características fijas, y las nuevas capas se entrenan para realizar la clasificación.

```

for layer in base_model.layers:
    layer.trainable = False

```

Figura 36. Técnica Feature Extraction

En el entrenamiento de este modelo no se aplicó la técnica de *Fine-Tuning* ya que, este consiste en descongelar algunas capas superiores del modelo pre-entrenado y entrenarlas junto con las capas personalizadas agregadas al final del modelo. Sin embargo, debido al número limitado de imágenes y los recursos computacionales con los que se contaba, se optó por utilizar la técnica Feature Extraction donde se congelaron completamente todas las capas de Resnet50 y solo se entrenaron las nuevas capas añadidas (GlobalAveragePooling2D, Dense de 1024 neuronas y la capa de salida) para reducir el tiempo de entrenamiento. Por lo tanto, no se aplica *Fine-Tuning* en este caso.

Entrenamiento de los modelos

Los modelos VGG16, InceptionV3 y ResNet50 fueron entrenados utilizando diversas configuraciones, incluyendo ajustes en los hiperparámetros, técnicas de regularización, y diferentes estrategias de optimización que se establecieron anteriormente, y se visualizan en la Tabla 12. Las líneas de código utilizadas en el proceso de entrenamiento, así como las configuraciones específicas para cada modelo, se presentan en la Tabla 13.

Tabla 13. Aplicación de los Modelos

Modelo	Código de entrenamiento	Descripción
VGG16	<pre> history = custom_model.fit(train_generator, epochs=epochs, validation_data=validation_generator, steps_per_epoch=train_generator.samples validation_steps=validation_generator.samples callbacks=[early_stopping, model_checkpoint] </pre>	<p>El código entrena el modelo VGG16 utilizando datos del generador de entrenamiento (train_generator) durante un número especificado de épocas. Evalúa el rendimiento en datos de validación (validation_generator) y emplea early_stopping y model_checkpoint para gestionar el entrenamiento y guardar el mejor modelo.</p>
InceptionV3	<pre> history = model.fit(train_ds, validation_data=val_ds, epochs=epochs, steps_per_epoch=len(train_ds), validation_steps=len(val_ds), callbacks=[early_stopping, model_checkpoint]) </pre>	<p>El código entrena el modelo InceptionV3 con los datos de entrenamiento (train_ds) y validación (val_ds) durante un número definido de épocas (epochs). steps_per_epoch y validation_steps determinan el número de pasos por época para cada conjunto de datos. Los callbacks early_stopping y model_checkpoint gestionan el entrenamiento y guardan el mejor modelo.</p>
ResNet50	<pre> history = model.fit(train_generator, epochs=epochs, validation_data=validation_generator, steps_per_epoch=train_generator.samples validation_steps=validation_generator.samples callbacks=[early_stopping, model_checkpoint]) </pre>	<p>El código entrena el modelo utilizando train_generator para el conjunto de entrenamiento y validation_generator para el conjunto de validación durante un número especificado de épocas (epochs). El parámetro steps_per_epoch define el número de pasos por época basándose en el número de muestras del generador de entrenamiento, mientras que validation_steps hace lo mismo para el generador de validación. Los callbacks early_stopping y model_checkpoint se utilizan para gestionar el entrenamiento y guardar el modelo en su mejor estado.</p>

Estos modelos fueron entrenados utilizando la configuración de hiperparámetros descritos en la Tabla 12. El entrenamiento se realizó empleando los tres conjuntos de datos especificados en la Tabla 7. Los resultados se visualizan en valores de pérdida y precisión, en donde, una pérdida más baja indica que el modelo está haciendo predicciones más precisas y que las discrepancias entre las predicciones y los valores reales son menores, mientras que una precisión más alta indica que el modelo está haciendo un mayor número de predicciones correctas, a medida que la pérdida disminuye, la precisión tiende a aumentar, ya que el modelo se está ajustando mejor a los datos, por lo cual la selección del modelo dependió de estos resultados en cada entrenamiento. Los valores obtenidos para cada métrica se encuentran alojados en el repositorio de Drive¹²³.

Tabla 14. Comparación de rendimiento de modelos con él: Dataset1, Dataset2 y Dataset3

Conjunto de datos	Modelo	Pérdida	Precisión
Dataset1	VGG16	0.031866	0.988%
	InceptionV3	0.831641	0.952%
	ResNet50	0.013442	0.996%
Dataset2	VGG16	0.013645	0.998%
	InceptionV3	0.031614	0.989%
	ResNet50	0.017055	0.996%
Dataset3	VGG16	0.021235	0.993%
	InceptionV3	0.036071	0.985%
	ResNet50	0.010852	0.998%

¹ Código de entrenamiento con Dataset1: <https://acortar.link/Dataset1>

² Código de entrenamiento con Dataset2: <https://acortar.link/Dataset2>

³ Código de entrenamiento con Dataset3: <https://acortar.link/Dataset3>

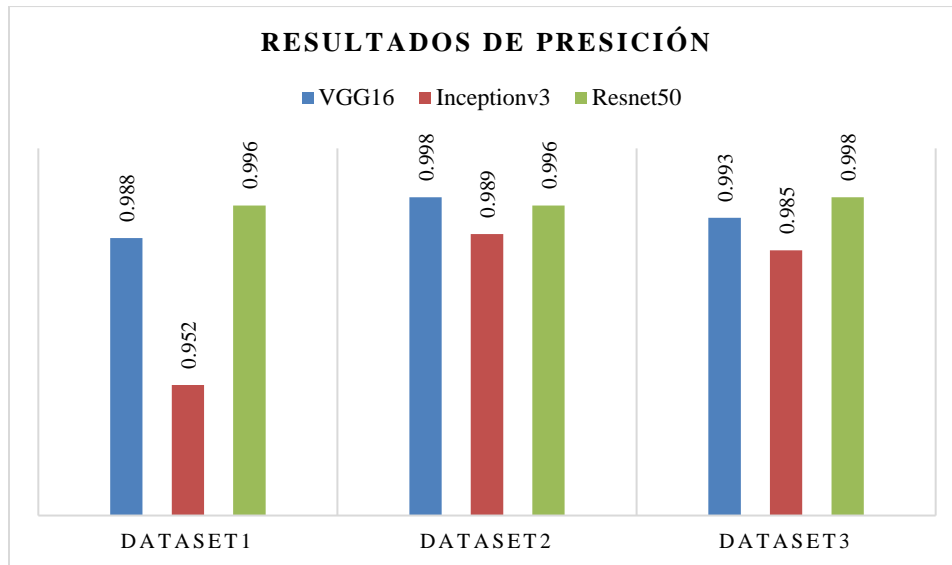


Figura 37. Gráfica que ilustra los resultados de presión

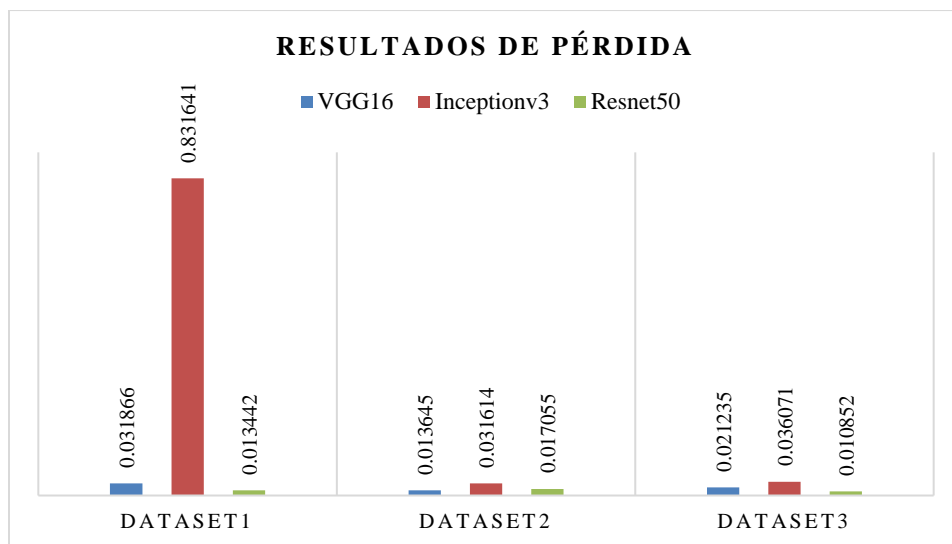


Figura 38. Gráfica que ilustra los resultados de pérdida

De acuerdo al análisis minucioso de los resultados obtenidos, considerando tanto la pérdida (Loss) como la precisión (Accuracy) se obtuvo que:

- Dataset1: El modelo ResNet50 tiene la mejor precisión (99.6%) y la menor pérdida (0.013442).
- Dataset2: El modelo VGG16 tiene la mejor precisión (99.8%) y una pérdida muy baja (0.013645).
- Dataset3: El modelo ResNet50 tiene la mejor precisión (99.8%) y la menor pérdida (0.010852).

Por lo tanto, se determina que el modelo ResNet50 entrenado con el Dataset3 presenta la mayor precisión y la menor pérdida comparativamente con las pérdidas de VGG16 con el Dataset1 y ResNet50 con el Dataset2, por lo cual se lo seleccionó para ser implementado en la siguiente fase.

Selección del Modelo para la Clasificación de Hojas de Tomate con Tizón Tardío

Tras analizar los resultados, se eligió el modelo de clasificación ResNet50, con el conjunto de datos Dataset3. Esta arquitectura permitió extraer características relevantes y detalladas de las imágenes de hojas de tomate, identificar patrones asociados con la enfermedad del tizón tardío y clasificar las hojas como sanas o enfermas de manera efectiva. En la Figura 39 se observa la organización en bloques de esta arquitectura, misma que está conformada de la siguiente manera:

- Entrada (Input Image): Se inicia con la entrada de una imagen de hoja de tomate que tiene un tamaño estándar de 224 x 224 píxeles y 3 canales de color (rojo, verde y azul).
- Extracción de Características
 - Bloque Inicial o Residual: Detecta características básicas.
 - Etapa 1 (3 x): Aprende características simples.
 - Etapa 2 (4 x): Captura características intermedias.
 - Etapa 3 (6 x): Aprende características avanzadas.
 - Etapa 4 (3 x): Detecta características de alto nivel.
 - Clasificación Final: Predice si la hoja está sana o enferma.

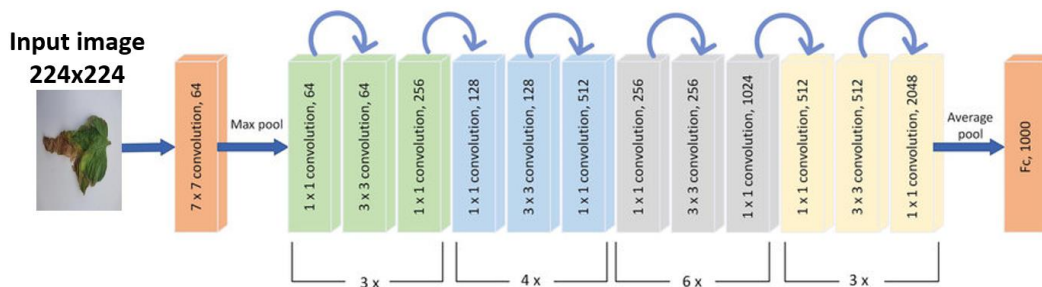


Figura 39. Arquitectura de Clasificación de Hojas de Tomate

En la Figura 39 señala el proceso de la clasificación de acuerdo a las características del modelo ResNet50. El proceso comienza al ingresar una imagen de la hoja, ajustada a un tamaño de 224 x 224 píxeles. A partir de ahí, la imagen pasa por una serie de etapas donde el modelo va aprendiendo a reconocer características clave de las hojas, gracias a los bloques residuales. Estos

bloques permiten que el modelo descubra patrones visuales más profundos, finalmente, una capa de clasificación toma toda esa información y determina si la hoja está sana o enferma.

Posteriormente, el modelo ResNet50 fue entrenado con el Dataset3, considerando distintos tamaños de batch, los resultados se evaluaron mediante las métricas de pérdida y precisión. A continuación, en la Tabla 15 se presentan los valores obtenidos en cada métrica.

Tabla 15. Resultados del modelo

Modelo	épocas	Batch size	Perdida	Precisión
ResNet50	100	8	0.026939	0.996
		16	1.589457e-08	1.00
		32	0.010852	0.998
		64	0.000268	1.00

Como se observa en la tabla anterior, tras el análisis de los datos obtenidos se observa que la pérdida más baja se alcanza con un tamaño de batch de 16, con un valor más aproximado a cero, lo que indica que las predicciones del modelo son bastante precisas y muy cerca a los valores reales. En cuanto a la métrica de precisión se obtuvo que el entrenamiento con los tamaños de batch de 16 y 64 alcanzaron una precisión de 1.00, siendo superior al tamaño de batch 32 y 8, los cuales obtuvieron 0.998 y 0.996 respectivamente, lo que quiere decir que el modelo predijo de forma correcta todas las instancias en el conjunto de validación.

En base a este análisis, se consideró que el mejor peso del modelo entrenado se obtiene con un tamaño de batch de 16, ya que no solo proporciona la pérdida más baja si no que alcanza una precisión satisfactoria de 1.00. Por lo tanto, para este conjunto de datos “Dataset3” y configuración del modelo, este tamaño de batch fue el más óptimo.

A continuación, se evaluó el modelo con los datos de prueba, que cuenta con 480 imágenes con hojas sanas y con tizón tardío, se realizó una tabla comparativa con la instancia real de la hoja y la predicha por el modelo, obteniendo un resultado óptimo, ya que el modelo logro clasificar correctamente a qué categoría correspondía cada hoja como lo indica la matriz de confusión en la Figura 40. A continuación, se presenta una parte de los resultados en la Tabla 16, sin embargo, la

documentación completa de esta comparación se encuentra alojada en un repositorio de Google Drive⁴⁵.



Figura 40. Comparación visual de imágenes del modelo ResNet50

En la Figura 41 se puede visualizar que el modelo muestra resultados elocuentemente buenos, alcanzando a una precisión del 100% tanto en el entrenamiento como en la validación, donde el modelo ha aprendido bien las características importantes de los datos y puede aplicarlas a nuevos casos sin caer en errores por sobreajuste, en la pérdida en el entrenamiento como en la validación muestran una baja constante, lo que refleja un aprendizaje bueno, en la época 64 se detecta una pequeña caída en la precisión en la validación, bajando temporalmente al 93.33%. Este comportamiento se podría dar al ruido en los datos de validación, a pesar de eso es notable cómo el modelo se recupera rápidamente, alcanzando nuevamente una precisión cercana al 100% en las épocas siguientes.

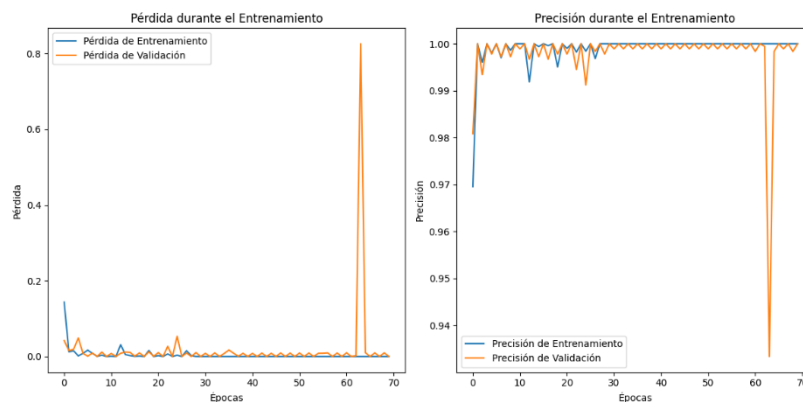


Figura 41. Curvas de precisión y pérdida durante el entrenamiento del modelo

⁴ Tabla comparativa de los resultados de Test: <https://acortar.link/oREQs4>

⁵ Resultados de clasificación del modelo: <https://acortar.link/F18jry>

Tabla 16. Resultados de clasificación: etiquetas reales vs. predicciones de ResNet50

Imágen	Etiqueta Verdadera	Predicción del modelo ResNet50
img1	Tizón Tardío	Tizón Tardío
img2	Tizón Tardío	Tizón Tardío
img3	Tizón Tardío	Tizón Tardío
img4	Sanas	Sanas
img5	Tizón Tardío	Tizón Tardío
img6	Tizón Tardío	Tizón Tardío
img7	Tizón Tardío	Tizón Tardío
img8	Sanas	Sanas
img9	Tizón Tardío	Tizón Tardío
img10	Sanas	Sanas
img11	Sanas	Sanas
img12	Tizón Tardío	Tizón Tardío
img13	Sanas	Sanas
img14	Tizón Tardío	Tizón Tardío
img15	Tizón Tardío	Tizón Tardío
img16	Sana	Sana
img17	Tizón Tardío	Tizón Tardío
img18	Tizón Tardío	Tizón Tardío
Img480	Tizón -Tardío	Tizón Tardío

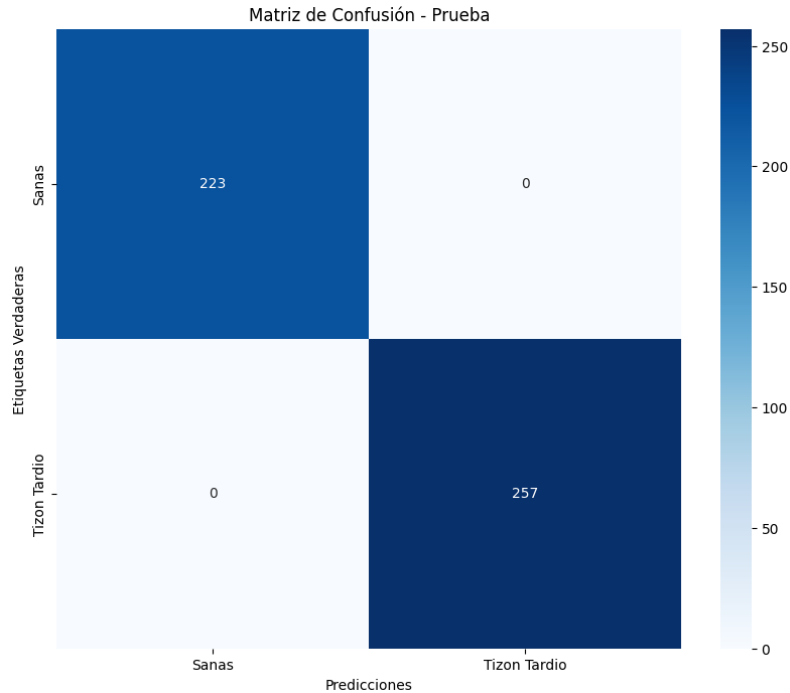


Figura 42. Matriz con los datos de test

Como parte del objetivo 2 se realizó un prototipo basado en la arquitectura ResNet50, seleccionado tras un análisis de los resultados obtenidos durante el entrenamiento con diferentes configuraciones de datos. Este prototipo fue entrenado utilizando el conjunto de datos "Dataset3", optimizado con un tamaño de batch de 16, alcanzando una precisión alta en la clasificación de imágenes de hojas sanas y enfermas. El modelo demostró una capacidad para generalizar a nuevos datos, con curvas de precisión y pérdida que demostraron un aprendizaje estable. Este prototipo logró cumplir con los objetivos del proyecto al ofrecer una herramienta para la detección automatizada de la enfermedad en las plantas de tomate, con resultados altamente precisos y reproducibles.

6.3. Objetivo 3: Validar el modelo de visión por computador en un escenario de prueba aplicada a los invernaderos del cantón Saraguro.

Fase 4: Evaluar

Esta última fase de la metodología Design Thinking permitió llevar a cabo las pruebas del modelo en un entorno real en invernaderos en el cantón Saraguro.

6.3.1 Tarea 6: Validar y contrastar los resultados obtenidos en un escenario de prueba con la opinión de un experto.

- **Aporte del Estudiante**

Para facilitar esta validación en un entorno real, se desarrolló un prototipo web básico y sencillo. Este prototipo permite a los usuarios cargar una imagen de una hoja de tomate y obtener una predicción inmediata sobre si la hoja está sana o afectada por tizón tardío. La interfaz está diseñada para ser simple y accesible, enfocándose únicamente en la funcionalidad esencial: iniciar sesión, registrar usuario, cargar la imagen y mostrar el resultado del análisis

Para garantizar que la aplicación fuera accesible y pudiera ser desplegada rápidamente, se utilizó el framework de código abierto en Python Streamlit, que permitió el desarrollo del prototipo⁶ web de forma rápida y sencilla.

- **Funcionalidades**

- a. **Interfaz intuitiva**

El prototipo web ha sido diseñada con una interfaz de usuario intuitiva y fácil de usar, permitiendo un registro de usuario previo.

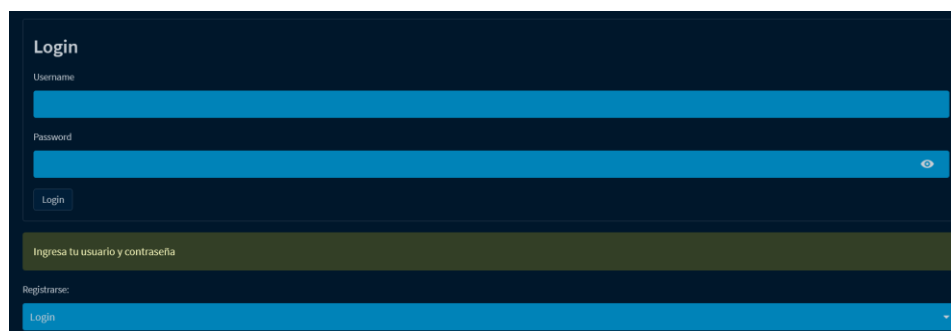


Figura 43. Pantalla acceso a la aplicación

⁶ Enlace de prototipo web: <https://hojadetectadatizontardio.streamlit.app/>



b. Visualización de diagnósticos

Una vez cargada las imágenes, se puede visualizar el diagnóstico de las mismas, si la hoja es sana entonces el prototipo indicará el texto “No es Tizón Tardío” y si la hoja presenta características de la enfermedad se mostrará el resultado “Sí es Tizón Tardío”.

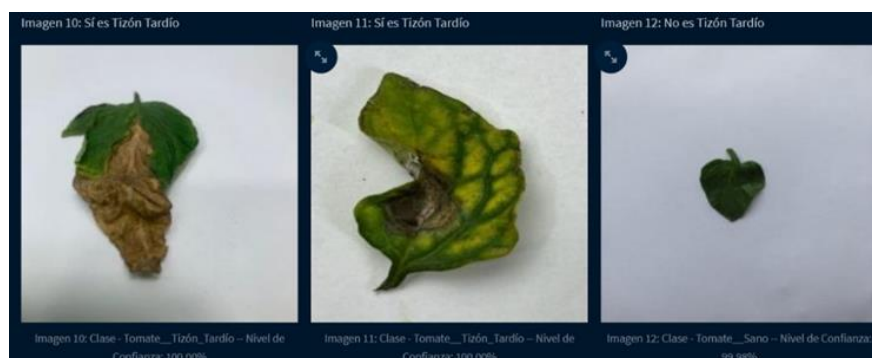


Figura 44. Visualización de diagnósticos desde la aplicación

En el **Anexo 3** se proporciona al usuario un manual que ofrece una orientación más clara sobre la utilización de esta herramienta, indicando las instrucciones para navegación en la aplicación, la interpretación de los resultados y las mejores prácticas para sacar el máximo provecho de la herramienta.

Esta tarea implica el trabajo de campo, donde se aplicará el modelo desarrollado en un escenario de prueba real en los invernaderos del cantón Saraguro. A continuación, se detallan las actividades realizadas en esta práctica de campo:



- **Captura de imágenes:** Se tomaron 100 imágenes de las hojas de tomate de riñón presentes en los invernaderos, utilizando un marco con fondo blanco, mismas que fueron clasificadas por el experto como sanas y con tizón tardío.

- **Carga en el prototipo web:** Las imágenes capturadas en los invernaderos se subieron en el prototipo web desarrollado para su diagnóstico.
- **Diagnóstico de enfermedad:** El modelo implementado en el prototipo web mostró el resultado de clasificación de la imagen de la hoja cargada, siendo sana o con tizón tardío. Estos resultados fueron comparados con el diagnóstico del experto.

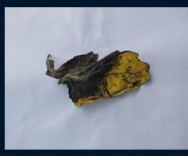





Pruebas del modelo Resnet50 en un entorno real







Una vez desarrollado el prototipo se procedió a realizar el levantamiento del conjunto de datos para las respectivas pruebas en un entorno real, por lo cual, se realizó la captura de 100 imágenes directamente en invernaderos del cantón Saraguro ubicados en los sectores Urdaneta, Tuncarta y Tenta, utilizando un marco de madera con un fondo blanco, sobre el cual se tomaron las fotografías para tener mejor visibilidad de las características de las mismas. En este proceso se contó con la colaboración del Ing. Miguel Condolo, técnico de apoyo agropecuario de la Dirección de Gestión Ambiental y Producción del GADMIS, quien en calidad de experto realizó el diagnóstico de cada hoja para posteriormente ser comparado con el resultado emitido por el modelo a través del prototipo web. En la Tabla 17, se muestran los resultados de predicción para cada imagen, las cuales se encuentran alojadas en un repositorio⁷ de Google Drive.






Tabla 17. Diagnóstico del Modelo

Nombre de la imagen	Imagen	Diagnóstico del modelo	Opinión del experto
img_1		Sí es Tizón Tardío	Tizón tardío
img_2		Sí es Tizón Tardío	Tizón tardío

⁷ Predicciones en un entorno real: <https://n9.cl/resultadosentornoreal>

img_3	<p>Imagen 3: No es Tizón Tardío</p> 	<p>No es Tizón Tardío</p>	<p>Sana</p>
img_4	<p>Imagen 4: Sí es Tizón Tardío</p> 	<p>Sí es Tizón Tardío</p>	<p>Tizón tardío</p>
img_5	<p>Imagen 5: No es Tizón Tardío</p> 	<p>No es Tizón Tardío</p>	<p>Sana</p>
img_6	<p>Imagen 6: Sí es Tizón Tardío</p> 	<p>Sí es Tizón Tardío</p>	<p>Tizón tardío</p>
img_7	<p>Imagen 7: Sí es Tizón Tardío</p> 	<p>Sí es Tizón Tardío</p>	<p>Tizón Tardío</p>
img_8	<p>Imagen 8: Sí es Tizón Tardío</p> 	<p>Sí es Tizón Tardío</p>	<p>Tizón tardío</p>
img_9	<p>Imagen 9: No es Tizón Tardío</p> 	<p>No es Tizón Tardío</p>	<p>Sana</p>

img_10	<p>Imagen 10: No es Tizón Tardío</p> 	<p>No es Tizón Tardío</p>	<p>Sana</p>
img_11	<p>Imagen 11: Sí es Tizón Tardío</p> 	<p>Sí es Tizón Tardío</p>	<p>Tizón tardío</p>
img_12	<p>Imagen 12: No es Tizón Tardío</p> 	<p>No es Tizón Tardío</p>	<p>Sana</p>
img_13	<p>Imagen 13: No es Tizón Tardío</p> 	<p>Sí es Tizón Tardío</p>	<p>Tizón tardío</p>
img_14	<p>Imagen 14: Sí es Tizón Tardío</p> 	<p>Sí es Tizón Tardío</p>	<p>Tizón Tardío</p>
img_15	<p>Imagen 15: Sí es Tizón Tardío</p> 	<p>Sí es Tizón Tardío</p>	<p>Tizón tardío</p>

img_16	<p>Imagen 16: No es Tizón Tardío</p> 	<p>No es Tizón Tardío</p>	<p>Sana</p>
img_17	<p>Imagen 17: No es Tizón Tardío</p> 	<p>No es Tizón Tardío</p>	<p>Sana</p>
img_18	<p>Imagen 18: Sí es Tizón Tardío</p> 	<p>Sí es Tizón Tardío</p>	<p>Tizón Tardío</p>
img_99	<p>Imagen 19: No es Tizón Tardío</p> 	<p>No es Tizón Tardío</p>	<p>Sana</p>
img_100	<p>Imagen 20: Sí es Tizón Tardío</p> 	<p>Sí es Tizón Tardío</p>	<p>Tizón Tardío</p>

Como se muestra en la tabla anterior, de las 100 imágenes evaluadas, 50 correspondían a hojas sanas, y todas fueron clasificadas correctamente por el modelo. Las 50 imágenes restantes correspondían a hojas con tizón tardío, de las cuales 41 fueron clasificadas correctamente, mientras que 9 fueron incorrectamente identificadas como sanas. Para calcular el porcentaje total de clasificaciones correctas se aplicó la fórmula que se indica en la Tabla 18, mediante la cual se obtuvo un resultado del 91% de precisión.

Tabla 18. Resultados de la fase de pruebas en un entorno real

Hojas	Total	Predicciones correctas	Predicciones incorrectas	Porcentaje de clasificación correcta
Sanas	50	50	0	$\frac{VP + VN}{\text{Total}}$
Enfermas	50	41	9	$\frac{41 + 50}{100}$
Total	100	91	9	= 0,91

Estos resultados demuestran que el modelo es confiable para su uso en la identificación automática del tizón tardío en hojas de tomate, sugiriendo que puede ser una herramienta útil para los agricultores en la detección de esta enfermedad en un entorno de producción agrícola.

7. Discusión

La sección de discusión se enfoca en analizar y reflexionar sobre los resultados y hallazgos obtenidos durante el desarrollo de esta investigación. A partir de los resultados obtenidos, se puede establecer lo siguiente:

7.1 Objetivo 1: Identificar métodos, técnicas basadas en inteligencia artificial y recopilar dataset con imágenes de hojas sanas y enfermas del tomate de riñón.

Para alcanzar el primer objetivo de este Trabajo de Titulación, se realizó un exhaustivo análisis de métodos y técnicas basadas en inteligencia artificial, enfocadas en el desarrollo de modelos de visión por computador para la detección y clasificación de enfermedades en hojas de plantas vegetales, con un enfoque particular en el tizón tardío en hojas de tomate de riñón. La revisión sistemática de literatura permitió identificar los enfoques más efectivos, destacando el uso predominante de Redes Neuronales Convolucionales (CNN) sobre otros métodos como Redes Neuronales Convolucionales Regionales (R-CNN) y Máquinas de Vectores de Soporte (SVM). Las CNN se destacaron por su capacidad para extraer características relevantes y aprender patrones complejos en las imágenes, lo que las convierte en la opción preferida para la tarea de clasificación de enfermedades.

En cuanto a la recopilación de datos, se exploraron diversos repositorios de imágenes, seleccionando PlantVillage como la fuente principal debido a su amplia colección de imágenes clasificadas y su pertinencia para el análisis de enfermedades en hojas de tomate. Además, se complementó este dataset con imágenes recolectadas en entornos reales, lo que permitió enriquecer el conjunto de datos con ejemplos más representativos y variados.

La integración de datos provenientes de PlantVillage junto con las imágenes obtenidas en campo resultó en un dataset robusto y diverso, esencial para el entrenamiento y evaluación de modelos de clasificación de enfermedades en hojas de tomate de riñón.

7.2 Objetivo 2: Desarrollar un modelo de visión por computador para la detección automática de la enfermedad Tizón Tardío (*Phytophthora infestans*) en las hojas del tomate de riñón.

El objetivo principal de este trabajo fue desarrollar un modelo de visión por computador capaz de detectar automáticamente la enfermedad de tizón tardío en las hojas del tomate de riñón.

Para alcanzar este objetivo, se emplearon técnicas y algoritmos de aprendizaje automático, centrados en el uso de modelos basados en Redes Neuronales Convolucionales (CNN).

En primera instancia, en base a la revisión literaria se seleccionaron cinco modelos destacados: VGG16, ResNet50, InceptionV3, GoogleNet y DenseNet121. Los modelos fueron evaluados en función de su precisión, estableciendo un umbral mínimo de 90%. Los modelos VGG16, ResNet50 e InceptionV3 sobresalieron con tasas de precisión superiores al 99%, motivo por el cual fueron elegidos para su implementación en el trabajo de titulación. Se realizó un primer experimento donde se entrenaron estos modelos con tres conjuntos de datos: Dataset1 (imágenes de PlantVillage), Dataset2 (datos recolectados), y Dataset3 (combinación de Dataset1 y Dataset2).

Los resultados demostraron que, con el Dataset3, ResNet50 alcanzó una precisión del 99.8% y una pérdida de 0.010852, lo que destaca su capacidad para ajustar correctamente las predicciones a los datos. Al comparar con los resultados con trabajos relacionados [65], ResNet50 también mostró un alto rendimiento con una precisión del 99.95%, en conjuntos de datos diferentes. Este rendimiento es consistente con otros trabajos como [66], donde VGG16 y ResNet50 alcanzaron precisiones de 99.06% y 99.00%, respectivamente. El modelo ResNet50 entrenado con Dataset3 en este estudio superó ligeramente estas cifras, lo que subraya la efectividad estos resultados se obtuvieron con una combinación de datos obtenida en los sembríos de Saraguro con los datos de PlantVillage.

A continuación, el modelo ResNet50 con el Dataset3 fue sometido a un segundo experimento, en el cual se ajustaron diferentes tamaños de lote y se fijó un número específico de épocas para el entrenamiento. Este enfoque permitió optimizar aún más el modelo, mejorando su capacidad para detectar automáticamente el tizón tardío en las hojas del tomate de riñón. En esta etapa de análisis se obtuvieron los siguientes resultados: al utilizar batch size de 8, la precisión alcanzó un 99.6% y la pérdida fue de 0.026939. Al aumentar el batch size a 16, la precisión mejoró al 100%, con una pérdida extremadamente baja de $1.589457e-08$, para un tamaño de lote de 32, la precisión fue del 99.8% y la pérdida se situó en 0.010852. Finalmente, con un tamaño de lote de 64, el modelo logró mantener una precisión del 100% con una pérdida de 0.000268. Esto indica que con un batch size de 16 el modelo tuvo un mejor desempeño, por lo que este peso fue seleccionado para la implementación en un entorno real.

7.3 Objetivo 3: Validar el modelo de visión por computador en un escenario de prueba aplicada a los invernaderos del cantón Saraguro.

El Objetivo 3 fue validar y evaluar el modelo de visión por computador desarrollado para la detección de enfermedades en las hojas de tomate de riñón en los invernaderos del cantón Saraguro. Para ello se desarrolló una aplicación web básica, a la cual se le integró el modelo entrenado, esto permitió tener una interfaz amigable para el usuario al momento de realizar el diagnóstico de la enfermedad en la hoja.

En primera instancia se recolectaron 100 nuevas imágenes en entornos reales, con la característica de que cada imagen fue tomada con un fondo blanco, para esta práctica de campo se tuvo la presencia del experto quien validó si los resultados emitidos por el modelo fueron correctos o incorrectos.

Finalmente se compararon las evaluaciones del modelo con la opinión del experto (Ver Tabla 17. Diagnóstico del Modelo) obteniendo que de 50 hojas enfermas el modelo clasificó correctamente 41, y de 50 hojas sanas el modelo clasificó de forma correcta 50, lo que representa una precisión total del 91%.

Estos resultados destacan la funcionalidad del modelo para su aplicación en entornos reales. A diferencia de los estudios revisados durante la revisión sistemática de la literatura, en los que las evaluaciones generalmente se limitan al conjunto de datos de prueba original, este trabajo fue más allá al recolectar nuevas imágenes capturadas en invernaderos locales, completamente desconocidas para el modelo. Esta estrategia permitió validar su desempeño en condiciones reales, lo que refuerza su capacidad de generalización y utilidad práctica en escenarios agrícolas.

8. Conclusiones

En base a los resultados obtenidos durante el desarrollo del presente Trabajo de Titulación se pudo concluir que:

- Los resultados obtenidos a través de la RSL muestran que las redes neuronales convolucionales (CNN) como ResNet50, VGG16 e InceptionV3 han sido altamente eficaces en la clasificación de enfermedades en hojas de plantas, logrando precisiones superiores al 90%.
- La creación de un set de datos mediante la combinación de imágenes recolectadas en los invernaderos de Saraguro y los datos recopilados en línea, fueron fundamentales para la implementación del modelo, ya que aportó diversidad y representatividad al mismo, otorgando la capacidad de generalizar mejor en distintos escenarios.
- El modelo Resnet50 en fase de entrenamiento demostró mejor rendimiento a comparación de los modelos VGG16 y InceptionV3, obteniendo una precisión de 0,99% y una pérdida de 0,0108%, por lo que se lo seleccionó para integrarlo en un prototipo web.
- La optimización del modelo ResNet50, mediante ajustes en el tamaño de lote y las épocas de entrenamiento, permitió alcanzar una precisión del 100% con una pérdida mínima, demostrando que una configuración adecuada de los parámetros mejora el rendimiento sin comprometer la precisión ni la fiabilidad del modelo.
- La aplicación del método transfer learning con la técnica de feature extraction al congelar todas las capas de ResNet50 y entrenar solo las capas finales añadidas, permitió aprovechar las características previamente aprendidas en un gran conjunto de datos, reduciendo el tiempo de entrenamiento y optimizando el rendimiento del modelo para la tarea de clasificación sin necesidad de ajustar las capas profundas.
- Las pruebas realizadas al modelo mediante la integración en el prototipo web, con 100 imágenes capturadas en invernaderos del cantón Saraguro, mostraron un total del 91% de clasificaciones correctas en relación al diagnóstico emitidos por el experto en enfermedades de plantas, lo que demuestra su eficiencia en escenarios reales.

9. Recomendaciones

A partir de los resultados obtenidos en este Trabajo de Titulación, se proponen las siguientes recomendaciones para futuras investigaciones orientadas a la aplicación de técnicas de inteligencia artificial en la agricultura, especialmente en la detección de enfermedades en cultivos:

- Actualizar y ampliar el conjunto de datos con imágenes de hojas de tomate de riñón con más diversidad de enfermedades que permitan al modelo ampliar el reconocimiento de patologías en este cultivo.
- Implementar el modelo en la nube, permitiendo su actualización continua mediante el entrenamiento con nuevos datos. Esta estrategia facilitaría la creación de versiones mejoradas del modelo.
- Integrar el modelo en una herramienta portable como una aplicación móvil, que este al alcance del usuario en todo momento.
- Evaluar el impacto en la productividad agrícola a través de estudios adicionales para evaluar cómo el uso del modelo ayuda en mejorar la productividad y la calidad de los cultivos en los invernaderos del cantón Saraguro.

10. Bibliografía

- [1] Ministerio de Agricultura y Ganadería, “Pobladores de Saraguro se benefician con la Gran Minga Agropecuaria – Ministerio de Agricultura y Ganadería.” Accessed: Jul. 16, 2024. [Online]. Available: https://n9.cl/agricultura_y_ganaderia
- [2] S. Kamoun and C. Smart, “Late Blight of Potato and Tomato in the Genomics Era,” Jul. 2005, doi: 10.1094/PD-89-0692.
- [3] D. Mora and L. Guerrero, “Industria 4.0: el reto en la ruta hacia las organizaciones digitales,” Nov. 2020, doi: 10.32719/25506641.2020.8.7.
- [4] B. García Ortega, “Industria 4.0. La cuarta revolución industrial,” May 2021, doi: <http://hdl.handle.net/10251/165996>.
- [5] M. Marroquín and A. Navarro Yáñez, “Nuevas tecnologías y agricultura 4.0: impacto en los recursos humanos de la industria agrícola en Centroamérica,” Apr. 2021, doi: <http://hdl.handle.net/11531/46846>.
- [6] S. Santos Valle and F. Josef Kienzle, “Organización de las Naciones Unidas para la Alimentación y la Agricultura,” 2021, Accessed: Jul. 16, 2024. [Online]. Available: <http://www.fao.org/publications/es>
- [7] A. Ojeda, “Plataformas tecnológicas en la Agricultura 4.0: Una mirada al desarrollo en Colombia,” Jan. 2022, doi: <http://dx.doi.org/10.17981/cesta.03.01.2022.02>.
- [8] W. G. Astudillo and O. C. Camposano, “Agricultura 4.0 tecnologías empleadas en labores agrotécnicas,” 2022, doi: <http://dspace.utb.edu.ec/handle/49000/13190>.
- [9] Y. Estrada Fortes, “Papel de los genes calcineurin b-like protein 10 (Slcbl10) y respiratory burst oxidase g (Slrboh) en la respuesta de tomate a estreses abióticos,” May 2023. doi: <http://hdl.handle.net/10201/134704>.
- [10] M. Cuichán, D. Orbe, A. Salazar, and J. Muñoz, “Guía sobre el uso de base de datos de la encuesta de superficie y producción agropecuaria continua,” Apr. 2023.
- [11] INEC and Ecuador en cifras, “Información ambiental y tecnificación agropecuaria-Módulo métodos de producción y ambiente,” Apr. 2024.
- [12] A. Yitzak et al., “Application of fertigation on tomato crop yield, Balzar-Guayas area,” Sep. 2023, Accessed: Apr. 12, 2024. [Online]. Available: <https://orcid.org/0009-0002-6601-4854>
- [13] M. Jefferson, “Prefectura fortalece desarrollo productivo en Saraguro.” Accessed: Aug. 12, 2024. [Online]. Available: <https://n9.cl/lo8ih>

- [14] L. Cardona, J. Zapata, and N. Aguirre, "Tizón tardío Koppert Ecuador." Accessed: Apr. 09, 2024. [Online]. Available: <https://n9.cl/r565t>
- [15] L. Quispe, "Asociación de tomate, maíz como alternativa de control del tizón Tardío (Phytophthora Infestans) en Serranuyoc distrito de Santa Ana Provincia de la Convención," 2019, doi: <https://repositorio.unsaac.edu.pe/handle/20.500.12918/3623>.
- [16] C. Murillo and M. Pazos, "Manejo integrado del mildiu (Phytophthora infestans L.) en el cultivo de tomate (Solanum lycopersicum L.)," 2024, doi: <http://dspace.utb.edu.ec/handle/49000/16133>.
- [17] M. del C. Hernández, "Evaluación de la actividad antifúngica in vitro de extractos totales de dos especies vegetales Marco y Molle frente a uno de los hongos patógenos causantes de Phytophthora infestans en Tomate Riñón.," Feb. 2024, doi: <https://repositorio.puce.edu.ec/handle/123456789/42835>.
- [18] M. Estol, "Metodologías ágiles y plataformas de intercambio digitales," Jun. 2020, doi: 35.272.798.
- [19] E. Bermejo, "Inteligencia artificial y pensamiento del diseño o Desing Thinking," Dec. 2017.
- [20] R. Lasse, Inteligencia artificial 101 cosas que debes saber hoy sobre nuestro futuro. 2018. doi: 39308_Inteligencia_artificial.
- [21] M. Arenas, G. Arriagada, M. Mendoza, and C. Prieto, "Una breve mirada al estado actual de la Inteligencia Artificial", Accessed: Aug. 13, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Dartmouth_workshop
- [22] M. Ramon López, "Hacia la inteligencia artificial progresos, retos y riesgos." Accessed: Jul. 22, 2018. [Online]. Available: 10.7203/metode.9.11145
- [23] F. Negre, "Técnicas y aplicaciones del Machine Learning e Inteligencia Artificial en educación." Accessed: Aug. 13, 2024. [Online]. Available: <https://doi.org/10.5944/ried.27.1.37491>
- [24] D. Hinstroza Ramírez, "El machine Learning a través de los tiempos, y los aportes a la humanidad," 2018, doi: <https://n9.cl/wujgr>.
- [25] J. Peña Lorenzo, "Aplicación de técnicas de aprendizaje profundo (deep learning) para la detección de objetos en Industria 4.0," 2020. doi: <http://uvadoc.uva.es/handle/10324/43255>.

- [26] L. Téllez Galán, “Inteligencia Artificial.” Accessed: Aug. 13, 2024. [Online]. Available: <https://lenintellezgalan.wordpress.com/>
- [27] M. Damián, “Redes Neuronales: Conceptos Básicos y Aplicaciones”, Accessed: Aug. 13, 2024. [Online]. Available: <https://n9.cl/k8qmk>
- [28] L. Jesús, “Traducción automática de Alfabeto Dactilológico a Alfabeto Convencional,” Jun. 2023.
- [29] P. Ponce, M. Pusedá Chulde, and M. Ortega Bustamante, “Segmentación de imágenes agrícolas adquiridas con dron mediante algoritmos paralelos,” vol. 4, no. 2, Apr. 2023, doi: 10.53358/ideas.v4i2.861.
- [30] C. Pineda, “Diseño de un sistema de reconocimiento automático de vehículos mediante el uso de redes neuronales profundas (DNN),” 2018.
- [31] S. Yeray, “Placa programable Spresense de Sony. Desarrollo de una aplicación de visión por computador.,” Junio 2022.
- [32] J. Pedraza Caro, “La Inteligencia Artificial en la sociedad: Explorando su Impacto Actual y los Desafíos Futuros.” Accessed: Sep. 16, 2024. [Online]. Available: <https://oa.upm.es/75068/>
- [33] L. Lanzarini et al., “Aprendizaje automático y visión por computadora,” Apr. 2019, doi: <http://sedici.unlp.edu.ar/handle/10915/76973>.
- [34] K. Moncayo, “Visión por computador para reconocimiento de malezas en cultivos de tomate riñón de invernadero, mediante redes neuronales,” 2022, doi: <https://repositorio.utn.edu.ec/handle/123456789/11567>.
- [35] N. Araujo, “Método Semisupervisado para la Clasificación Automática,” 2009.
- [36] K. Moncayo, “Visión por computador para reconocimiento de malezas en cultivos de tomate riñón de invernadero, mediante redes neuronales,” 2022.
- [37] A. Garcia Sandoval, “Aplicación del aprendizaje supervisado y no supervisado Enhanced Reader,” Article · May 2024.
- [38] R. Victor, “Aprendizaje No Supervisado en Machine Learning,” Jun 12, 2019. Accessed: Aug. 13, 2024. [Online]. Available: <https://n9.cl/8ptjh>
- [39] J. Matich, “Redes Neuronales: Conceptos Básicos y Aplicaciones”, Accessed: Aug. 13, 2024. [Online]. Available: <https://n9.cl/k8qmk>
- [40] X. Basogain, “Redes neuronales artificiales y sus aplicaciones,” 2019.

- [41] D. Jarrin and D. Vinueza, “Desarrollo de un sistema prototipo de pago de frutas a través del entrenamiento de redes neuronales artificiales convolucionales,” Jul. 2020, doi: <http://bibdigital.epn.edu.ec/handle/15000/21069>.
- [42] S. Mario, “Inspiración biológica de las redes neuronales artificiales,” Feb 13, 2018. Accessed: Aug. 13, 2024. [Online]. Available: <https://n9.cl/k9xl>
- [43] Á. Gama and F. Moncholí, “Aplicación de percepción mejorada para personas con problemas visuales usando deep learning y dispositivos vestibles,” Jun. 2020.
- [44] J. Juárez, “Clasificación de imágenes con redes profundas,” Medium. Accessed: Aug. 13, 2024. [Online]. Available: <https://n9.cl/tuch7>
- [45] P. Recuero, “Matriz de confusión.” Accessed: Aug. 13, 2024. [Online]. Available: <https://n9.cl/qndoo>
- [46] E. Godoy, “Métodos clásicos de clasificación: comparación y aplicación,” Jul. 2021, doi: <http://hdl.handle.net/11086/19768>.
- [47] J. Guillermo, “Comprensión de la Matriz de Confusión y Cómo Implementarla en Python.” Accessed: Sep. 09, 2024. [Online]. Available: <https://n9.cl/vqlyr>
- [48] C. Miguel and G. Francisco, “Aplicación de técnicas de aprendizaje por refuerzo a navegación visual,” May 2020.
- [49] D. Conde, “Inteligencia Artificial con TensorFlow para predicción de comportamientos,” 2018.
- [50] C. Cortés, “Herramientas modernas en redes neuronales: la librería Keras,” Jan. 2017.
- [51] A. Géron, “Aprende Machine Learning con Scikit-Learn, Keras y TensorFlow.” Accessed: Aug. 13, 2024. [Online]. Available: <https://n9.cl/lnwgx>
- [52] J. Zamorano, “Comparison and analysis of classification methods with scikit-learn and TensorFlow libraries in Python,” Sep. 2019, doi: <https://hdl.handle.net/10630/19037>.
- [53] F. Esteban, “Librería scikit-learn python,” 19 de Abril del 2023. Accessed: Aug. 13, 2024. [Online]. Available: <https://www.tokioschool.com/noticias/que-es-scikit-learn/>
- [54] A. Moreno Díaz, “Análisis comparativo de arquitecturas de redes neuronales para la clasificación de imágenes,” Jan. 2020.
- [55] P. Gayathri, A. Dhavileswarapu, S. Ibrahim, R. Paul, and R. Gupta, “Exploring the Potential of VGG-16 Architecture for Accurate Brain Tumor Detection Using Deep Learning,”

- Journal of Computers, Mechanical and Management, vol. 2, no. 2, Jun. 2023, doi: 10.57159/gadl.jcmm.2.2.23056.
- [56] A. Martín, “Investigación de Detección de Bordes a Radiografías con Deep Learning,” May 2023, doi: <https://oa.upm.es/74981/>.
- [57] W. Peng and T. Yumin, “Estimation of economic indicators using residual neural network resnet50,” Nov. 2019, pp. 206–209. doi: 10.1109/ICDMW.2019.00039.
- [58] M. Suvaditya, “The Annotated ResNet-50. Explaining how ResNet-50,” Aug 18, 2022. Accessed: Aug. 13, 2024. [Online]. Available: <https://n9.cl/resnet-50>
- [59] J. Caya, “Evaluación de modelos de redes neuronales convolucionales aplicado a radiografías de Tórax, para apoyar al proceso de diagnóstico de neumonía asociada al covid-19,” 2020, doi: <https://hdl.handle.net/20.500.14138/3523>.
- [60] J. Morocho Jiménez, “Red Neuronal Convolucional,” Jan. 2019, doi: <http://bibdigital.epn.edu.ec/handle/15000/19940>.
- [61] 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, 2021, doi: 10.1109/ACCESS.2021.3131002.
- [62] Z. Xiao, Y. Shi, G. Zhu, J. Xiong, and J. Wu, “Leaf Disease Detection Based on Lightweight Deep Residual Network and Attention Mechanism,” IEEE Access, vol. 11, pp. 48248–48258, 2023, doi: 10.1109/ACCESS.2023.3272985.
- [63] C. Sunil, C. Jaidhar, and N. Patil, “Cardamom Plant Disease Detection Approach Using EfficientNetV2,” IEEE Access, vol. 10, pp. 789–804, 2022, doi: 10.1109/ACCESS.2021.3138920.
- [64] E. Moupojou et al., “FieldPlant: A Dataset of Field Plant Images for Plant Disease Detection and Classification With Deep Learning,” IEEE Access, vol. 11, pp. 35398–35410, 2023, doi: 10.1109/ACCESS.2023.3263042.
- [65] L. Li, S. Zhang, and B. Wang, “Plant Disease Detection and Classification by Deep Learning - A Review,” 2021, Institute of Electrical and Electronics Engineers Inc. doi: 10.1109/ACCESS.2021.3069646.

- [66] Y. Zhao et al., “Plant Disease Detection Using Generated Leaves Based on DoubleGAN,” *IEEE/ACM Trans Comput Biol Bioinform*, vol. 19, no. 3, pp. 1817–1826, 2022, doi: 10.1109/TCBB.2021.3056683.
- [67] Y. Zhang, C. Song, and D. Zhang, “Deep Learning-Based Object Detection Improvement for Tomato Disease,” *IEEE Access*, vol. 8, pp. 56607–56614, 2020, doi: 10.1109/ACCESS.2020.2982456.
- [68] S. Sakib Hossain, M. Shakrin Jahan, and P. Osim Kumar, “PlantDet: A Robust Multi-Model Ensemble Method Based on Deep Learning For Plant Disease Detection,” *IEEE Access*, vol. 11, pp. 34846–34859, 2023, doi: 10.1109/ACCESS.2023.3264835.
- [69] A. Wani Javaid, S. Sharma, M. Muzamil, S. Ahmed, S. Sharma, and S. Singh, “Machine Learning and Deep Learning Based Computational Techniques in Automatic Agricultural Diseases Detection: Methodologies, Applications, and Challenges,” Jan. 01, 2022, Springer Science and Business Media B.V. doi: 10.1007/s11831-021-09588-5.
- [70] W. Albattah, M. Nawaz, A. Javed, M. Masood, and S. Albahli, “A novel deep learning method for detection and classification of plant diseases,” *Complex and Intelligent Systems*, vol. 8, no. 1, pp. 507–524, Feb. 2022, doi: 10.1007/s40747-021-00536-1.
- [71] S. Nandhini and K. Ashokkumar, “Improved crossover based monarch butterfly optimization for tomato leaf disease classification using convolutional neural network,” *Multimed Tools Appl*, vol. 80, no. 12, pp. 18583–18610, May 2021, doi: 10.1007/s11042-021-10599-4.
- [72] M. Nawaz et al., “A robust deep learning approach for tomato plant leaf disease localization and classification,” *Sci Rep*, vol. 12, no. 1, Dec. 2022, doi: 10.1038/s41598-022-21498-5.
- [73] M. Moussafir, H. Chaibi, R. Saadane, A. Chehri, A. El Rharras, and G. Jeon, “Design of efficient techniques for tomato leaf disease detection using genetic algorithm-based and deep neural networks,” *Plant Soil*, vol. 479, no. 1–2, pp. 251–266, Oct. 2022, doi: 10.1007/S11104-022-05513-2/METRICS.
- [74] D. Kerre and H. Muchiri, “Detecting the Simultaneous Occurrence of Strawberry Fungal Leaf Diseases with a Deep Normalized CNN,” *ACM International Conference Proceeding Series*, pp. 147–154, Mar. 2022, doi: 10.1145/3529399.3529424.

- [75] M. Ahmad, M. Abdullah, H. Moon, and D. Han, "Plant Disease Detection in Imbalanced Datasets Using Efficient Convolutional Neural Networks with Stepwise Transfer Learning," *IEEE Access*, vol. 9, pp. 140565–140580, 2021, doi: 10.1109/ACCESS.2021.3119655.
- [76] X. Wang, J. Liu, and X. Zhu, "Early real-time detection algorithm of tomato diseases and pests in the natural environment," *Plant Methods*, vol. 17, no. 1, Dec. 2021, doi: 10.1186/s13007-021-00745-2.
- [77] C. Deiana, D. Granda, and M. Sardella, "El Método científico," 2018.
- [78] Cristina Ortega, "Método analítico." Accessed: Aug. 13, 2024. [Online]. Available: <https://www.questionpro.com/blog/es/metodo-analitico/>
- [79] B. Kitchenham, "Systematic review in software engineering," pp. 1–2, Sep. 2012, doi: 10.1145/2372233.2372235.
- [80] C. Manterola, J. Rivadeneira, and T. Otzen, "The Research Question and its Association with the Levels of Evidence and Grades of Recommendation in Clinical Research Studies," *Int. J. Morphol.*, vol. 42, no. 4, pp. 1020–1028, 2024.

11. Anexos

Anexo 1. Revisión Sistemática de Literatura

Método de Barbara Kitchenham

El método de la revisión sistemática de literatura fue desarrollado por Bárbara Kitchenham, una destacada investigadora en el campo de la ingeniería de software. Este enfoque tiene como objetivo identificar, evaluar y sintetizar sistemáticamente la evidencia de estudios primarios, manteniendo la neutralidad y la reproducibilidad sobre los artículos basados en evidencia. La revisión sistemática de literatura permite obtener una perspectiva integral y orientada sobre la investigación en curso [79]. A diferencia del mapeo sistemático, que requiere un análisis más limitado de la evidencia en un área temática determinada, la revisión sistemática se concentra en responder preguntas de investigación específicas mediante un análisis integral y detallado de los estudios disponibles. Como se puede evidenciar en la Figura 45, tomada de [79] la especificación del proceso que lleva para realizar una revisión sistemática de literatura propuesto por Barbara Kitchenham.

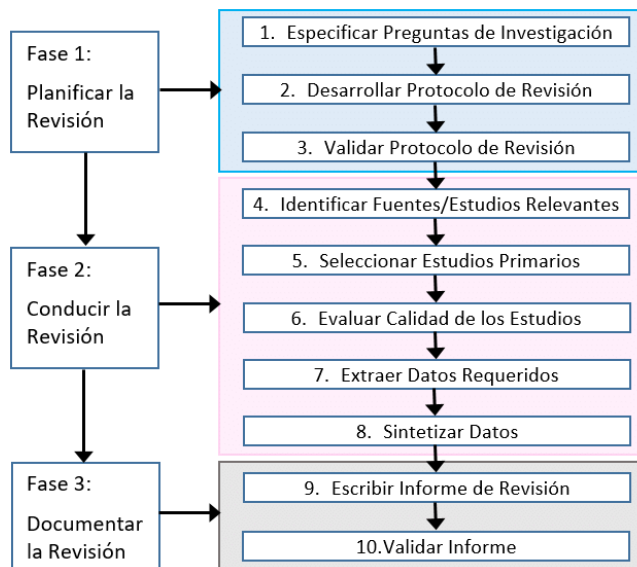


Figura 45. Enfoque de RSL de Kitchenham

1. Materiales y métodos

La presente revisión sistemática de literatura (RSL) se basó en el enfoque delineado en la metodología de Bárbara Kitchenham [79], el cual se desglosa en tres etapas esenciales:

- **Planificar la revisión**

- ✓ Especificar preguntas de investigación
- ✓ Desarrollar protocolo de revisión
- ✓ Validar protocolo de revisión
- ✓ Conducir la revisión

- **Identificar fuentes/estudios relevantes**

- ✓ Seleccionar estudios primarios
- ✓ Evaluar la calidad de los estudios
- ✓ Extraer datos requeridos
- ✓ Sintetizar datos

- **Documentar la revisión**

- ✓ Escribir informe de revisión
- ✓ Validar informe

Algunas de las tareas que se describen en estas fases no son completamente obligatorias como las siguientes:

- La puesta en marcha de una RSL, ya que depende de la revisión que se esté haciendo y sobre qué base se realice.
- Evaluar protocolo de revisión y la evaluación del informe también son opcionales, ya que depende los procedimientos que garanticen la calidad.

Después de haber visto los puntos anteriormente escritos, en la presente RSL, no se realizaron todos los pasos de la metodología de Bárbara Kitchenham, por lo que se pudo realizar las fases que más flexibilidad brinda al investigador, por lo que depende de lo que necesite y el alcance que requiera. Por consiguiente, en la Tabla 19, se presentan las fases y tareas que fueron tomadas en cuenta:

Tabla 19. Metodología de RSL adaptadas de Barbara Kitchenham

Fases	Tareas
Planificación de la revisión	Identificar necesidad de una revisión
	Especificar las preguntas
	Elaboración protocolo revisión
Conducción de la revisión	Identificar fuentes
	Seleccionar estudios relevantes
	Sintetizar datos
Revisión de informes	Elaborar informe de revisión

Resultados

2.1. Planificación de la revisión

2.1.1. Identificación de la necesidad de una revisión

Con la presente RSL, se quiere determinar técnicas y métodos utilizados para la detección de la enfermedad de Tizón Tardía en las hojas de tomate riñón, para esto se quiere la aplicación de técnicas de aprendizaje automático y de esta manera poder identificar la viabilidad de elaborar un diseño y creación de un modelo predictivo, que sea capaz de identificar la enfermedad en las hojas de tomate riñón. Todo esto se ha elaborado con el fin de poder seleccionar un método eficiente, teniendo en cuenta los recursos que se disponen para la realización de este trabajo, y también teniendo como referencia trabajos similares a esta investigación.

2.1.2. Especificación de las preguntas de investigación

En la Tabla 20 se pueden observar las interrogantes de investigación formuladas con el propósito de dirigir el progreso de la RSL, asegurando que el estudio aborde la investigación de manera encaminada. Para poder guiarse se ha elaborado una serie de preguntas de investigación, con el fin de dar un enfoque al trabajo de investigación.

Tabla 20. Preguntas de investigación el contexto de la RSL

ID	Preguntas de Investigación
PI_1	¿Cuáles son los enfoques existentes basados en Visión por Computador para identificar las enfermedades en hoja?

PI_2 ¿Qué modelos de aprendizaje automático son más efectivos para la detección temprana de enfermedades en hojas de tomate de riñón?

PI_3 ¿Cuál es la influencia del tamaño y calidad del dataset en el rendimiento de los modelos de inteligencia artificial en la identificación de hojas sanas y enfermas del tomate de riñón?

2.1.3. Desarrollo del protocolo de revisión

2.1.3.1. Estrategia de búsqueda

Para realizar la revisión sistemática de literatura de manera efectiva, es esencial contar con un protocolo detallado que planifique de antemano el proceso completo, por este motivo, se ha elegido incorporar el método PICOC, el cual es una recomendación respaldada por Petticrew y Roberts [80], autores que han desarrollado una guía práctica para llevar a cabo RSL.

- ✓ **P**opulation (Población): ¿Quién?
- ✓ **I**ntervention (Intervención): ¿Qué o cómo?
- ✓ **C**omparison (Comparación): ¿Con qué comparar?
- ✓ **O**utcomes (Resultados): ¿Qué se busca conseguir/mejorar?
- ✓ **C**ontext (Contexto): ¿En qué tipo de organización y bajo qué circunstancias?

Esta metodología se basa en cinco criterios que facilitan una definición de la cadena de búsqueda. No obstante, en esta revisión sistemática de literatura (RSL), se excluye el criterio "Comparación", ya que el enfoque se centra en identificar la intervención en lugar de compararla. Por esta razón, los criterios finales se reducen a cuatro: Población (P), Intervención (I), Resultados (O) y Contexto (C).

2.1.3.2. Fuentes bibliográficas

Para las fuentes bibliográficas se ha seleccionado las siguientes bibliotecas virtuales:

- ✓ IEEE Digital Library (<https://www.ieee.org/>)
- ✓ ACM Digital Library (<https://dl.acm.org/>)
- ✓ Springer Link (<https://link.springer.com/>)
- ✓ SciELO (<https://scielo.org/es/>)

2.1.3.3. Definir palabras claves para el problema de investigación

Se pudieron definir palabras claves, las cuales ha permitido construir las cadenas de búsqueda.

- Compute vision
- Deep learning
- Leaf diseases
- Methods
- Models
- Techniques
- Classification
- dataset
- detection

2.1.3.4. Cadenas de búsqueda

Para establecer la cadena de búsqueda se formaron a partir de las palabras claves obtenidas del enfoque PICOC. Estas expresiones se adaptaron a las particularidades de cada biblioteca digital seleccionada, las cadenas generadas para cada biblioteca se presentan en la Tabla 21.

Tabla 21. Criterios de búsqueda para las bibliotecas virtuales

Bibliotecas Virtuales	Cadenas de Búsqueda
ACM Digital Library	("leaf diseases") AND ("computer vision" OR "machine learning") AND ("methods" OR "models" OR "techniques")
IEEE Digital Library	("leaf diseases") AND ("computer vision" OR "machine learning") AND ("methods" OR "models" OR "techniques")
Scielo	("Algorithms" OR "Computer Vision" OR "Image Processing" OR "Machine Learning" OR "Convolutional Neural Networks" OR "digital Image Classification") AND ("Disease Detection" OR "Plant Disease classification" OR "Leaf Disease Detection") date: 2018-2023 WoS Áreas Temáticas: Horticultura

Science direct	("Algorithms" OR "Computer Vision" OR "Image Processing" OR "Machine Learning" OR "Convolutional Neural Networks" OR "Digital Image Classification") AND ("Disease Detection" OR "Plant Disease clasification" OR "Leaf Disease Detection")
----------------	---

2.1.3.5. Criterios de inclusión y exclusión

Se consideró lo siguiente:

- Artículos científicos publicados en revistas o congresos
- Artículos científicos o libros con un año de publicación 2019
- Artículos científicos en inglés o español
- Artículos donde usan métodos, técnicas y herramientas de IA para identificar enfermedades en las hojas de tomate de riñón.

Criterios de exclusión

Se excluyó los siguientes criterios:

- Artículos con años de menor publicación al año 2019
- Modelos que no hablen de lo relacionado con detección de objetos
- Métodos, técnicas y herramientas no entendibles y con poco contenido.

2.2. Realizar la revisión

2.2.1. Identificar la necesidad de la investigación(publicaciones/estudios)

La presente RSL tiene como objetivo responder las preguntas de investigación, la selección de cada estudio primario fue llevada a cabo mediante la comparación y validación de los resultados provenientes de diversas fuentes. Esta metodología fue empleada con el propósito de asegurar que la información presente calidad y aporte de manera pertinente a la revisión sistemática de literatura.

2.2.2. Seleccionar estudios primarios

Las investigaciones recopiladas de cada base de datos científica se encuentran registrados en la Tabla 22 estos estudios fueron sometidos a tres evaluaciones, lo cual facilitará la elección de los artículos que cumplen con los criterios de inclusión y exclusión.

Tabla 22. Estudios encontrados en la base de datos científicos

Base de datos científico	Número de estudios encontrados
IEEE Digital Library	126
ACM Digital Library	41
Springer Link	73
TOTAL	240

2.2.2.1. Análisis uno: Título y Resumen

Se realiza un análisis detallado del título y resumen de cada documento. Después de completar la revisión y análisis, se decide si la investigación se incluirá o excluirá. La Tabla 23 muestra la cantidad de documentos aprobados en la primera fase de análisis. En situaciones de duplicados en varias bases de datos, se prioriza el estudio con mayor presencia para su evaluación.

Tabla 23. Cantidad de artículos aceptados, rechazados y duplicados del primer análisis

Base de datos científico	Estudios aceptados	Estudios rechazados
IEEE Digital Library	25	101
ACM Digital Library	27	14
Springer Link	68	5
TOTAL	120	120

2.2.2.2. Análisis dos: Introducción, Métodos y Conclusiones

Una vez concluida la fase inicial de análisis, se realiza la lectura de la introducción, los métodos y las conclusiones de los artículos científicos previamente seleccionados. Al término de este procedimiento, se categorizan los informes como incluidos o excluidos. En la Tabla 24 se presenta la cantidad de documentos aprobados durante el segundo análisis. Aquellos estudios inaccesibles para el análisis de la introducción, los métodos y las conclusiones debido a su falta de disponibilidad en cualquier medio han sido desestimados.

Tabla 24. Cantidad de artículos aceptados, rechazados y duplicados del segundo análisis

Base de datos científico	Estudios Aceptados	Estudios Rechazados
IEEE Digital Library	15	10
ACM Digital Library	2	25
Springer Link	12	56
TOTAL	29	91

2.2.2.3. Análisis tres: Lectura crítica de artículos

Una vez completada la fase inicial de análisis, se realiza una evaluación crítica individual de los documentos que han sido seleccionados. Al finalizar la revisión y análisis de la información, se procede a categorizar cada artículo como aceptado o descartado. La Tabla 25 presenta la cantidad de documentos aprobados en el tercer análisis. Estos análisis han experimentado un proceso de evaluación crítica, en el que se determinó si su contenido aborda las preguntas de investigación.

Tabla 25. Cantidad de artículos aceptados, rechazados y duplicados del tercer análisis

Base de datos científico	Estudios Aceptados	Estudios Rechazados
IEEE Digital Library	9	6
ACM Digital Library	1	1
Springer Link	6	6
TOTAL	16	13

Tal como se indica en la Tabla 25, tras la implementación de los tres análisis, se logró un recuento de 16 trabajos aprobados (TA). Estos trabajos no solo cumplen con los criterios de selección, sino que también contienen información pertinente para responder las preguntas de investigación en esta revisión sistemática de literatura. La lista de estos estudios se presenta en la Tabla 26.

Tabla 26. Trabajos aprobados (TA) durante la selección de estudios

Ref.	Título	Ref.
TA01	Neural Networks for Texture Feature Extraction. Applications to Leaf Disease Classification in Precision Agriculture	[61]
TA02	Leaf Disease Detection Based on Lightweight Deep Residual Network and Attention Mechanism	[62]
TA03	Cardamom Plant Disease Detection Approach Using EfficientNetV2	[63]
TA04	FieldPlant: A Dataset of Field Plant Images for Plant Disease Detection and Classification with Deep Learning	[64]
TA05	Plant Disease Detection and Classification by Deep Learning—A Review	[65]
TA06	Plant Disease Detection Using Generated Leaves Based on DoubleGAN	[66]
TA07	Deep Learning-Based Object Detection Improvement for Tomato Disease	[67]
TA08	PlantDet: A Robust Multi-Model Ensemble Method Based on Deep Learning for Plant Disease Detection	[68]
TA09	Machine Learning and Deep Learning Based Computational Techniques in Automatic Agricultural Diseases Detection: Methodologies, Applications, and Challenges	[69]
TA10	A novel deep learning method for detection and classification of plant diseases	[70]
TA11	Improved crossover-based monarch butterfly optimization for tomato leaf disease classification using convolutional neural network	[71]
TA12	A robust deep learning approach for tomato plant leaf disease localization and classification	[72]
TA13	Design of efficient techniques for tomato leaf disease detection using genetic algorithm-based and deep neural networks	[73]

TA14	Detecting the Simultaneous Occurrence of Strawberry Fungal Leaf Diseases with a Deep Normalized CNN	[74]
TA15	Plant Disease Detection in Imbalanced Datasets Using Efficient Convolutional Neural Networks with Stepwise Transfer Learning	[75]
TA16	Early real-time detection algorithm of tomato diseases and pests in the natural environment	[76]

2.2.3. Sintetizar los datos

Tras finalizar el análisis de cada Trabajo Aceptado (TA), se extrajo información relevante acerca de los métodos, técnicas basadas en inteligencia artificial, además, se recopiló información acerca de los modelos y enfoques utilizados para obtener los conjuntos de datos. Esta fase también permitió abordar las preguntas de investigación planteadas.

- **P1 ¿Cuáles son los métodos, técnicas basadas en Visión por Computador para identificar las enfermedades en hoja?**

A continuación, se muestra la Tabla 27, se puede evidenciar que los modelos basados en Redes Neuronales Convolucionales (CNN) son los que más se utilizan en los últimos años, dentro de las técnicas para identificar enfermedades en las hojas está el aprendizaje no supervisado que de acuerdo con los artículos analizados es el más común al momento de realizar esta tarea.

Tabla 27. Métodos, técnicas basadas en Visión por Computador

Ref.	Título	Métodos/Técnicas	Año	Ref.
TA01	Neural Networks for Texture Feature Extraction. Applications to Leaf Disease Classification in Precision Agriculture	CNN, SVM	2021	[61]
TA02	Leaf Disease Detection Based on Lightweight Deep Residual Network and Attention Mechanism	CNN	2023	[62]
TA03	Cardamom Plant Disease Detection Approach Using EfficientNetV2	CNN	2022	[63]
TA04	FieldPlant: A Dataset of Field Plant Images for Plant Disease Detection and Classification with Deep Learning	CNN	2023	[64]

TA05	Plant Disease Detection and Classification by Deep Learning—A Review	CNN	2021	[65]
TA06	Plant Disease Detection Using Generated Leaves Based on DoubleGAN	CNN, RCNN	2022	[66]
TA07	Deep Learning-Based Object Detection Improvement for Tomato Disease	CNN	2020	[67]
TA08	PlantDet: A Robust Multi-Model Ensemble Method Based on Deep Learning for Plant Disease Detection	CNN	2023	[68]
TA09	Machine Learning and Deep Learning Based Computational Techniques in Automatic Agricultural Diseases Detection: Methodologies, Applications, and Challenges	CNN, RNN, SVM	2022	[69]
TA10	A novel deep learning method for detection and classification of plant diseases	CNN	2022	[70]
TA11	Improved crossover-based monarch butterfly optimization for tomato leaf disease classification using convolutional neural network	CNN	2021	[71]
TA12	A robust deep learning approach for tomato plant leaf disease localization and classification	CNN	2022	[72]
TA13	Design of efficient techniques for tomato leaf disease detection using genetic algorithm-based and deep neural networks	CNN	2022	[73]
TA14	Detecting the Simultaneous Occurrence of Strawberry Fungal Leaf Diseases with a Deep Normalized CNN	CNN	2022	[74]
TA15	Plant Disease Detection in Imbalanced Datasets Using Efficient Convolutional Neural Networks with Stepwise Transfer Learning	CNN	2021	[75]
TA16	Early real-time detection algorithm of tomato diseases and pests in the natural environment	CNN	2021	[76]

- **P2. ¿Qué modelos de aprendizaje automático son más efectivos para la detección temprana de enfermedades en hojas?**

A continuación, se muestra la Tabla 28, sobre la detección temprana de la enfermedad en hojas mediante el aprendizaje automático, de acuerdo a la investigación se puede evidenciar que se han utilizado diversos modelos con resultados significativos en la precisión.

Tabla 28. Modelos de aprendizaje automático son más efectivos para la detección temprana

TA	Ref.	Modelos	Precisión
TA1	[61]	AlexNet	89,25%
		ResNet50	89,79%
		VGG16	93,94%
TA2	[62]	GoogLeNet+InceptionV3	91,19%
		AlexNet	95,64%
		VGG16	98,74%
		Xception145	98,68%
		ResNet18	98,02%
TA3	[63]	EfficientNetV2-S	98,28%
		EfficientNetV2-L	98,26%
TA4	[64]	MobileNet	82,9%
		VGG16	80,54%
		InceptionResNetV2	81,81%
		InceptionV3	82,54%
TA5	[65]	AlexNet	96,53%
		VGG16	95,33%
		ResNet50	99,95%
TA6	[66]	VGG16	99,06%
		ResNet50	99,00%
		DenseNet121	99,22%
TA7	[67]	Faster RCNN	97,01%
		Faster RCNN-mobile	97,37%

		Faster RCNN-res101	98,54%
		InceptionResNetV2	79,39%
TA8	[68]	EfficientNetV2L	93,44%
		Xception	91,54%
		VGG16	81,22%
		VGG19	85,72%
		SVM	99,33%
TA9	[69]	AlexNet	97,22%
		VGG16	99,25%
		GoogleNet	99,69%
		ResNet50	96,27%
		GoogleNet+ InceptionV3	99,35%
TA10	[70]	MobileNet-Beta	99,85%
		EfficientNet	99,94%
		VGG16	99,98%
TA11	[71]	InceptionV3	99,97%
		GoogleNet	87,16%
TA12	[72]	ResNet-101	89,95%
		Xception	88,25%
		VGG-19	90,39%
		ResNet50	99,97%
		VGG16	97,1%
TA13	[73]	ResNet50	96,7%
		GoogleNet	88,50%
TA14	[74]	VGG 16	94,69%
		Resnet-50	88,94%
		MobileNetV3	98,39%
TA15	[75]	VGG-16	71,40%
		InceptionV3	99,91%
TA16	[76]	YOLOv3	91,81%

A continuación, se muestra la Tabla 29, la precisión de diversos modelos de aprendizaje automático aplicado a la detención.

Tabla 29. Muestra precisión de diversos modelos de aprendizaje automático

Modelo	Precisión	TA
VGG16	99,98%	TA11
ResNet50	99,97%	TA12
InceptionV3	99,91%	TA15
GoogleNet	99,69%	TA9
DenseNet121	99,22%	TA6

Tras analizar detenidamente la Tabla 29 muestra que la precisión de diversos modelos de aprendizaje automático utilizados en la detección de enfermedades en hojas de tomate de riñón, podemos identificar claramente que algunos modelos destacan por su alto rendimiento en términos de precisión. Entre estos modelos, sobresale el modelo VGG16 con una precisión impresionante del 99.06%, seguido de cerca por el modelo VGG19 con un 99.22%. Estos resultados son extraordinariamente altos y sugieren que los modelos basados en la arquitectura VGG son altamente efectivos para la detección temprana de enfermedades en hojas de tomate de riñón. No obstante, otros modelos también han demostrado ser altamente relevantes, como EfficientNetV2-S y EfficientNetV2-L, con una precisión del 98.28% y 98.26%, respectivamente, junto con GoogleNet, que alcanza un destacado 98.74% de precisión. Además, el modelo YOLOv3 se sitúa con una precisión del 91.81%, lo que lo convierte en otra opción viable para esta tarea. Mientras varios modelos han demostrado ser efectivos para la detección de enfermedades en hojas de tomate de riñón, VGG16 y VGG19 destacan con la mayor precisión, lo que los posiciona como opciones de alto rendimiento en esta área de investigación.

P3. ¿Cómo se obtuvo el dataset utilizado para llevar a cabo el entrenamiento del modelo?

A continuación, se muestra en la Tabla 30, se detalla el origen del dataset utilizado en distintas investigaciones, que, para llevar a cabo el entrenamiento de los modelos.

Tabla 30. Dataset utilizado para llevar a cabo el entrenamiento del modelo

Ref.	Título	Dataset	N.º imágenes	Clase
[61]	Convolutional Neural Networks for Texture Feature Extraction. Applications to Leaf Disease Classification in Precision Agriculture	Outex-TC-00013 PlantVillage	1360	12
[62]	Leaf Disease Detection Based on Lightweight Deep Residual Network and Attention Mechanism	PlantVillage	54,305	38
[63]	Cardamom Plant Disease Detection Approach Using EfficientNetV2	PlantVillage Cálamo Dataset 2021	56,008	41
[64]	FieldPlant: A Dataset of Field Plant Images for Plant Disease Detection and Classification with Deep Learning	PlantVillage PlantDoc FieldPlant	5,170	15
[65]	Plant Disease Detection and Classification by Deep Learning—A Review	PlantVillage Data recopilada	54309	26
[66]	Plant Disease Detection Using Generated Leaves Based on DoubleGAN	PlantVillage	31,361	15
[67]	Deep Learning-Based Object Detection Improvement for Tomato Disease	Data recopilada AIChallenger	4,178	5
[68]	PlantDet: A Robust Multi-Model Ensemble Method Based on Deep Learning for Plant Disease Detection	PlantVillage	2,710	6
[69]	Machine Learning and Deep Learning Based Computational Techniques in Automatic Agricultural Diseases Detection: Methodologies, Applications, and Challenges	PlantVillage	4,000	4

[70]	A novel deep learning method for detection and classification of plant diseases.	PlantVillage	54,306	38
[71]	Improved crossover-based monarch butterfly optimization for tomato leaf disease classification using convolutional neural network	Plant Village	6,208	4
[72]	A robust deep learning approach for tomato plant leaf disease localization and classification	PlantVillage Kaggle	54,306	10
[73]	Design of efficient techniques for tomato leaf disease detection using genetic algorithm-based and deep neural networks	PlantVillage	54,306	9
[74]	Detecting the Simultaneous Occurrence of Strawberry Fungal Leaf Diseases with a Deep Normalized CNN	Datos primarios PlantVillage	1,134	5
[75]	Plant Disease Detection in Imbalanced Datasets Using Efficient Convolutional Neural Networks with Stepwise Transfer Learning	PlantVillage ImageNet Dataset de Pimienta	99,507	24
[76]	Early real-time detection algorithm of tomato diseases and pests in the natural environment	Data recopilada	10,696	9

La mayoría de los artículos utilizaron el dataset de PlantVillage, que es una fuente ampliamente reconocida y utilizada en la comunidad de investigación agrícola y de detección de enfermedades en plantas. Además, algunos artículos también recopilaron sus propios datos para adaptarse a sus necesidades específicas de investigación. La obtención de dataset varió según el artículo, con una combinación de dataset existentes, como PlantVillage e ImageNet, y la recopilación de datos propios para abordar problemas específicos de detección de enfermedades en hojas de tomate de riñón.

2. Discusión

La Revisión Sistemática de la Literatura (RSL) llevada a cabo en esta investigación tenía como objetivo principal identificar los métodos y técnicas basados en Visión por Computador más utilizados para la detección de enfermedades en hojas de tomate de riñón. A través de esta revisión, se pretendía establecer una base sólida para la selección de la arquitectura de red neuronal que sería empleada en la posterior fase de desarrollo y entrenamiento del modelo.

La revisión sistemática de literatura arrojó resultados significativos que respaldan la elección de las Redes Neuronales Convolucionales (CNN) como la metodología más prevalente y efectiva para la clasificación de enfermedades a través de hojas de tomate de riñón. Estos resultados se encuentran respaldados por los trabajos seleccionados, en particular, el artículo TA10 y TA12.

TA10: "Nuevo método de aprendizaje profundo para la detección y clasificación de enfermedades de las plantas. "utiliza una CNN para detectar y clasificar enfermedades en plantas. El trabajo destaca el rendimiento superior de las CNN en comparación con otros métodos de detección, lo que respalda la preferencia por esta técnica.

TA12: "Un enfoque sólido de aprendizaje profundo para la localización y clasificación de enfermedades foliares de plantas de tomate" también se basa en una CNN para la detección de enfermedades en hojas de plantas de tomate. El artículo subraya la robustez y precisión de la CNN en la localización y clasificación de enfermedades en las hojas.

Este consenso en la preferencia por las CNN se relaciona directamente con las tres preguntas que guiaron esta revisión:

Primera Pregunta: ¿Cuáles son los métodos más utilizados para la detección de enfermedades en hojas de tomate de riñón?

La respuesta a esta pregunta, respaldada por TA10 y TA12, indica que las CNN son los métodos más utilizados en la detección de enfermedades en hojas de tomate de riñón.

Segunda Pregunta: ¿Cuáles son las técnicas más efectivas para la clasificación de enfermedades en hojas de tomate de riñón?

Las CNN, como se evidenció en TA10 y TA12, son altamente efectivas en la clasificación de enfermedades en hojas de tomate de riñón, lo que responde a esta pregunta.

Tercera Pregunta: ¿Cuál es la metodología preferida en la detección temprana de enfermedades en hojas de tomate de riñón?

La preferencia por las CNN, respaldada por la literatura revisada, demuestra que estas arquitecturas son la metodología preferida en la detección temprana de enfermedades en hojas de tomate de riñón.

La Revisión Sistemática de literatura confirmó que las Redes Neuronales Convolucionales (CNN) son la metodología más utilizada y efectiva en la detección y clasificación de enfermedades en hojas de tomate de riñón. Estos resultados proporcionaron una base sólida para la elección de esta técnica en el desarrollo del modelo de visión por computador en esta investigación.

3. Conclusión

A lo largo de los análisis realizados en las tablas, se han explorado diversas dimensiones de la aplicación de la visión por computador en la detección y clasificación de enfermedades en hojas de diferentes cultivos. Estos análisis han proporcionado una comprensión profunda de los enfoques, métodos, modelos y técnicas utilizados en la identificación de hojas sanas y enfermas, así como de los desafíos asociados con esta tarea. Aquí se presentan conclusiones detalladas que abarcan los aspectos más relevantes de los análisis realizados:

Los análisis han revelado una amplia gama de enfoques y técnicas utilizados en la detección de enfermedades en hojas. Desde el aprendizaje profundo (deep learning) hasta el aprendizaje supervisado y no supervisado, los investigadores han adoptado múltiples estrategias para abordar este desafío. Esto resalta la flexibilidad y la adaptabilidad de la visión por computador en entornos agrícolas.

Una de las conclusiones más significativas se refiere a la importancia del tamaño y la calidad del dataset en el rendimiento de los modelos de inteligencia artificial. Se ha demostrado que un dataset grande y de alta calidad mejora significativamente la precisión de los modelos en la identificación de hojas sanas y enfermas. Esto subraya la necesidad de recopilar y preparar dataset sólidos y representativos para el éxito de los modelos.

Los modelos basados en redes neuronales convolucionales (CNN) han demostrado ser altamente efectivos en la identificación de enfermedades en hojas. Modelos como VGG16, VGG19, ResNet50 y EfficientNetB7 han logrado tasas de precisión superiores al 95% en diferentes escenarios. Esto refuerza la capacidad de las CNN para aprender características relevantes y generalizar a diversas condiciones.

La inclusión de una variedad de datos en el dataset ha demostrado ser una estrategia efectiva para mejorar la capacidad de los modelos para enfrentar desafíos del mundo real. Los modelos

expuestos a diferentes condiciones de iluminación, ángulos y fondos han mostrado una mayor robustez y generalización.

Los análisis han utilizado la validación cruzada y la evaluación comparativa de diferentes modelos como enfoque para medir y comparar el rendimiento. Esta estrategia permite una evaluación más sólida y objetiva de los modelos, asegurando que los resultados sean confiables y generalizables.

Estos análisis tienen implicaciones importantes en la agricultura. La identificación temprana de enfermedades en hojas puede ayudar a los agricultores a tomar medidas preventivas y mejorar la producción de cultivos. Los modelos de visión por computador tienen el potencial de revolucionar la forma en que se abordan los problemas de salud de las plantas.

Los análisis realizados en las tablas han proporcionado una visión de cómo la visión por computador y la inteligencia artificial se aplican para la detección y clasificación de enfermedades en hojas de cultivos. Las conclusiones destacan la importancia de la calidad del dataset, la eficacia de los modelos y la diversidad de datos en el éxito de esta tarea. Estos conocimientos pueden tener un impacto significativo en la industria agrícola y allanar el camino para un manejo más eficiente y sostenible de los cultivos.

Anexo 2. Recolección de datos para la clasificación de tizón tardío en hojas de tomate de riñón

1. Introducción

En la descripción del proceso adquirir el conjunto de datos de hojas de tomate de riñón para la detección del tizón tardío, en el primer paso de este proceso es tomar fotografías de hojas infectadas por el tizón tardío y sanas, con el propósito de construir un dataset privado, que permita disponer de datos para su posterior entrenamiento, validación y prueba del modelo seleccionado. La recolección de estas imágenes se llevó a cabo en las localidades de Tenta, Tuncarta y Urdaneta, pertenecientes al cantón Saraguro, bajo la guía de un experto del municipio de Saraguro, quien fue el encargado de identificar y validar las hojas afectadas por la enfermedad.

2. Materiales

Cámara de teléfono móvil:

- Xiaomi Redmi Note 10
- Apertura de cámara: f/4.5
- Tamaño (Megapíxeles): 50 MP
- Procesador de móvil: MediaTek Helio G88 GPU ARM Mali-G52

Tablero de madera

- Dimensiones: 50x50 cm
- Color Fondo: Blanco

3. Procedimiento

• Identificación de Áreas de Recolección

Se identificaron y seleccionaron los huertos en las parroquias de Urdaneta, Tenta, y la comunidad de Tuncarta, pertenecientes al cantón Saraguro, para la recolección de hojas de tomate de riñón.

• Recolección de Muestras

Se recolectaron hojas de tomate de riñón con tizón tardío y hojas sanas. Las hojas se recogieron en bolsas para mantenerlas en condiciones adecuadas hasta el momento de tomar las fotos.

• Fotografía de las hojas

Cada hoja fue tomadas las fotos de manera individual, se utilizó un fondo blanco y se ajustaron las condiciones de iluminación para asegurar una captura clara y uniforme de las imágenes. Se variaron las condiciones al momento de tomar las fotos: algunas imágenes se tomaron

dentro de la casa, otras al aire libre, incluyendo los días lluviosos y bajo luz solar directo, lo que permitió tener la diversidad representativa en los datos recopilados.

- **Proceso de Revisión y Depuración**

Las imágenes capturadas fueron revisadas para verificar su calidad y claridad. Se excluyeron las imágenes que no cumplían con los criterios de calidad, como las borrosas o mal iluminadas.

- **Clasificación por Expertos**

El experto validó las imágenes para determinar si las hojas presentaban tizón tardío o eran sanas, las imágenes que no fueron aceptadas se eliminaron del conjunto de datos final.

- **Creación del Dataset**

El conjunto de datos final con las imágenes aceptadas, asegurando que servirán para la parte del entrenamiento del modelo esto permitirá entrenar el modelo y validar si las están con tizón tardío o no tienen el tizón radio.

4. Resultados

A continuación, se presenta la Tabla 31 que muestra el proceso de adquisidor y depuración de imágenes de hojas de tomate de riñón, desde la recolección inicial hasta el conjunto final validado.

Tabla 31. Proceso de adquisición y depuración de imágenes de hojas de tomate de riñón

Categoría	Imágenes iniciales	Imágenes depuradas	Imágenes finales
Hojas Sanas	849	132	717
Hojas con Tizón Tardío	905	175	730
TOTAL	1754	307	1447

- **Visita de campo y recolección de imágenes.**

Se realizaron visitas de campo con la guía del Ing. Miguel Condolo encargado del municipio del campo de la agricultura, estas visitas fueron realizadas en la parroquia Urdaneta, Tenta y la comunidad de Tuncarta para identificar y recolectar hojas de tomate de riñón con tizón tardío y hojas sanas.

- **Fotografías**

Las fotografías fueron tomadas individualmente sobre un fondo blanco para garantizar la claridad y la diversidad de las imágenes, también las imágenes se tomaron bajo diversas condiciones de iluminación para asegurar una representación completa de las características de las hojas.

- **Depuración y categorización**

Con la ayuda del experto en la agricultura el ing. Miguel Condolo, analizó las imágenes y las clasificó, donde válido si las hojas estaban afectadas por el tizón tardío o sanas, posterior a eso las imágenes fueron organizadas. Finalmente se consiguieron un total de 730 imágenes de hojas con tizón tardío y 717 imágenes de hojas sanas de tomate de riñón.

Luego las imágenes fueron procesadas y organizadas, posterior a la clasificación el conjunto de datos se denominó como “Dataset1”, para las siguientes fases de entrenamiento, validación y prueba del modelo de clasificación.

5. Conclusiones

La recolección de imágenes, tanto como la clasificación y validación realizada, ha permitido construir un dataset para su posterior utilización, en la clasificación del tizón tardío en hojas de tomate de riñón. La ayuda del experto Miguel Condolo fue esencial para identificar de las hojas afectadas.

6. Anexo

Zonas en el cantón Saraguro considerada para la presente investigación.



Figura 46. Zona de Estudio 1- de la Parroquia Urdaneta



Figura 47. Zona de Estudio 2- Parroquia de San Pablo de Tenta



Figura 48. Zona de Estudio 3- comunidad de Tuncarta

Fotográfico del proceso de adquisición de datos para la detección de tizón tardío



Figura 49. Recolección de hojas de tomate de riñón en Urdaneta



Figura 50. Recolección de hojas de tomate de riñón Tenta



Figura 51. Recolección de hojas de tomate de riñón Tuncarta



Figura 52. Organización para la Captura de Imágenes

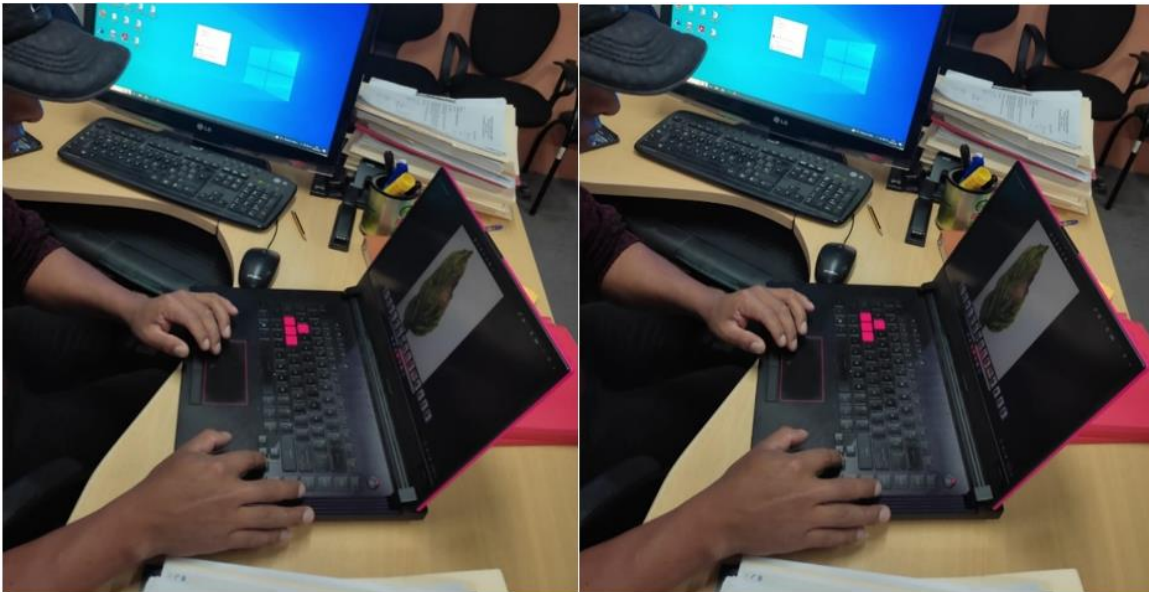


Figura 53. Clasificación de imagen con la ayuda del experto

Anexo 3. Manual de Usuario

Descripción del manual

El presente manual de usuario está diseñado para poder guiar de cómo usar del prototipo web para la detección del tizón tardío, en las hojas de tomate de riñón. A continuación, se podrá visualizar las instrucciones de como acceder y utilizar para detectar el tizón tardío.

Inicio

Bienvenid@ al manual de usuario del prototipo web diseñada para brindar acceso a los resultados del modelo de visión por computador, permitiendo monitorear la salud de tus cultivos de tomate de riñón.

Objetivo:

Proporcionar una guía detallada de cómo utilizar el prototipo web, desde la creación de una cuenta, hasta la evaluación de hojas de tomate para la detección de tizón tardío.

Requisitos Previos:

Antes de iniciar, tener en cuenta lo siguiente:

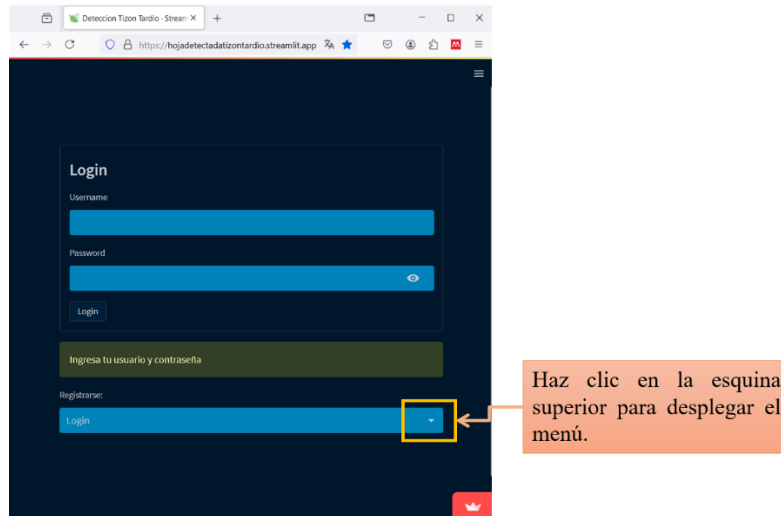
- Contar con acceso a Internet.
- Un navegador actualizado (Chrome, Firefox, etc.).

Paso 1. Acceso a la Aplicación

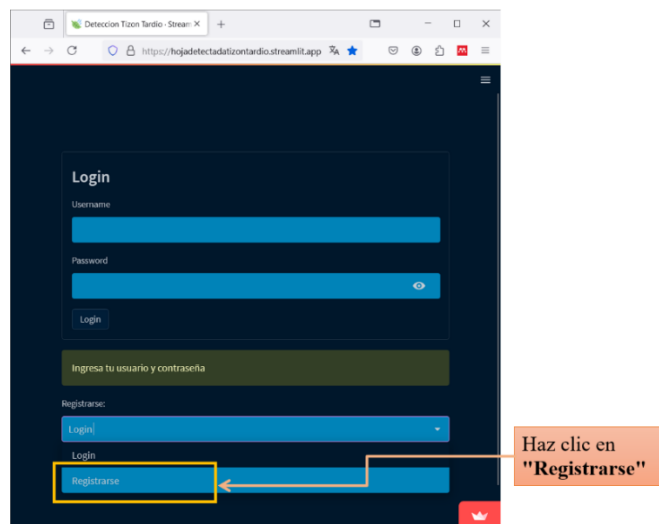
- Ingresa al prototipo web a través de tu navegador utilizando la dirección URL proporcionada: <https://hojadetectadatizontardio.streamlit.app/>.

Paso 2. Creación de cuenta

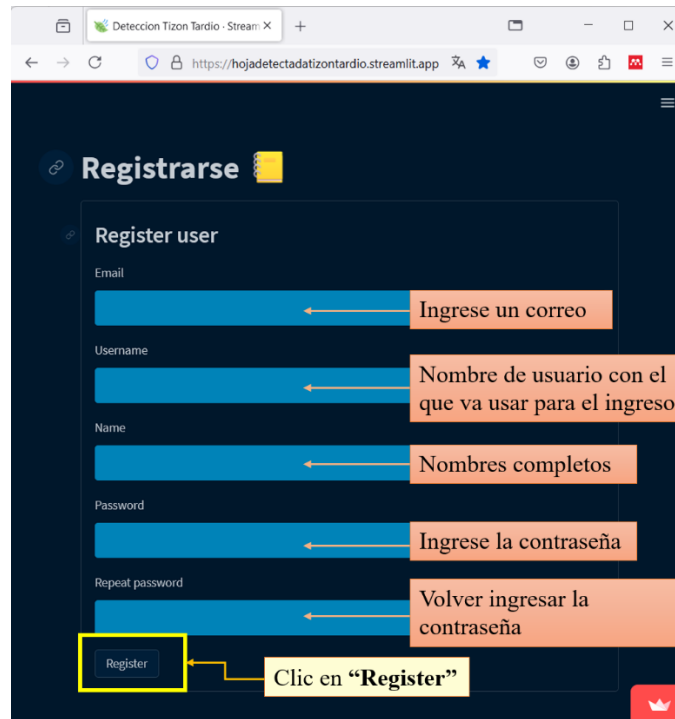
- Para poder utilizar la aplicación se tiene que realizar el registro. En la esquina superior derecha de la página de inicio, encontrarás un menú desplegable que contiene las opciones "Login" (Iniciar sesión) y "Registrarse" (Crear cuenta).



- Para poder realizar crear la cuenta haz clic en la opción "Registrarse" en el menú desplegable, una vez que selecciones "Registrarse", el formulario de registro se desplegará en la misma página.

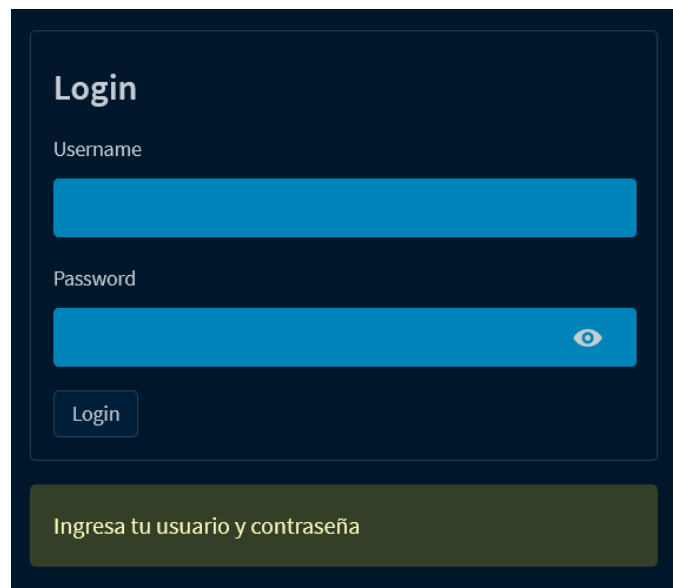


- Complete los campos del formulario de Registro como:
 - Email**, ingrese su correo electrónico.
 - Username (Nombre de Usuario)**, utilice un nombre de usuario para tu cuenta
 - Nome (Nombre)**, ingrese su nombre completo.
 - Password (Contraseña)**, utiliza una contraseña
 - Repeat Password (Repetir Contraseña)**, repite la contraseña para confirmar que coincide con la anterior.
 - Para registrarse**, haz clic el botón Register, para completar el proceso de creación de cuenta.



Paso 3. Inicio de sesión

- Para el inicio de sesión, se debe desplazar a la parte superior, ingrese las credenciales como tu nombre de usuario y contraseña.
- Para poder acceder a la aplicación, haz clic en el botón "Login" de esta manera podras iniciar sesión.



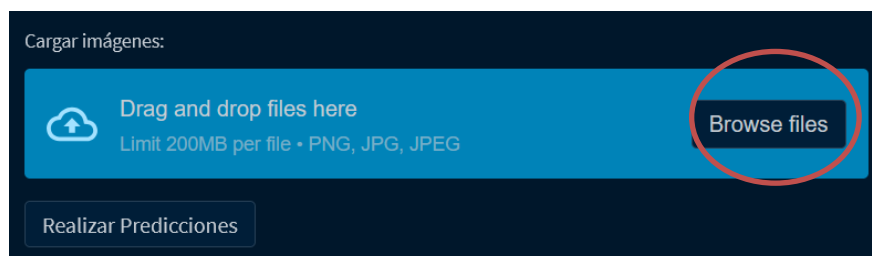
Paso 4. Interfaz de Usuario

Una vez que hayas iniciado sesión, serás dirigido a la interfaz principal de la aplicación, donde se visualiza un breve saludo, por ejemplo “Bienvenida Sisa” que confirma que ha iniciado la sección, la aplicación está diseñada para detectar señales de tizón a través del análisis de las hojas.

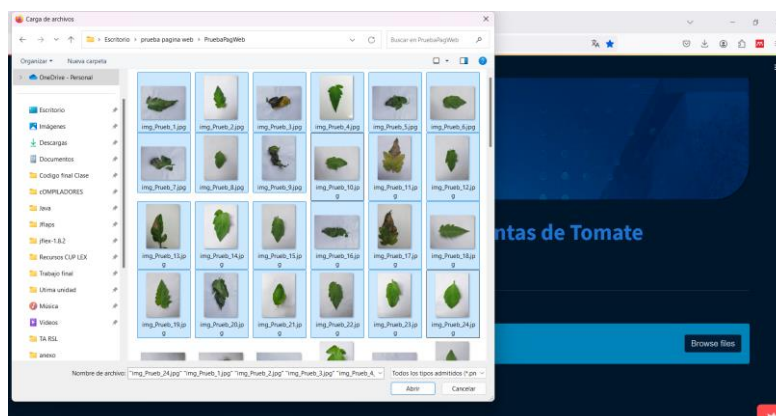


Paso 5. Carga de datos

Para cargar las hojas de tomate de riñón que deseas analizar, selecciona la opción "Browse files".



Desde allí, podrás seleccionar las imágenes que deseas enviar para su análisis. Una vez seleccionada las imágenes, la plataforma comenzará a cargar las imágenes de manera automática.



A continuación, verás una barra de progreso indicando que las imágenes se están subiendo para su análisis.

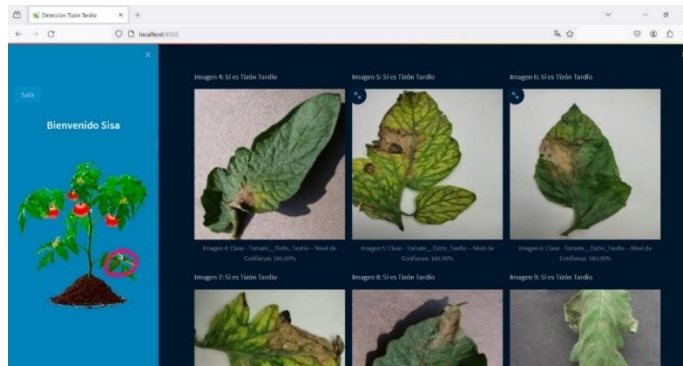


Realización de predicciones y visualización de diagnósticos

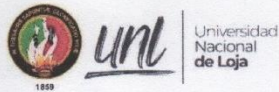
Una vez que la barra de progreso haya terminado, selecciona la opción "Realizar predicciones" para visualizar los resultados del modelo de visión por computador.



Finalmente podrás observar los diagnósticos generados por el modelo, donde se indicará si las hijas presentan tizón tardío o si están saludables.



Anexo 4. Certificado de validación del conjunto de datos por el Técnico de Apoyo Agropecuario del Gobierno Autónomo Descentralizado del Cantón Saraguro



Ing. Miguel Condolo
TECNICO DE APOYO AGROPECUARIO DE LA DIRECCIÓN DE GESTIÓN
AMBIENTAL Y PRODUCCIÓN GADMIS


CERTIFICA:

Que el **DATASET** construido por la Srta. **Sisa Ñusta Parra Tene**, de nacionalidad ecuatoriana, con cédula Nro. **1105194367**, egresada de la Carrera de **Ingeniería en Sistemas** de la Universidad Nacional de Loja, posee un total de 1,447 fotografías capturadas en tres sectores de plantaciones de tomate de riñón del cantón Saraguro, en condiciones de invernadero. Las muestras recolectadas se distribuyen en 717 hojas sanas y 730 hojas enfermas con Tizón tardío de tomate de riñón.

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


Saraguro, 9 de abril del 2023




Ing. Miguel Ángel Condolo
TECNICO DE APOYO AGROPECUARIO DE LA DIRECCIÓN DE GESTIÓN
AMBIENTAL Y PRODUCCIÓN GADMIS

C.I.: 1900474188

Anexo 5. Certificación de Evaluación del prototipo Web



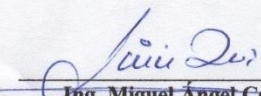
**Ing.
Miguel Ángel Condolo**
TECNICO DE APOYO AGROPECUARIO DE LA DIRECCIÓN DE GESTIÓN
AMBIENTAL Y PRODUCCIÓN GADMIS


CERTIFICA:

Que se realizaron pruebas de funcionamiento del modelo desarrollado en el trabajo de titulación de la Srta. Sisa Ñusta Parra Tene, titulado **Modelo predictivo basado en visión por computador para identificar la enfermedad del Tizón Tardío (*Phytophthora infestans*) en las hojas del tomate de riñón (*Solanum Lycopersicum*) del Cantón Saraguro**, mismo que en las pruebas experimentales funcionó satisfactoriamente, razón por la cual considero que es una herramienta muy útil para los agricultores que son parte de las escuelas agroecológicas del cantón.

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

Saraguro, 9 de abril del 2023


Ing. Miguel Ángel Condolo
TECNICO DE APOYO AGROPECUARIO DE LA DIRECCIÓN DE GESTIÓN
AMBIENTAL Y PRODUCCIÓN GADMIS



Anexo 6. Certificado de traducción del resumen al inglés



UNIDAD EDUCATIVA COMUNITARIA INTERCULTURAL BILINGÜE "RUMIÑAHUI"

Acuerdo Nro. 224-12 (20-12-2012)
Píchig - San Lucas - Loja - Ecuador

CERTIFICADO DE TRADUCCIÓN

San Lucas, 15 de noviembre de 2024

Leda. Carmen Paulina Chalán Saca

**DOCENTE DE INGLÉS EN LA UNIDAD EDUCATIVA COMUNITARIA
INTERCULTURAL BILINGÜE "RUMIÑAHUI"**

CERTIFICO:

Que he realizado una traducción minuciosa del resumen del trabajo de titulación denominado: **"Modelo predictivo basado en visión por computador para identificar la enfermedad del Tizón Tardío (*Phytophthora infestans*) en las hojas del tomate de riñón (*Solanum Lycopersicum*) del Cantón Saraguro"**, de autoría de la estudiante **Sisa Ñusta Parra Tene**, con cédula de identidad Nro. **1105194367**, perteneciente a la Carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja. La traducción al inglés cumple con las normas y características propias del idioma extranjero.

Es cuanto puedo certificar en honor a la verdad, facultado al interesado, Sisa Ñusta Parra Tene, hacer uso legal del presente, según estime conveniente.

Atentamente,

Leda. Carmen Paulina Chalán Saca

C.I: 1150174520

Nro. Reg. Senecyt: 1031-2023-2757924

LICENCIADA EN PEDAGOGÍA DEL IDIOMA INGLÉS