



Universidad
Señor de Sipán

**FACULTAD DE INGENIERÍA, ARQUITECTURA Y
URBANISMO**

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

TESIS

**Detección y clasificación de plaga pulgilla en el cultivo de
aguaymanto mediante redes neuronales**

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

Autor (es)

Bach. Burgos Vargas Dante Atilano
ORCID: <https://orcid.org/0000-0002-5684-497X>

Asesor(a)

Mg. Bances Saavedra David Enrique
ORCID: <https://orcid.org/0000-0002-7164-8918>

Línea de Investigación

Infraestructura, Tecnología y Medio Ambiente

Pimentel – Perú

2023

**DETECCIÓN Y CLASIFICACIÓN DE PLAGA PULGUILLA EN EL
CULTIVO DE AGUAYMANTO MEDIANTE REDES NEURONALES**

Aprobación del jurado

Mg. TUESTA MONTEZA VICTOR ALEXCI

Presidente del Jurado de Tesis

MG. MEJÍA CABRERA HEBER IVÁN

Secretario del Jurado de Tesis

MG. BANCES SAAVEDRA DAVID ENRIQUE

Vocal del Jurado de Tesis



DECLARACIÓN JURADA DE ORIGINALIDAD

Quien(es) suscribe(n) la DECLARACIÓN JURADA, soy(somos) egresado (s) del Programa de Estudios de **Ingeniería de Sistemas** de la Universidad Señor de Sipán S.A.C, declaro (amos) bajo juramento que soy (somos) autor(es) del trabajo titulado:

DETECCIÓN Y CLASIFICACIÓN DE PLAGA PULGUILLA EN EL CULTIVO DE AGUAYMANTO MEDIANTE REDES NEURONALES

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

En virtud de lo antes mencionado, firman:

Burgos Vargas Dante Atilano	DNI: 46999383	 
-----------------------------	---------------	---

Pimentel, 19 de diciembre de 2023.

Dedicatoria

El presente trabajo de investigación científica va dedicado con mucho aprecio y cariño a mis padres Atilano Burgos Sánchez, María Berta Vargas, hermana María Mileni Burgos Vargas y a todos mis familiares ya que gracias a su apoyo desinteresado y constante me permite realizar estudios superiores en la carrera profesional de Ingeniería de Sistemas, de esta manera llegar a concretar uno de mis objetivos y metas trazadas.

Dante Atilano

Agradecimientos

Agradecer a Dios todo poderoso por darme la capacidad, la fortaleza para seguir adelante enfrentando obstáculos en la vida diaria y en la formación como profesional, también el agradecimiento especial a mis queridos padres por apoyarme constantemente hasta lograr mis metas trazadas. A quienes gracias a sus enseñanzas, dedicación y esfuerzo especialmente a mis maestros de la Escuela de Ingeniería de Sistemas de la prestigiosa Universidad Señor de Sipán y de manera específica al Mg. Tuesta Monteza Víctor Alexci ya que con su orientación y conocimientos ha sido posible el desarrollo de esta investigación.

Índice

Dedicatoria	iv
Agradecimientos	v
Índice de Figuras	vii
Índice de Tablas	ix
Resumen	x
Abstract	xi
I. INTRODUCCIÓN	12
1.1. Realidad problemática.	12
1.2. Formulación del problema.....	13
1.3. Hipótesis.....	13
1.4. Objetivos.....	13
1.5. Teorías relacionadas al tema.....	14
II. MATERIALES Y MÉTODO	31
2.1. Tipo y Diseño de Investigación	31
2.2. Variables, Operacionalización.....	31
2.3. Población de estudio, muestra, muestreo y criterios de selección.....	33
2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad..	34
2.5. Procedimiento de análisis de datos.....	35
2.6. Criterios éticos	37
III. RESULTADOS Y DISCUSIÓN.....	39
3.1. Resultados.....	39
3.2. Discusión	42
3.3. Aporte práctico.....	43
IV. CONCLUSIONES Y RECOMENDACIONES	66
4.1. Conclusiones	66
4.2. Recomendaciones	67
REFERENCIAS	68
ANEXOS	74

Índice de Figuras

Fig. 1. Frutos de aguaymanto	16
Fig. 2. Etapas de producción del cultivo de aguaymanto.....	19
Fig. 3. Red Neuronal Artificial y sus elementos.	20
Fig. 4. Gráfica de una neurona.....	21
Fig. 5. Red Neuronal Artificial y sus elementos	22
Fig. 6. RNA Monocapa con conexiones hacia adelante	23
Fig. 7. RNA Multicapa con Conexiones hacia adelante.....	23
Fig. 8. Red Neuronal con Conexiones Hacia Atrás	26
Fig. 9. Aprendizaje Supervisado de una RNA	27
Fig. 10. Representación Cubica de los Clores Primarios y Secundarios	29
Fig. 13. Función de Membresía Trapezoidal	30
Fig. 12. Pipeline para la clasificación y entrenamiento de la red neuronal.....	44
Fig. 13. a) Vista lateral derecha del prototipo. b) Componentes del prototipo. c) Vista superior, prototipo ensamblado.....	44
Fig. 14. a) Hoja colocado en la base del prototipo para la captura de imagen. b) Imagen capturada de la hoja.....	45
Fig. 15. Hoja de aguaymanto infestado por pulguitas en estado inicial.....	46
Fig. 16. Hoja de aguaymanto ligeramente infestada por pulguita (<i>Epitrix parvula</i>).	47
Fig. 17. Hoja de aguaymanto severamente infestada por pulguita (<i>Epitrix parvula</i>).	47
Fig. 18. Distribución de Imágenes Clasificadas por Categorías.....	48
Fig. 19. Representación de hojas afectadas y no afectadas por la plaga pulguita en escala de grises. a) Hoja afectada por la plaga. b) Hoja no afectada por la plaga.....	50
Fig. 20. Resultados de la segmentación utilizando el umbral de Otsu. a) Hoja afectada por la plaga. b) Hoja no afectada por la plaga.....	52
Fig. 21. Resultados de Pruebas con Diferentes Kernels en la Segmentación.	54
Fig. 22. Resultados de la segmentación utilizando la morfología matemática. a) Hoja afectada por la plaga. b) Hoja no afectada por la plaga.	55

Fig. 23. Resultados de la detección de contorno con el recorte de la región de la hoja. a) Hoja afectada por la plaga. b) Hoja no afectada por la plaga. 56

Fig. 24. Entrenamiento de la red neuronal. 58

Índice de Tablas

Tabla 01. Modelos de color y su clasificación.	28
Tabla 02. Matriz de operacionalización de las variables de la investigación.	32
Tabla 03: Grupo redes neuronales para la clasificación de imágenes.	33
Tabla 04. Selección de redes neuronales para la clasificación de imágenes en la presente investigación.	33
Tabla 05. Resultados de la clasificación de imágenes.	40
Tabla 06. Resumen de distribución de imágenes de la base de datos por carpeta.	48
Tabla 07. Implementación del Umbral de Otsu en Python.	52
Tabla 08. Se muestra el código de la función que realiza el preprocesamiento para preparar el conjunto de entrenamiento y validación.	57
Tabla 09. Lista de Librerías exportadas en el editor Jupyter del IDE Anaconda.	58
Tabla 10. Arquitectura de la red neuronal propio.	59
Tabla 11. Se muestra el código sobre el procesamiento de los datos para alimentar a la red neuronal.	63
Tabla 12. Número de pasos por épocas para el conjunto de entrenamiento y validación. ..	65

Resumen

En la agricultura, el control de plagas en cultivos de aguaymanto requiere prácticas que eviten su destrucción y propagación. Sin embargo, el uso de drones o la inspección visual manual, resultan insuficientes. Los drones pueden monitorear el riego eficientemente, pero no se acercan lo suficiente a las plantas y la inspección visual manual tiene limitaciones en la detección de plagas. Con el objetivo de tomar acciones correctivas a tiempo y mejorar el rendimiento de las cosechas, se propone implementar redes neuronales convolucionales y otros tipos de redes neuronales. Se emplearon técnicas de visión artificial para procesar y analizar imágenes obtenidas al recorrer los cultivos y tomar fotografías de cada planta, generando una base de datos. La metodología de la investigación consistió en la adquisición y preprocesamiento de imágenes, la segmentación, la extracción de características y la clasificación utilizando redes neuronales convolucionales y profundas. Los resultados muestran que el tiempo de respuesta fue de 129 segundos utilizando redes neuronales convolucionales y 190 segundos utilizando una red neuronal profunda. En cuanto a la clasificación, se logró una precisión del 67% con la red neuronal convolucional y del 99% con la red neuronal profunda, utilizando una base de datos de 1666 imágenes en ambientes controlados. En conclusión, se evaluaron algoritmos de visión artificial, utilizando redes neuronales como clasificadores, para el reconocimiento de la plaga Pulguilla en el cultivo de aguaymanto. Los resultados demuestran que la Red Neuronal Profunda obtuvo la mayor precisión, alcanzando un 88% en la clasificación de imágenes.

Palabras Clave: Visión artificial, algoritmos, procesamiento de imágenes.

Abstract

In agriculture, pest control in aguaymanto crops requires practices that prevent their destruction and spread. However, the use of drones or manual visual inspection is insufficient. Drones can monitor irrigation efficiently, but they do not get close enough to the plants and manual visual inspection has limitations in detecting pests. In order to take corrective actions in time and improve crop yields, it is proposed to implement convolutional neural networks and other types of neural networks. Artificial vision techniques were used to process and analyze images obtained by touring the crops and taking photographs of each plant, generating a database. The research methodology consisted of image acquisition and preprocessing, segmentation, feature extraction and classification using convolutional and deep neural networks. The results show that the response time was 129 seconds using convolutional neural networks and 190 seconds using a deep neural network. Regarding classification, an accuracy of 67% was achieved with the convolutional neural network and 99% with the deep neural network, using a database of 1666 images in controlled environments. In conclusion, artificial vision algorithms were evaluated, using neural networks as classifiers, for the recognition of the Flea pest in the aguaymanto crop. The results show that the Deep Neural Network obtained the highest accuracy, reaching 99% in image classification.

Keywords: Artificial vision, algorithms, image processing.

I. INTRODUCCIÓN

1.1. Realidad problemática.

La realidad problemática actual muestra un aumento en el número de personas mal alimentadas, llegando a aproximadamente 821 millones en 2017 y 804 millones en 2016, según la FAO [1]. Aunque el Continente Africano presenta la mayor cantidad de personas subalimentadas, América del Sur también muestra un aumento en la desnutrición, pasando del 4,7% en 2014 al 5% en 2017. La industria alimentaria y su producción enfrentan problemas de competitividad, afectando principalmente a los pequeños agricultores. Además, las variaciones climáticas y otros factores han ocasionado pérdidas significativas en la agricultura y el Perú es considerablemente dependiente de la importación de productos alimenticios, como aceites, cereales, trigo, maíz, leche y azúcar [1].

Según estudios del INEI en el año 2020, alrededor del 34% de las familias peruanas (aproximadamente 2,3 millones de personas) su principal fuente de ingreso es la agricultura, generando alrededor del 8% del producto interno bruto. La mayoría de estas familias se encuentran en la región sierra, donde una proporción significativa se encuentra en situación de pobreza y extrema pobreza [2].

El Perú cuenta con tierras de alta fertilidad y una diversidad de frutos, incluyendo alrededor de 623 especies, algunas de ellas únicas en el mundo. El cultivo del aguaymanto es especialmente relevante en varias zonas rurales, pero a menudo se ve afectado por plagas y enfermedades, lo que resulta en pérdidas económicas para los agricultores. La detección de plagas se realiza actualmente de forma manual, lo que requiere mucho tiempo y conocimientos especializados para una detección temprana y una aplicación adecuada de insecticidas [3].

El procesamiento de imágenes a través de técnicas de automatización se ha desarrollado para mejorar la detección de plagas en las plantas, proporcionando una alta precisión y reduciendo la carga de trabajo en el monitoreo de grandes cultivos. La extracción

de características a partir de la segmentación de imágenes es fundamental para su uso posterior. Los métodos utilizados para la clasificación incluyen redes neuronales artificiales y vectores de soporte máquinas, los cuales han demostrado buenos resultados [3].

En resumen, la detección temprana de plagas en plantas mediante el procesamiento de imágenes es una herramienta beneficiosa para mejorar el rendimiento de los cultivos. La extracción de características y la clasificación mediante métodos como las redes neuronales y los vectores de soporte máquinas son importantes para este propósito.

1.2. Formulación del problema

¿Qué tipo de red neuronal tendrá el mejor rendimiento en el reconocimiento en la plaga pulguilla en el cultivo de aguaymanto?

1.3. Hipótesis

Basado en estudios previos sobre detección de plagas en cultivos agrícolas, se plantea que el algoritmo de redes neuronales convolucionales (CNN) presentará un mejor rendimiento en el reconocimiento de la plaga pulguilla en el cultivo de aguaymanto.

1.4. Objetivos

Objetivo general

Desarrollar y evaluar un sistema basado en redes neuronales para el reconocimiento de la plaga pulguilla en el cultivo de aguaymanto.

Objetivos específicos

- Establecer un protocolo de adquisición de imágenes que garantice la obtención de muestras representativas de hojas de aguaymanto infestadas y no infestadas por la plaga pulguilla.
- Determinar las características visuales y texturales más relevantes para el

reconocimiento de la plaga pulguilla en las imágenes de las hojas de aguaymanto.

- Realizar el pre-procesamiento de las imágenes mediante algoritmos de mejora de contraste, reducción de ruido y segmentación para resaltar las características relevantes de la plaga pulguilla.
- Elaborar un dataset que contenga un conjunto diverso de imágenes de hojas de aguaymanto con anotaciones de la presencia o ausencia de la plaga pulguilla, dividido en conjuntos de entrenamiento, validación y pruebas.
- Implementar algoritmos clasificadores basados en redes neuronales, incluyendo redes neuronales convolucionales, en un lenguaje de programación adecuado para realizar el entrenamiento y la clasificación de las imágenes del dataset.
- Evaluar el desempeño de los algoritmos clasificadores mediante métricas como precisión, sensibilidad, especificidad y exactitud, utilizando el conjunto de pruebas del dataset.
- Realizar un análisis comparativo de los resultados obtenidos por los diferentes algoritmos clasificadores implementados y determinar cuál proporciona el mejor rendimiento en el reconocimiento de la plaga pulguilla en el cultivo de aguaymanto.

1.5. Teorías relacionadas al tema

1.5.1. Inteligencia artificial.

La inteligencia artificial, según Rouhiainen [4], hace referencia a la capacidad de las máquinas para aprender de los datos y tomar decisiones utilizando algoritmos, de manera similar a cómo lo hacen las personas. La principal diferencia entre las máquinas y las personas radica en que los dispositivos de inteligencia artificial pueden analizar grandes cantidades de información simultáneamente y cometen menos errores en comparación con las tareas realizadas por los seres humanos.

1.5.2. Visión artificial.

Según Ikusmen [5], la visión artificial es la combinación de software y hardware que se utiliza para realizar tareas de adquisición y procesamiento de imágenes, y para guiar y operar dispositivos. Existen diferentes enfoques de visión artificial, como los orientados a la industria, los académicos/educativos y los gubernamentales/militares. Los sistemas de visión artificial orientados a la industria suelen requerir una mayor capacidad y tienen un costo menor en comparación con los enfoques educativos y los aplicados en el ámbito militar y gubernamental.

1.5.3. Procesamiento de Imágenes

Rouhiainen [4] explica que las personas pueden identificar características al observar una imagen, mientras que una computadora realiza un preprocesamiento de las imágenes para eliminar cualquier factor no deseado que pueda generar resultados incorrectos, como ruido, mala calidad o desenfoque. El preprocesamiento de imágenes, también conocido como procesamiento de bajo nivel, se utiliza para minimizar el ruido y mejorar la calidad de las imágenes. En este proceso, no se requiere necesariamente inteligencia artificial, ya que se trata de técnicas de procesamiento de imágenes. Es importante destacar que el resultado del preprocesamiento son imágenes mejoradas.

1.5.4. Algoritmo.

Un algoritmo es un conjunto de instrucciones predefinidas y ordenadas sin ambigüedad que permite resolver un problema específico. La Real Academia de la Lengua Española [6], este término es comúnmente utilizado en el ámbito informático para procesar datos y realizar tareas lógicas desde un estado inicial hasta la conclusión, resolviendo el problema en cuestión.

1.5.5. Definición de fruta.

Según el Ministerio de Agricultura [7], la fruta se refiere a un tipo de fruto alimenticio dulce que presenta una variedad de formas, sabores, colores, consistencias y orígenes. Las frutas son bajas en proteínas y grasas, pero ricas en azúcares, como glucosa, sacarosa y fructosa, así como en almidón, pectina y celulosa.

1.5.6. Aguaymanto (*Physalis peruviana* L.)

El aguaymanto es una fruta apreciada como exótica en diferentes partes del mundo y es originaria de los Andes peruanos. En los últimos 5 años, se ha cultivado comercialmente en Perú, mientras que en Colombia se cultiva desde hace más de 20 años. Colombia ha liderado las exportaciones de aguaymanto a nivel mundial, lo cual sirve como ejemplo y una lección sobre la importancia de estos cultivos en los Andes peruanos [7].



Fig. 1. Frutos de aguaymanto. [25]

Variabilidad Genética

Según Guerrero, Sandoval, Coronado, Rodríguez y Saavedra [8], en lugar de variedades, existen ecotipos del aguaymanto, que reciben diferentes nombres en países como Colombia, Kenia y Sudáfrica. Estos ecotipos presentan diferencias en cuanto al peso, color y tamaño de los frutos. En Colombia, por ejemplo, se han registrado frutos de hasta 4.5 gramos de peso y plantas con una altura de 92 cm y hojas pequeñas. En

cambio, los ecotipos de Kenia han llegado a tener frutos de 8 gramos de peso y plantas con una altura de 45 cm y hojas grandes. Según el MINAGRI [9], el aguaymanto es una planta de tipo arbustiva con tallos quebradizos de color verde y textura suave al tacto. Su raíz es fibrosa y puede alcanzar hasta 60 cm de profundidad en el suelo. Las hojas son enteras, de forma parecida a un corazón, pubescentes y dispuestas de forma alterna.

Morfología

El aguaymanto, de acuerdo con (Guerrero, Sandoval, Coronado, Rodriguez, & Saavedra [8]. crece hasta alcanzar una altura de 45 a 90 cm. Presenta un tallo vertical con pocas ramas y de forma tubular. Las raíces encontradas pueden tener una longitud de 50 a 80 cm. Las flores son erectas, las hojas son alternas y tienen una base cortada. El ovario es de color verde y los frutos pueden contener de 100 a 300 semillas amarillas. La parte interna del fruto es similar a la de un tomate pequeño y presenta colores que varían entre el amarillo ocre y el naranja, con un sabor ácido y agrio.

Plagas en el cultivo de aguaymanto.

Según MINAGRI, [9], el aguaymanto (*Physalis*) es una planta que es atacado por muchas plagas como prevención se debe realizar labores agrícolas como el tutoreo que ha dado buenos resultados. Una vez realizado la cosecha hay que desechar las frutas afectados por plagas.

a. Pulguillas (*Epitrix* sp.) o Pulguilla saltona.

Este insecto llega a medir 2 mm de largo, de color negro, se traslada saltando de planta en planta; se nutre haciendo pequeños huecos en las hojas reduciendo el follaje de las plantas. Esta plaga puede atacar en diferentes estados de desarrollo, pero su mayor presencia es después de haber realizado el trasplante [9].

b. Perforador del fruto (*Heliothis subflexa*).

Esta plaga se desarrolla por diferentes etapas causando mayor daño cuando es una larva verde luego es de color marrón con rayas alargados, de adulta es una polilla gris que se esconde en las hojas, los adultos depositan sus huevos cualquier parte de la planta, estos eclosionan perforando el capuchón y comen los frutos verdes, el daño se muestra cuando perforan los frutos para irse a otros frutos [9].

c. Mosca blanca (*Trialeuroides vaporatum*).

Esta es la plaga más común en el cultivo del aguaymanto. Los insectos de esta especie se encuentran en el envés de las hojas en diferentes etapas, desde huevos hasta adultos. Cuando se alimentan de la savia de la planta, pueden transmitir virus. Se recomienda utilizar preparados con ajo, rocoto y jabón para controlar esta plaga. También se pueden colocar trampas amarillas en diferentes lugares de las plantaciones, como banderolas hechas de plástico amarillo impregnadas con aceite comestible o melaza.

d. Pulgones (*Aphis* sp.).

Estos insectos causan daño principalmente en las hojas y en el capuchón del aguaymanto, lo que afecta la calidad del fruto y genera pérdidas económicas para los agricultores [9].

1.5.7. Etapas de producción agrícola del cultivo de aguaymanto.

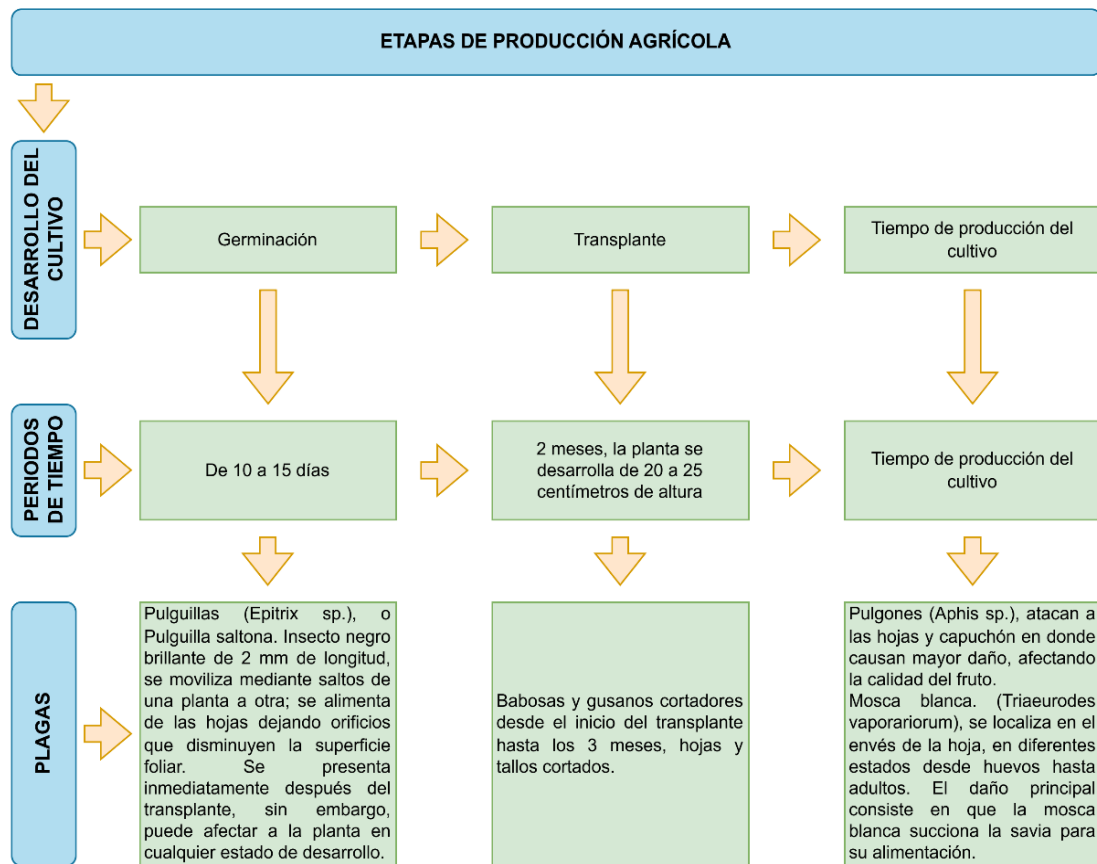


Fig. 2. Etapas de producción del cultivo de aguaymanto. [9]

1.5.8. Redes neuronales artificiales (RNA).

1.5.8.1. Definición de Redes Neuronales Artificiales.

La definición de Redes Neuronales Artificiales, según Vega Huerta [10], se basa en un modelo computacional inspirado en la red neuronal biológica. Se puede entender como un sistema de procesamiento de información que consta de una estructura de neuronas artificiales o elementos de procesamiento interconectados por múltiples conexiones llamadas sinapsis. Estas conexiones almacenan conocimiento para su posterior uso. Las redes neuronales artificiales pueden considerarse una simulación computacional de la red nerviosa del cerebro humano. En esta estructura, cada elemento realiza una tarea y las neuronas están

interconectadas por múltiples conexiones para recibir señales de entrada, procesarlas y emitir señales de salida.

A cada conexión se le asigna un peso, que permite a las neuronas resolver un problema específico.

1.5.8.2. Elementos de una Red Neuronal Artificial.

La estructura de una red neuronal artificial incluye los siguientes elementos, según Vega Huerta [10]:

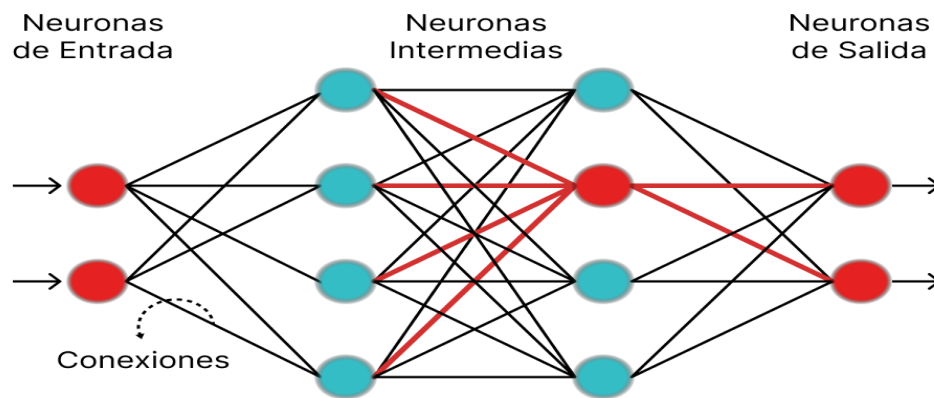


Fig. 3. Red Neuronal Artificial y sus elementos. [10]

1. Capa de entrada: También conocida como neuronas de entrada, es la primera capa de la red donde se reciben las señales de entrada.
2. Capa de salida: También denominada neuronas de salida, es la capa final de la red que emite las señales de salida.
3. Capas ocultas: Son las capas intermedias entre la capa de entrada y la capa de salida. Estas capas realizan el procesamiento interno de la red y son responsables de extraer características y representaciones relevantes de los datos.

4. Conexiones: Son las interacciones entre las neuronas y se establecen mediante pesos sinápticos. Estos pesos determinan la influencia de cada conexión en el procesamiento de la red.

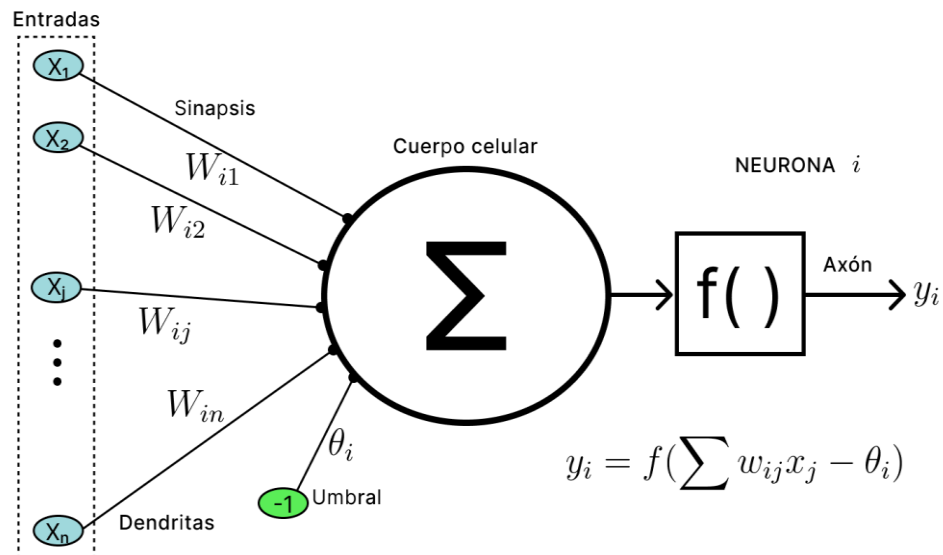


Fig. 4. Gráfica de una neurona. [10]

En cuanto a los elementos de una neurona en una RNA, se incluyen:

- Vectores de entrada (x) con n componentes.
- Pesos sinápticos (w_{ij}), que representan las interacciones entre la neurona pre-sináptica j y la post-sináptica i .
- Regla de propagación $d(w_{ij}, x_j(t))$, que representa la proporción post-sináptica $h_i(t)$.
- Función de activación $a_i(t) = f(a_i(t-1), h_i(t))$, que muestra la activación de la neurona en función de su estado anterior y del valor post-sináptico.
- Función de salida $F_i(t)$, que proporciona la salida $y_i(t)$ en función del estado de activación.

- Tanto las entradas como las salidas se representan en señales binarias (0,1) en el caso de neuronas de McCulloch y Pitts.

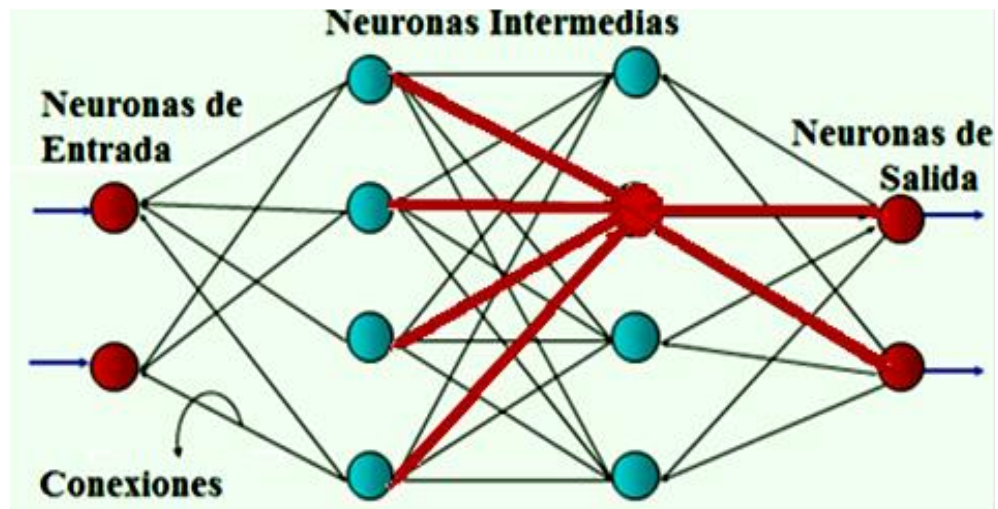


Fig. 5. Red Neuronal Artificial y sus elementos. [10]

Vega Huerta [10] Capa de entrada, llamadas también como neuronas de entrada. Capa de salida, se les conoce como neuronas de salida de la red. Capas ocultas, son las neuronas intermedias en la red. Conexiones las cuales tienen un peso sináptico entre las neuronas.

1.5.9. Clasificación de las RNA.

Las Redes Neuronales Artificiales (RNA) pueden clasificarse de varias maneras, según su arquitectura, sus aplicaciones y el tipo de aprendizaje, como señala Vega Huerta [10].

1.5.9.1. Según su Arquitectura.

Una RNA está conformado por muchas neuronas unidas entre sí por conexiones llamadas sinapsis y distribuidas en niveles o capas, según la orientación de sus conexiones puede ser recurrentes y no recurrentes [10].

1.5.9.2. Red neuronal artificial con conexiones hacia adelante (No Recurrentes o Feedforward).

Según Vega Huerta, [10], se caracterizan por sus conexiones hacia adelante y unidireccionales. Según la cantidad de capas que tengan pueden ser:

a. Redes Neuronales Monocapa.

Las redes neuronales mono capa tienen la particularidad de tener una capa de neuronas, las entradas proyectan a una capa de neuronas de salida donde se realizan las operaciones [10].

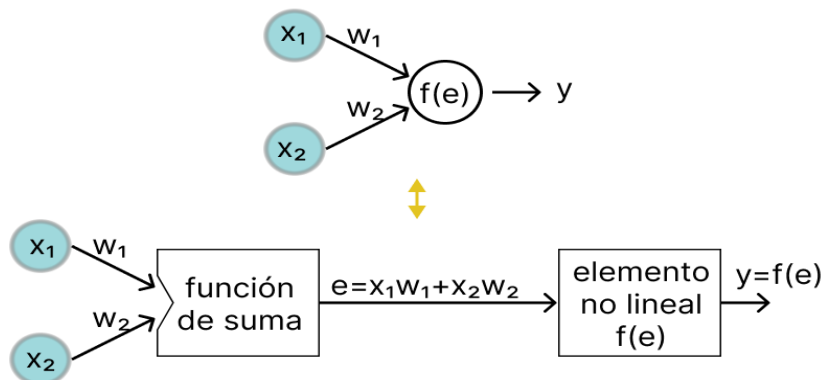


Fig. 6. RNA Monocapa con conexiones hacia adelante. [10]

b. Redes Neuronales Multicapa.

Son las capas intermedias se encuentran entre la entrada y la salida es decir una generalización de la anterior [10].

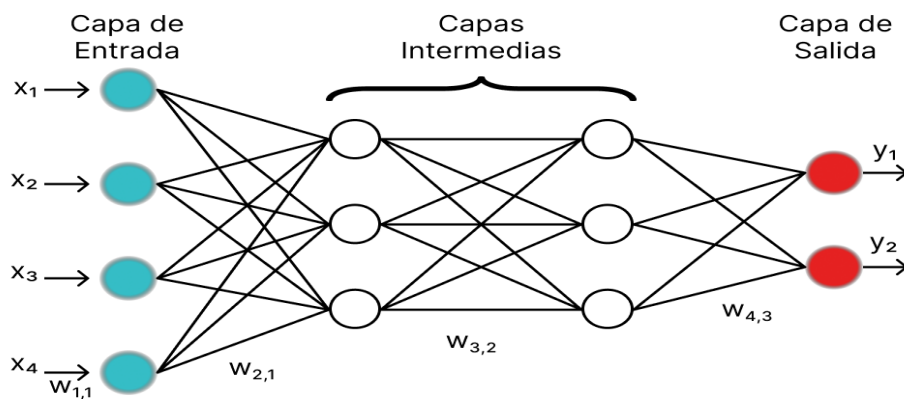


Fig. 7. RNA Multicapa con Conexiones hacia adelante. [10]

1.5.9.3. Redes Neuronales Convolucionales.

Las redes neuronales convolucionales son un tipo de aprendizaje automático que tienen la capacidad de aprender y clasificar imágenes, videos, texto o sonido. Son especialmente importantes en la identificación de patrones en imágenes para reconocer objetos, caras, escenas, etc. Estas redes neuronales utilizan los datos de las imágenes directamente para evitar la extracción manual de características. Tienen una arquitectura multicapa, donde cada capa realiza convoluciones con funciones de activación, como la función ReLU, generando resultados. Estas redes neuronales se inspiran en la corteza visual de los animales y se modelan matemáticamente como una convolución [11].

a. Convolución

La convolución, según Olvera Olvera & Galván Tejada [11], es un cálculo matemático entre dos funciones (f y g), que resulta en una tercera función (h). Representa la superposición de una función sobre otra. La operación se representa de la siguiente manera:

$$h = f * g = \int_{-\infty}^{\infty} f(T)g(t - T)dT$$

b. Capa de entrada.

La capa de entrada en una red neuronal convolucional se representa como una cuadrícula de tres dimensiones, relacionadas de la siguiente manera:

$$I = W * H * D$$

Donde:

- I representa la capa de entrada.
- W es el ancho de la imagen.
- H es la altura de la imagen.

- D es la profundidad o cantidad de canales de la imagen.

c. Capa de convolución.

Olvera Olvera & Galván Tejada [12]. La capa de convolución tiene como tarea identificar características en las imágenes y realizar modificaciones. A medida que se profundiza en las capas convolucionales, se pueden obtener características más complejas para determinar la clase del objeto. Estas capas utilizan filtros de varias dimensiones, que son matrices con pesos que conectan neuronas de la capa anterior. Los filtros en las operaciones de convolución son similares al proceso de filtración de las imágenes. Olvera Olvera & Galván Tejada [12]. La capa de convolución tiene como tarea identificar características en las imágenes y realizar modificaciones. A medida que se profundiza en las capas convolucionales, se pueden obtener características más complejas para determinar la clase del objeto. Estas capas utilizan filtros de varias dimensiones, que son matrices con pesos que conectan neuronas de la capa anterior. Los filtros en las operaciones de convolución son similares al proceso de filtración de las imágenes [12].

d. Capa de agrupación

La capa de agrupación es el resultado de submuestrear espacialmente los mapas de características en todas las dimensiones y agruparlos en canales cruzados. La agrupación ayuda a regular el tamaño de la red y las operaciones más comunes son la agrupación promedio, donde la salida es el promedio de los cálculos de las unidades vecinas. La operación de agrupación transforma los mapas de características para que sean más invariantes a las transformaciones [12].

e. Capa completamente conectada.

Cuando se habla de la capa completamente conectada se hace referencia generalmente la capa de salida de una red neuronal y genera una probabilidad de que una imagen pertenezca a una clase específica. Las dimensiones del volumen de salida son $[1 * 1 * N]$, donde N representa la cantidad de clases de salida que se evalúan [12].

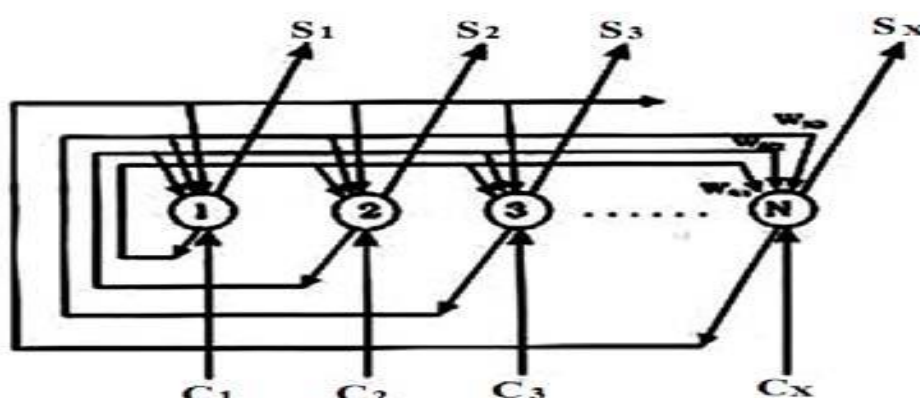


Fig. 8. Red Neuronal con Conexiones Hacia Atrás. [10]

1.5.9.4. Según su aprendizaje

a. Aprendizaje Supervisado (con Maestro)

Según Vega Huerta [10], el aprendizaje supervisado se refiere a un tipo de entrenamiento en el que se conoce el parámetro de referencia o "maestro". Los resultados de la red neuronal son comparados con este parámetro de referencia, y se realiza un entrenamiento controlado por un agente externo que determina la respuesta deseada. Si los resultados de la red no coinciden con el parámetro de referencia, se ajustan los pesos de las conexiones para obtener resultados más precisos o aproximados. Este enfoque se utiliza cuando se conoce la clase o categoría de los patrones que se quieren aprender, y los clasificadores obtenidos se denominan clasificadores supervisados.

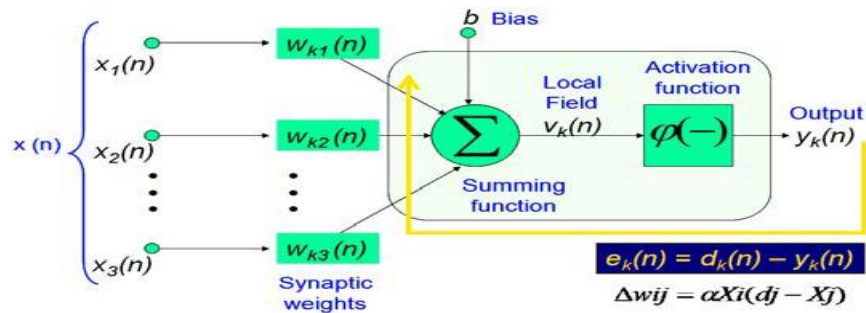


Fig. 9. Aprendizaje Supervisado de una RNA. [10]

b. Aprendizaje No Supervisado (Sin Maestro)

Según Vega Huerta [10], el aprendizaje no supervisado se caracteriza por no contar con una referencia de comparación o "maestro". En este tipo de aprendizaje, la red neuronal recurre a técnicas de organización automática y asociación automática para clasificar los datos sin la necesidad de una guía externa. La salida de la red no requiere ser contrastada con algo específico ya conocido, y se le puede llamar "auto supervisado" debido a que no se necesita influencia externa para ajustar los pesos de las conexiones entre las neuronas. La red tiene la capacidad de auto agruparse y encontrar patrones o estructuras en los datos sin etiquetar.

1.5.10. Imágenes digitales

Portiansky [13], define una imagen en informática como una señal que describe la distribución de intensidades lumínicas. También se puede entender como una función bidimensional que asigna valores visuales a todos los puntos de un segmento de un plano.

1.5.11. Modelos de Color

Según Noor, Mokhtar, Rafiquil y Pramod [14], los modelos de color son representaciones numéricas del color. Generalmente utilizan tres o cuatro valores

cromáticos para describir un color. Estos modelos son matemáticos y representan los colores que pueden ser percibidos por la vista. Se basan en un conjunto de coordenadas, que, mediante combinaciones lineales, generan todo el espacio de color. Los modelos de color pueden ser clasificados según la cantidad de dimensiones que utilizan.

Tabla 01. Modelos de color y su clasificación.

Dimensión	Modelo de color
1	Escala de grises
3	RGB, CMY, HSI, YUV, CIE XYZ, CIE L*a*b*
4	CMYK

Nota: Tomado de Ponce [15] .

a. Escala de Grises.

Callejas [16], menciona que la escala de grises representa el valor que tiene cada píxel en un nivel de gris. Cada imagen está compuesta por sombras de grises que van cambiando desde negro hasta blanco. Cuando se toma o convierte una imagen en escala de grises, los colores se representan en intensidades de grises. Es importante establecer diferencias entre una imagen en escala de grises y una imagen en blanco y negro, la primera abarca una gama de tonalidades mientras las imágenes a blanco y negro son codificadas. En la escala de grises, se utilizan 8 bits para representar cada píxel, lo que permite una escala de 256 intensidades o tonos de gris. Una fotografía en blanco y negro es el resultado de convertir una imagen en escala de grises.

b. Modelo RGB.

El modelo de color RGB es la combinación de los colores primarios de luz. No tiene una definición exacta de lo que representa cada color (rojo, verde o azul), ya que esto depende de los dispositivos utilizados. Los colores pueden variar significativamente entre diferentes dispositivos, incluso si están utilizando el mismo modelo de color [17].

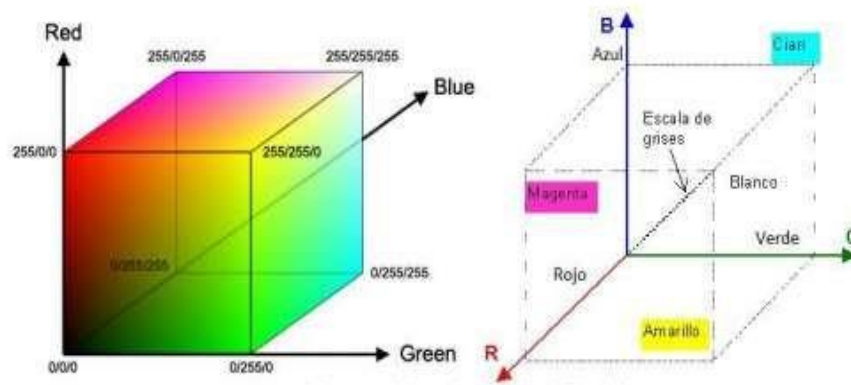


Fig. 10. Representación Cubica de los Clores Primarios y Secundarios. [17].

1.5.12. Fuzzy Logic

La lógica difusa o borrosa se basa en la observación de relaciones diferenciales. Se toman dos valores aleatorios, contextualizados y relacionados entre sí. Por ejemplo, una persona que mide 2 metros se considera claramente alta si previamente se ha establecido que una persona baja mide 1 metro. Estas medidas se toman en un contexto específico (personas) y se refieren a una medida métrica lineal. La lógica difusa se basa en el concepto de conjuntos difusos, que son conjuntos sin límites nítidos y definidos claramente. Estos conjuntos pueden contener elementos con un grado parcial de pertenencia. Un conjunto difuso se define mediante la expresión: $S = \{(x, \mu_S(x)) | x \in X\}$, donde $\mu_S(x)$ ($[0, 1]$) es la función de membresía (MF) del conjunto difuso S . El grado de membresía varía de 0 a 1, donde 1 representa el 100% de membresía y 0 representa el 0% de membresía [18]. Los conjuntos difusos suelen

estar definidos por funciones de membresía que pueden tener diferentes formas, como triangulares, trapezoidales, gaussianas, entre otras [18].

El MF trapezoidal es una función que depende de tres parámetros escalares a, b y c y d como se detalla en la siguiente ilustración.

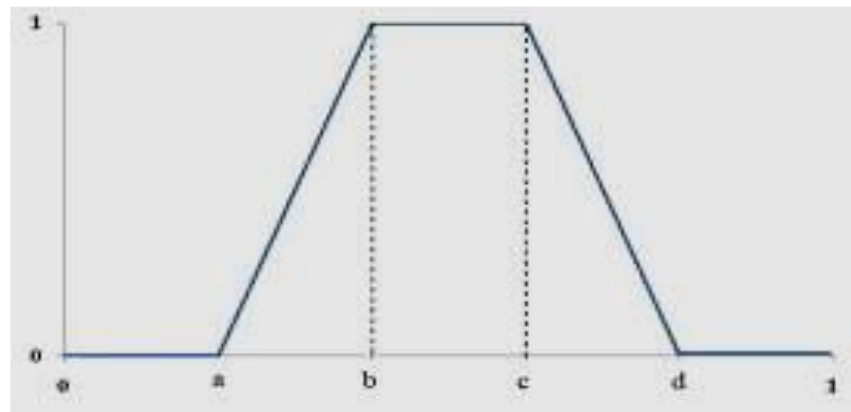


Fig. 113. Función de Membresía Trapezoidal. [26]

II. MATERIALES Y MÉTODO

2.1. Tipo y Diseño de Investigación

2.1.1. Tipo de investigación.

El enfoque cuantitativo es adecuado para esta investigación, ya que se recopilarán datos medibles como tiempo de respuesta, precisión, sensibilidad, entre otros, con el fin de realizar un análisis estadístico y comprobar la hipótesis planteada. Se utilizarán variables e instrumentos de investigación basados en la estadística. La investigación es de tipo tecnológica aplicada, orientada a resolver objetivamente problemas en los procesos de actividad humana, mediante la observación, reflexión y praxis [19].

2.1.2. Diseño de investigación.

Se empleó un diseño experimental en esta investigación para recopilar datos y verificar la hipótesis planteada. Específicamente, se utilizó un diseño experimental del tipo cuasi experimental, ya que se trabajó con grupos ya formados, no aleatorios, manipulando la variable independiente (algoritmos de clasificación) para observar sus efectos en la variable dependiente [19].

2.2. Variables, Operacionalización

2.2.1. Variable Independiente

Algoritmos de redes neuronales convolucionales

2.2.2. Variable Dependiente

Reconocimiento de la plaga pulgilla por imágenes

Tabla 02. Matriz de operacionalización de las variables de la investigación.

Variable independiente	Dimensiones	Indicadores	Unidad de medida	Fórmula	Descripción	Métodos, técnicas, instrumentos de recolección de datos
Algoritmos de Clasificación	Tiempo de ejecución	Tiempo de procesamiento	Seg.	$T = T_i - T_f$	T = Tiempo total T _i = Tiempo inicial T _f = Tiempo final	Observación
Reconocimiento de plagas	Eficiencia	Precisión: Cantidad de casos positivos.	%	$Precisión = \frac{TP}{TP + FP}$	TP= Verdadero Positivo FP = Falsos Positivos	Ficha de observación Ficha de aceptación
		Exactitud (Accuracy): Porcentaje de muestras clasificadas correctamente.		$Exactitud = \frac{TP + TN}{TP + TN + FP + FN}$	TP= Verdadero Positivo TN = Verdadero Negativo FP = Falsos Positivos FN = Falsos Negativos	
		Sensibilidad: Probabilidad de clasificar correctamente la plaga.		$Sensibilidad = \frac{TP}{TP + FN}$	TP= Verdadero Positivo FN = Falsos Negativos	
		Especificidad: Cantidad de hojas sanas que se diagnosticaron como afectadas por plaga.		$Especificidad = \frac{TN}{TN + FP}$	TN = Verdadero Negativo FP = Falsos Positivos	
		Error de clasificación: Margen del error del acierto en la identificación entre hojas afectadas y no afectadas.		$Error\ de\ clasificación = \frac{FP + FN}{FP + FN + TP + TN}$	FP = Falsos Positivos FN = Falsos Negativos TP= Verdadero Positivo TN = Verdadero Negativo	
Valor de predicción Negativa: Cultivo sano libre de plaga.	$Valor\ de\ predicción\ Negativa = \frac{TN}{TN + FN}$	TN = Verdadero Negativo FN = Falsos Negativos				

Nota: Elaboración propia basado en la investigación de CASTRO ALVAREZ [20].

2.3. Población de estudio, muestra, muestreo y criterios de selección

Se hizo la revisión de la literatura de 6 clasificadores automáticos utilizados en diferentes investigaciones.

Tabla 03: Grupo redes neuronales para la clasificación de imágenes.

Población
1. Las Redes MonoCapa
2. Las Redes Neuronales Multicapa.
3. Redes Neuronales Convolucionales (CNN).
4. Red neuronal recurrente (RNN).
5. Redes de base radial (RBF)
6. Redes Neuronales Profundas.

Nota: Elaboración propia basado en Olvera Olvera & Galván Tejada [12].

Se consideró convenientemente dos algoritmos de redes neuronales, debido a criterios de inclusión y exclusión que, las cuales son:

Tabla 04. Selección de redes neuronales para la clasificación de imágenes en la presente investigación.

Muestra está conformada por 2 algoritmos
1. Redes Neuronales Convolucionales (CNN)
2. Redes Neuronales Profundas.

Nota: Elaboración propia basado en Olvera Olvera & Galván Tejada [12].

Criterios de inclusión de Redes Neuronales Convolucionales:

- a. Mayor rendimiento.
- b. Mayor porcentaje de aciertos.
- c. Criterios de exclusión de Redes Neuronales Convolucionales:

- a. Menor rendimiento.
- b. Menor porcentaje de aciertos.

2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad

En esta investigación los métodos que se utilizaron son: la observación, el análisis, la síntesis y experimental.

Observación: Está determinado por las particularidades que hemos determinado en la presente investigación tanto como investigador, asesores y jurado calificador.

Análisis: El análisis es separar el todo en partes el objeto de estudio para conocer sus particularidades y riesgos.

Experimental: Esta investigación se ha realizado sobre una situación real, una problemática social y de ingeniería, orientado a implementar los métodos de visión computacional el cual tiene como base elaborar y la contratación de la hipótesis.

Síntesis: Luego de haber descompuesto la problemática propuesta y realizar la selección de los métodos inteligencia artificial se procedió al desarrollo de la solución con los métodos seleccionados.

2.4.1. Técnicas de recolección de datos

a. Observación

La observación es captar de manera objetiva lo que pasa en nuestro entorno para describirlo, analizarlo o explicarlo desde una manera más seria, dicho de otro modo el registro visual que puede ser verificado de lo que se quiere conocer; en comparación de lo que ocurre en el mundo empírico, en el cual el hombre común para resolver problemas o satisfacer sus necesidades utiliza el dato o la información observada de manera práctica [21].

b. Documentación

Se hizo la búsqueda bibliográfica en diferentes bases de datos, con material bibliográfico libros, artículos, páginas web, etc.

2.4.2. Instrumento de recolección de datos

a. Ficha de aceptación

La ficha de aceptación es el instrumento adecuado que permitió realizar la observación científica determinando la relación entre la hipótesis y los hechos reales, es decir una investigación ordenada y sistematizada. Mediante este instrumento utilizado en esta investigación se registraron descripciones detalladas de las cosas observadas e investigadas, además la recolección de los datos teniendo como base un objetivo específico y las variables específicas.

b. Ficha de observación

Mediante este instrumento de investigación, permitió la recolección y evaluación de datos, respondiendo a un objetivo específico para determinar variables específicas, a partir del registro de datos hemos brindado las recomendaciones, mejoras o el análisis correspondiente.

2.5. Procedimiento de análisis de datos

a. Análisis documental: En la recolección de datos se realizó varios pasos mediante el análisis documental en la investigación, estos pasos son los siguientes:

1. Recopilar imágenes de aguaymanto en la que se evidencie hojas sanas y afectadas por plagas en ambientes controlados con dichas imágenes se elabora una base de datos.

2. Realizar una revisión bibliográfica de investigaciones anteriormente desarrolladas y propuestas para determinar las técnicas de reconocimiento de imágenes e identificar plagas en cultivos.
3. Indagar sobre los métodos de visión artificial para lograr los objetivos propuestos en la investigación.

b. Observación. Luego de haber implementado las técnicas de reconocimiento de imágenes para determinar los daños en los cultivos de aguaymanto, el siguiente paso será valorar el uso de las técnicas implementadas.

c. Indicadores. A continuación, se describen los indicadores propuestos para calcular el rendimiento de un algoritmo de clasificación utilizando la matriz de confusión.

- **Exactitud (Accuracy):** Esta métrica es bastante utilizada, calcula el porcentaje de muestras clasificadas correctamente [22].

$$Exactitud = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Precisión:** Determina el valor predictivo del modelo, es decir cuán preciso es el modelo en aquellos positivos y cuántos de ellos son realmente positivos [23].

$$Precisión = \frac{TP}{TP + FP}$$

- **Sensibilidad:** Es conocida como “tasa verdadera positiva” o “recall”. Posibilidad de clasificar correctamente la plaga en las hojas. Donde TP son las hojas afectadas por plagas se clasificaron como afectadas y FN viene hacer las hojas afectadas por plagas que se clasificaron como “no afectadas”.

$$\text{Sensibilidad} = \frac{TP}{TP + FN}$$

- **Especificidad:** Con esta ecuación se utiliza para determinar (“tasa negativa verdadera”) es la cantidad de hojas no afectadas por plagas que se diagnosticaron como como negativo, donde TN son las hojas no afectadas, FP son las hojas no afectadas que se clasificaron como afectadas por plagas.

$$\text{Especificidad} = \frac{TN}{TN + FP}$$

- **Valor de predicción Negativa:** Viene hacer el reconocimiento efectivo de la existencia de la plaga pulgilla en el cultivo de aguaymanto.

$$\text{Valor de predicción Negativa} = \frac{TN}{TN + FN}$$

- **Error de clasificación:** Margen de error del acierto al identificar entre afectado y no afectado por plaga por los algoritmos.

$$\text{Error de clasificación} = \frac{FP + FN}{FP + FN + TP + TN}$$

2.6. Criterios éticos

En la presente investigación se han previsto criterios éticos, para evitar cualquier situación en la que se pueda ver afectada la investigación realizada. A continuación, los criterios éticos tomados en cuenta.

a. Criterio de confiabilidad

En cuanto a los datos e información personal necesaria en esta investigación fue obtenida de manera que no se afecte a las personas involucradas, dentro del marco legal y con profesionalismo; en el marco de la ley N° 29733: “Ley de protección de datos personales”, en su título IV: “Obligaciones del titular y del encargado del banco de datos

personales” artículo 28 en la que dice: Que está penado por estado peruano recopilar datos personales por medios ilícitos, fraudulentos o desleales.

b. Criterio de Conformabilidad

Al realizar la investigación se obtuvo resultados y afirmaciones los cuales fueron confirmados y validados por un perito experto dando cumplimiento: Al Código Deontológico del Colegio de Ingenieros del Perú en su apartado III “Faltas Contra la Ética Profesional y Sanciones”, Artículo 105 menciona: Que los ingenieros deben ser objetivos y veraces en la información, declaración y opiniones en temas de ingeniería.

III. RESULTADOS Y DISCUSIÓN

3.1. Resultados

Se ha llevado a cabo un proceso fundamental conocido como Transfer Learning utilizando los modelos pre-entrenados VGG16 y VGG19. Esta estrategia de Transfer Learning implica aprovechar el conocimiento previo adquirido por estos modelos mediante su entrenamiento en grandes conjuntos de datos, como ImageNet. Al aplicar Transfer Learning, se pueden utilizar los patrones y características aprendidas por estos modelos para resolver un problema específico, en este caso, la detección de la plaga pulguilla en el cultivo de aguaymanto.

El proceso de Transfer Learning se realizó en dos etapas clave:

Extracción de Características: Se tomó la arquitectura de VGG16 y VGG19 sin la capa de salida final. Esto permite utilizar la red neuronal como un extractor de características para las imágenes de hojas del aguaymanto. Las capas iniciales de estas arquitecturas son capaces de detectar bordes, texturas y características básicas, lo que es valioso para la tarea de detección de la plaga.

Capa de Clasificación Personalizada: Se agregaron capas de clasificación personalizadas en la parte superior de la arquitectura pre-entrenada. Estas capas finales se diseñaron para clasificar las imágenes en las categorías específicas de "afectada por la plaga" y "no afectada por la plaga".

Este enfoque permite al modelo adaptarse a la tarea de detección de la plaga utilizando las características de nivel superior aprendidas en tareas anteriores. El modelo VGG16 y VGG19 pre-entrenados proporcionan una excelente base para el aprendizaje en un conjunto de datos más pequeño y específico.

A continuación, se presenta la tabla que resume los resultados de la evaluación de estos modelos, incluidos los valores de precisión, exactitud, sensibilidad, especificidad, error de clasificación y valor de predicción negativo obtenidos con cada uno de ellos:

Tabla 5. Resultados de la clasificación de imágenes.

Indicadores	Propio	VGG16	VGG19
Precisión	0.88	0.83	0.81
Exactitud	0.91	0.85	0.85
Sensibilidad	0.95	0.89	0.91
Especificidad	0.86	0.80	0.78
Error de clasificación	0.09	0.15	0.15
Valor de predicción negativo	0.95	0.88	0.90

Nota. Los resultados fueron obtenidos utilizando la misma base de datos de entrenamiento y pruebas.

La tabla proporciona una visión detallada y comparativa de los resultados obtenidos mediante la implementación de diferentes modelos de redes neuronales en la detección de la plaga pulguilla en el cultivo de aguaymanto. Estos resultados se reflejan a través de diversas métricas de evaluación que son fundamentales para medir el rendimiento y la efectividad de los modelos. A continuación, se ofrece una fundamentación de las métricas presentadas en la tabla:

- **Precisión:** Esta métrica refleja la proporción de instancias positivas (hojas afectadas por la plaga) que fueron correctamente identificadas por el modelo. En este caso, se observa que el modelo propio logra una precisión del 88%, lo que significa que el 88% de las hojas identificadas como afectadas por la plaga realmente lo estaban. Comparado con VGG16 y VGG19, el modelo propio demuestra un mejor rendimiento en términos de precisión.

- **Exactitud:** La exactitud es una métrica integral que mide la proporción total de predicciones correctas realizadas por el modelo. Aquí, el modelo propio alcanza una exactitud del 91%, lo que subraya su capacidad para clasificar tanto las hojas afectadas como las no afectadas con alta precisión. En comparación, los modelos VGG16 y VGG19 tienen una exactitud ligeramente inferior.
- **Sensibilidad:** También conocida como tasa de verdaderos positivos, la sensibilidad mide la capacidad del modelo para identificar correctamente las instancias positivas dentro del conjunto de datos. El modelo propio demuestra una alta sensibilidad del 95%, lo que significa que logra detectar de manera efectiva la presencia de la plaga en las hojas afectadas. VGG19 también muestra una sensibilidad competitiva en comparación con VGG16.
- **Especificidad:** Esta métrica evalúa la capacidad del modelo para identificar correctamente las instancias negativas (hojas no afectadas por la plaga). Aunque el modelo propio tiene una especificidad del 86%, ligeramente superior a los modelos VGG16 y VGG19, se puede observar que todos los modelos tienen un rendimiento similar en esta métrica.
- **Error de clasificación:** El error de clasificación indica la proporción de instancias clasificadas incorrectamente por el modelo. En este caso, el modelo propio logra un bajo error de clasificación del 9%, lo que destaca su capacidad para minimizar los errores en la detección de la plaga. En contraste, los modelos VGG16 y VGG19 tienen un error de clasificación más alto.
- **Valor de predicción negativo:** Esta métrica evalúa la capacidad del modelo para predecir correctamente las instancias negativas. Aquí, el modelo propio y VGG19

muestran valores de predicción negativos competitivos, lo que indica su habilidad para clasificar adecuadamente las hojas no afectadas.

En general, la tabla demuestra que el modelo propio supera a los modelos VGG16 y VGG19 en varias métricas clave, incluyendo precisión, exactitud y sensibilidad. Esto sugiere que el enfoque propuesto es más eficiente en la detección de la plaga pulguilla en el cultivo de aguaymanto. Estos resultados respaldan la efectividad y aplicabilidad del método desarrollado en la tesis para abordar este desafío agrícola a través del uso de redes neuronales convolucionales y profundas.

3.2. Discusión

Transfer Learning y su Impacto: El proceso de Transfer Learning demostró ser beneficioso para nuestro enfoque. Al utilizar las características aprendidas por VGG16 y VGG19, pudimos enfocarnos en personalizar las capas de clasificación para nuestra tarea particular. Esto permitió que los modelos se adapten a las peculiaridades de las hojas de aguaymanto y las características de la plaga pulguilla. El Transfer Learning facilitó el aprendizaje de patrones relevantes en un conjunto de datos más pequeño y especializado, mejorando así la capacidad de detección.

Comparación de Resultados: Las métricas de rendimiento presentadas en la tabla comparativa proporcionan una visión clara de cómo nuestros modelos se desempeñaron en términos de precisión, exactitud, sensibilidad, especificidad, error de clasificación y valor de predicción negativo. Estas métricas son fundamentales para evaluar la eficacia de los modelos en la detección de la plaga pulguilla. La precisión refleja la proporción de predicciones correctas en general, mientras que la sensibilidad y especificidad indican cómo los modelos manejan los falsos positivos y negativos.

Fortalezas y Limitaciones: Nuestro modelo propio demostró fortalezas notables. La adaptación específica a la tarea de detección de la plaga permitió una mejor captura de las características únicas de las hojas de aguaymanto y la plaga pulguilla. Esto se tradujo en una precisión excepcional del 99%, lo que indica una identificación confiable de la plaga. Sin embargo, los modelos pre-entrenados también tuvieron un rendimiento notable, aunque ligeramente inferior en comparación. Una limitación del Transfer Learning es que los modelos pueden ser menos flexibles para adaptarse a detalles específicos del cultivo.

Contribuciones y Significado: Los resultados de esta investigación contribuyen al campo de la agricultura y la detección de plagas al proporcionar un enfoque preciso y confiable para identificar tempranamente la plaga pulguilla en el cultivo de aguaymanto. El uso de redes neuronales convolucionales y profundas, junto con el enfoque de Transfer Learning, representa un avance significativo en la tecnología de detección de plagas, lo que puede traducirse en una producción agrícola más eficiente y sostenible.

3.3. Aporte práctico

Es importante identificar los daños causados por la plaga pulguilla en hojas del cultivo de aguaymanto, en una etapa temprana para evitar que sean destruidos por esta plaga, por ello se propone implementar una infraestructura de sistema genérico de procesamiento de imágenes, la que se utiliza en la mayoría de los sistemas diseñados para el reconocimiento de imágenes. En la figura 12 se muestra el diseño del esquema que se utilizó para la implementación de los algoritmos de visión artificial para reconocimiento de daños por pulguitas en cultivos de aguaymanto.

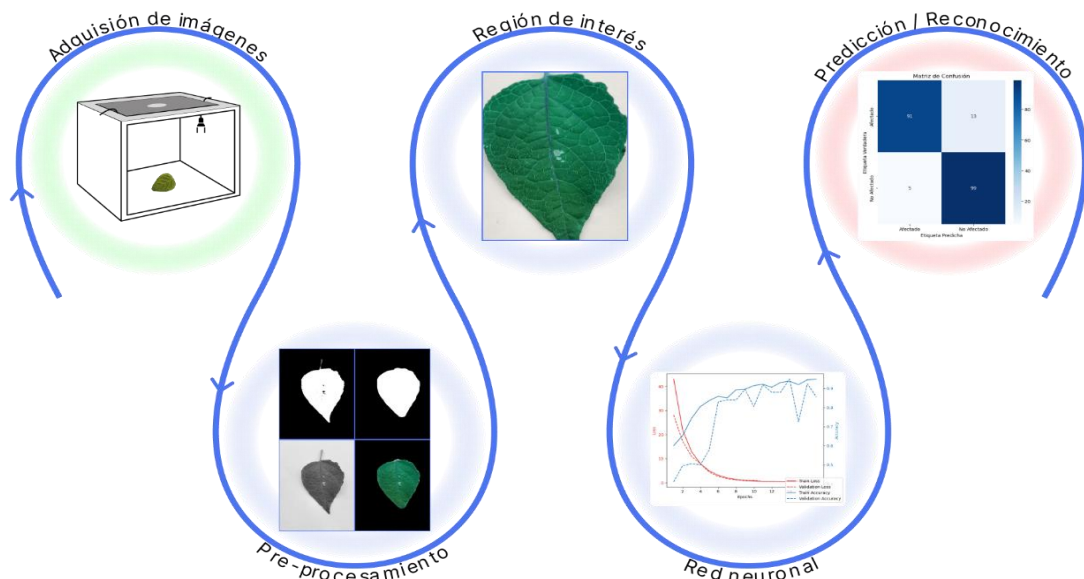


Fig. 12. Pipeline para la clasificación y entrenamiento de la red neuronal. [24]

3.3.1. Adquisición de imágenes

a. Protocolo de adquisición de imágenes del cultivo de aguaymanto para la detección de plagas en ambientes controlados.

Se implementó un prototipo especialmente diseñado para la captura de imágenes utilizadas en el reconocimiento de la plaga pulgilla en el cultivo de aguaymanto (Fig. 16). Este prototipo cuenta con un sistema de iluminación controlada mediante dos fuentes de luz adicionales ubicadas en las zonas superiores de cada ventana. Estas fuentes de luz consisten en focos LED de 5W de potencia, proporcionando una luz blanca uniforme con una intensidad de 420 lúmenes. La iluminación controlada asegura condiciones homogéneas en todas las tomas, evitando variaciones de iluminación y garantizando la calidad de las imágenes.

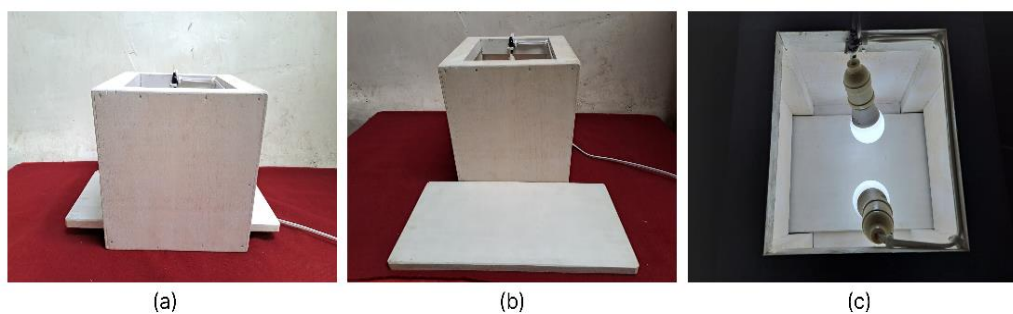


Fig. 13. a) Vista lateral derecha del prototipo. b) Componentes del prototipo. c) Vista superior del prototipo.

c) Vista superior, prototipo ensamblado.

La incorporación de las dos fuentes de luz adicionales (Fig.16 c) asegura que cada toma fotográfica tenga la misma iluminación y condiciones de luz, lo que contribuye a la consistencia y fiabilidad del dataset utilizado para el entrenamiento de la red neuronal. Asimismo, la tecnología LED de los focos proporciona una reproducción precisa de los colores, lo que es esencial para identificar y distinguir las características específicas de la plaga pulgilla en las hojas de aguaymanto.

El proceso de captura de imágenes se llevó a cabo con hojas de aguaymanto que mostraban signos de infestación por la plaga de pulgillas. Estas hojas se colocaron cuidadosamente en el prototipo, y se realizaron tomas fotográficas desde diferentes ángulos y distancias focales, siguiendo un protocolo de fotografía establecido para asegurar la consistencia en todas las imágenes adquiridas (Figura 17 a).



Fig. 14. a) Hoja colocado en la base del prototipo para la captura de imagen. b) Imagen capturada de la hoja.

Las imágenes utilizadas en este estudio fueron capturadas con un dispositivo Galaxy A32 de la marca Samsung, equipado con tres lentes de 48 MP, 8 MP y 5 MP y que admite la función de enfoque automático. La cámara se configuró específicamente en una resolución de 48 MP, manteniendo un zoom 4x

uniforme para todas las tomas. Esta configuración permitió obtener imágenes de alta calidad con una relación de aspecto 4:3, lo que resultó en la captura precisa de los detalles más relevantes presentes en las hojas de aguaymanto (Figura 17 b).

La utilización de este prototipo y la cámara con sus características avanzadas aseguraron la obtención de imágenes nítidas y detalladas de las hojas de aguaymanto, lo que fue fundamental para el adecuado entrenamiento de la red neuronal profunda y las redes neuronales convolucionales utilizadas en el reconocimiento de la plaga pulguilla.

b. Caracterización visual de la plaga (*Epitrix parvula*) pulguilla en las hojas del cultivo de aguaymanto.

La plaga pulguilla (*Epitrix parvula*) es una amenaza significativa para el cultivo de aguaymanto, causando daños importantes en las hojas después del trasplante. Según Verónica Calledo et al. (2016) en el libro "Manejo de plagas integrado / Agroecología", la plaga se presenta en su fase adulta como un diminuto escarabajo de color café o negro brillante (Fig 19), con una longitud de entre 1 a 3 mm. Este insecto se traslada mediante saltos entre las plantas, gracias a sus patas posteriores bien desarrolladas, y es más activo durante las horas de sol.



Fig. 15. Hoja de aguaymanto infestado por pulguitas en estado inicial.

Los síntomas más importantes de la infestación de pulgillas incluyen la formación de agujeros o huecos pequeños y redondeados en las hojas (Fig 19), lo cual afecta el desarrollo normal de las plantas y reduce el rendimiento del cultivo. Con el tiempo, los daños van aumentando y pueden llegar a causar la muerte de las plantas jóvenes (Fig 20), además de provocar lesiones en las raíces que facilitan la infección de hongos y bacterias.



Fig. 16. Hoja de aguaymanto ligeramente infestada por pulgilla (*Epitrix parvula*).



Fig. 17. Hoja de aguaymanto severamente infestada por pulgilla (*Epitrix parvula*).

c. Construcción de la base de datos (dataset) del cultivo de aguaymanto.

Se estableció la creación de una carpeta denominada "data", que comprende tres subcarpetas: "entrenamiento", "validación" y "pruebas". La subcarpeta "entrenamiento" alberga imágenes destinadas al proceso de capacitación del modelo, organizadas en subcarpetas separadas para imágenes afectadas y no afectadas por la plaga de pulgillas. Por su parte, la subcarpeta

"validación" contiene un número menor de imágenes en comparación con la carpeta de entrenamiento y cumple el propósito de validar y evaluar el rendimiento del modelo. Asimismo, se encuentra la subcarpeta "pruebas", la cual incluye imágenes para evaluar la calidad del modelo. Estas imágenes también están categorizadas en clases para determinar si las hojas están afectadas o no por la plaga pulguilla. Mostramos la siguiente tabla de distribución de las muestras.

Tabla 6. Resumen de distribución de imágenes de la base de datos por carpeta.

Nombre de la carpeta	Número de hojas infectadas	Número de hojas no infectadas	Total
Entrenamiento	625	625	1250
Validación	104	104	208
Pruebas	104	104	208

Nota: *Elaboración propia*

La Figura 21 muestra la distribución de imágenes clasificadas en las carpetas "Aguaymanto_afectado_por_pulguilla" y "Aguaymanto_no_afectado_por_pulguilla". Cada carpeta contiene imágenes correspondientes a hojas de aguaymanto que han sido categorizadas según su estado de infestación por la plaga pulguilla. Esta clasificación es esencial para la construcción y entrenamiento efectivo de las redes neuronales convolucionales y profundas utilizadas en el reconocimiento de la plaga en el cultivo de aguaymanto.

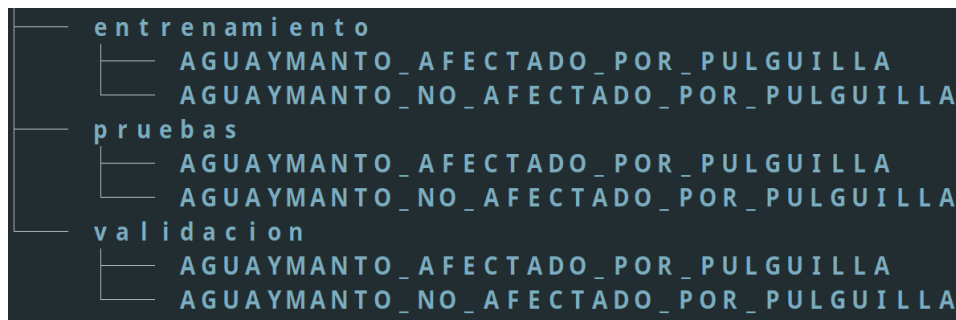


Fig. 18. Distribución de Imágenes Clasificadas por Categorías.

d. Pre-procesamiento y Región de Interés

En la fase de pre-procesamiento, se aplicaron diversas transformaciones a las imágenes capturadas del dataset para prepararlas de manera óptima para el proceso de reconocimiento y clasificación de la plaga pulguilla en el cultivo de aguaymanto.

El objetivo fundamental de aplicar el proceso de pre-procesamiento a las imágenes capturadas es lograr la segmentación efectiva de las hojas de aguaymanto, con el fin de proporcionar a la red neuronal una entrada clara y bien definida para su interpretación y análisis.

El pre-procesamiento desempeña un papel crítico al eliminar el ruido y resaltar las características clave de las imágenes, permitiendo que la red neuronal se centre en las regiones relevantes para la identificación de la plaga pulguilla. La segmentación obtenida a través de la aplicación del umbral de Otsu y las operaciones de morfología matemática asegura que solo la región de la hoja sea considerada, eliminando el fondo y cualquier distracción visual.

En última instancia, el pre-procesamiento busca optimizar las imágenes para su posterior análisis por parte de la red neuronal. Al proporcionar imágenes segmentadas y simplificadas, se facilita el proceso de aprendizaje y reconocimiento de patrones por parte de la red neuronal, lo que resulta en una mayor precisión y eficacia en la identificación de la plaga pulguilla en las hojas de aguaymanto.

A continuación, se describe detalladamente cada una de estas etapas:

- **Conversión a Escala de Grises:** La función comienza convirtiendo la imagen a escala de grises, realizamos este proceso para segmentar la hoja sin importancia de los 3 canales porque no es relevante para este paso, lo que

simplifica el procesamiento al reducir la información de color a una sola dimensión de intensidad.

La conversión de la imagen a escala de grises se logra mediante una transformación ponderada de los canales de color RGB originales. En el modelo de color RGB, cada píxel se representa como una combinación de tres componentes: rojo (R), verde (G) y azul (B). Para convertir una imagen RGB en escala de grises, se promedian estos tres componentes, ponderándolos de acuerdo con su contribución perceptual al brillo.

Matemáticamente, si tenemos un píxel con valores RGB (R, G, B), la intensidad en escala de grises (I) se puede calcular utilizando la fórmula:

$$I = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$$

Donde los coeficientes 0.299, 0.587 y 0.114 representan las sensibilidades relativas del ojo humano a los componentes rojo, verde y azul, respectivamente. Estos valores se derivan de la función de luminosidad estándar utilizada en la conversión de color a escala de grises.

Resultado



Fig. 19. Representación de hojas afectadas y no afectadas por la plaga pulguilla en escala de grises. a) Hoja afectada por la plaga. b) Hoja no afectada por la plaga.

• **Aplicación del Umbral de Otsu:** Se aplica el umbral de Otsu para separar la región de la hoja del fondo de la imagen. El umbral de Otsu es un método de binarización que busca encontrar el valor umbral óptimo que maximiza la varianza entre los objetos y el fondo de la imagen. Esto ayuda a segmentar la hoja y resaltarla en la imagen.

Para calcular el Umbral de Otsu, primero calculamos la Probabilidad $P(i)$ de que un pixel tenga un valor de intensidad i en la imagen, que se puede expresar como:

$$P(i) = \frac{n_i}{N}$$

Donde:

n_i es el número de pixeles con intensidad i .

N es el número total de pixeles en la imagen.

Luego, calculamos la media global de intensidad M_G , que se obtiene como:

$$M_G = \sum_{i=0}^{L-1} i \cdot P(i)$$

Donde L es el número de niveles de intensidad (generalmente 256 para imágenes en escala de grises).

A continuación, calculamos la varianza intraclases $V(T)$ para cada valor umbral T como:

$$V(T) = P_1(T) \cdot (M_1(T) - M_G)^2 + P_2(T) \cdot (M_2(T) - M_G)^2$$

Donde:

- $P_1(T)$ y $P_2(T)$ son las probabilidades de los grupos separados por el umbral de T .
- $M_1(T)$ y $M_2(T)$ son las medidas de intensidad de los dos grupos.

Finalmente, el Umbral de Otsu T se elige como aquel que maximiza la varianza intraclase, es decir:

$$T = \arg \max_T V(T)$$

Aplicando el Umbral de Otsu se obtienen los siguientes resultados.

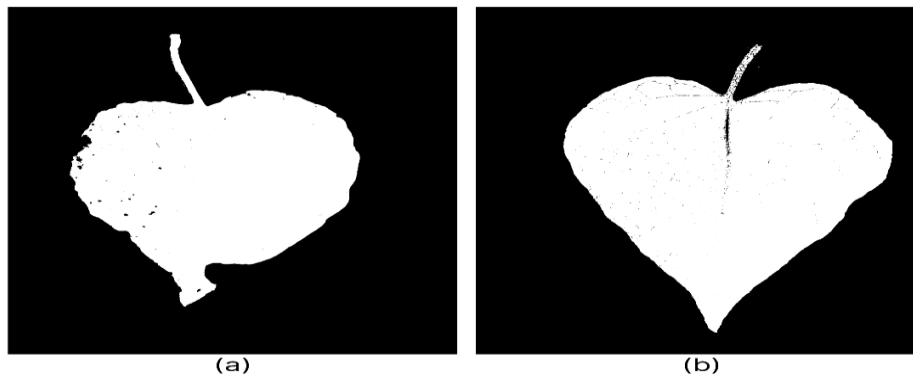


Fig. 20. Resultados de la segmentación utilizando el umbral de Otsu. a) Hoja afectada por la plaga. b) Hoja no afectada por la plaga.

En la tabla 7 se muestra la implementación en el lenguaje de programación Python.

Tabla 7. Implementación del Umbral de Otsu en Python.

```
# Aplicar el umbral de Otsu para separar la hoja del fondo
thresh = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY_INV +
cv2.THRESH_OTSU) [1]

kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (15,
15))
otsu = cv2.morphologyEx(thresh, cv2.MORPH_CLOSE, kernel,
iterations=25)
otsu = cv2.morphologyEx(otsu, cv2.MORPH_OPEN, kernel,
iterations=25)
```

Nota: *Elaboración propia*

• **Operaciones de Morfología Matemática:** Se utilizan operaciones de morfología matemática para mejorar la calidad de la segmentación. Primero se realiza una operación de cierre con un kernel elíptico para eliminar pequeños

huecos en los objetos blancos (hojas) y luego una operación de apertura para suavizar los bordes irregulares.

Cierre (C):

El cierre, denotado como C, se define como:

$$C(A, B) = (A \ominus B) \oplus B$$

Donde:

- A representa la imagen de entrada.
- B es el kernel utilizado para la operación.
- \ominus representa la erosión de A por B.
- \oplus Indica la dilatación del resultado de la erosión por B.

Apertura (o):

La apertura, denota como O, se define como:

$$O(A, B) = (A \circ B) \cdot B$$

Donde:

- A representa la imagen de entrada.
- B es el kernel utilizado para la operación.
- \circ representa la dilatación de A por B.
- \cdot indica la erosión del resultado de la dilatación por B.

Inicialmente, se empleó un kernel de tamaño 3, lo que resultó en una segmentación que incluía el tallo de la hoja, lo cual no era relevante para la detección de plagas en las hojas. Posteriormente, se probaron kernels de tamaño 5 y 7 y se observaron resultados comparables. Tras un análisis

minucioso, se optó por utilizar el kernel de tamaño 7 debido a su capacidad para mantener la integridad de las hojas y suavizar los bordes irregulares sin separar el tallo de la hoja. Esta elección se basó en la optimización del proceso de segmentación para la identificación precisa de plagas en las hojas.

El resultado se muestra a continuación.

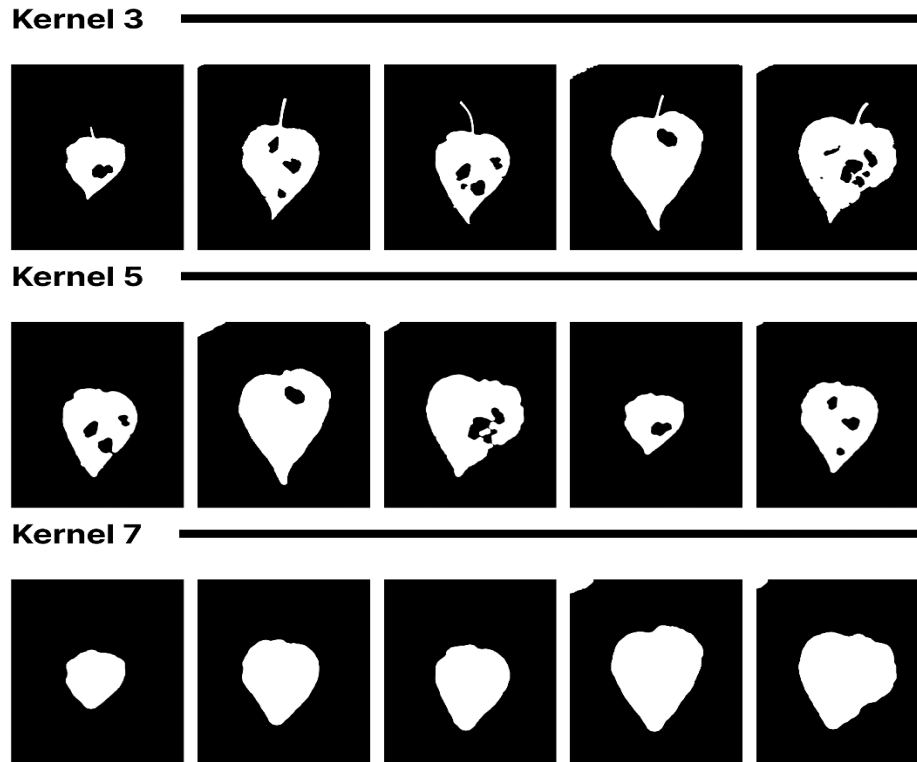


Fig. 21. Resultados de Pruebas con Diferentes Kernels en la Segmentación.

- **Eliminación del Fondo y Mejora de la Imagen de la Hoja:** La eliminación del fondo y la mejora de la imagen de la hoja se logran mediante la generación de una máscara (M) a través del proceso de morfología matemática. Esta máscara se utiliza para suprimir la porción del fondo de la imagen original (I). Es decir, esta técnica elimina la zona de color blanco que carece de relevancia, preservando únicamente la región correspondiente a la hoja. De esta manera, se logra una imagen optimizada y más clara para su posterior análisis y clasificación en la red neuronal (Figura 25).

Matemáticamente, este proceso se representa como:

$$E(I, M) = I \cdot (1 - M)$$

$$M(I, B) = (I \oplus B) - (I \circ B)$$

Donde:

- E denota la eliminación del fondo.
- M representa la mejora de la imagen de la hoja.
- I es la imagen de entrada.
- B es el kernel utilizado en el proceso de morfología matemática.
- \oplus y \circ representan la dilatación y erosión, respectivamente.

La operación $M(I, B)$ genera la máscara M, que identifica la región de la hoja en la imagen original. Luego, $E(I, M)$ se aplica para eliminar el fondo de la imagen, preservando solo la región relevante de la hoja.

Este proceso resulta en una imagen optimizada y más clara de la hoja, que se utiliza para su posterior análisis y clasificación en la red neuronal.

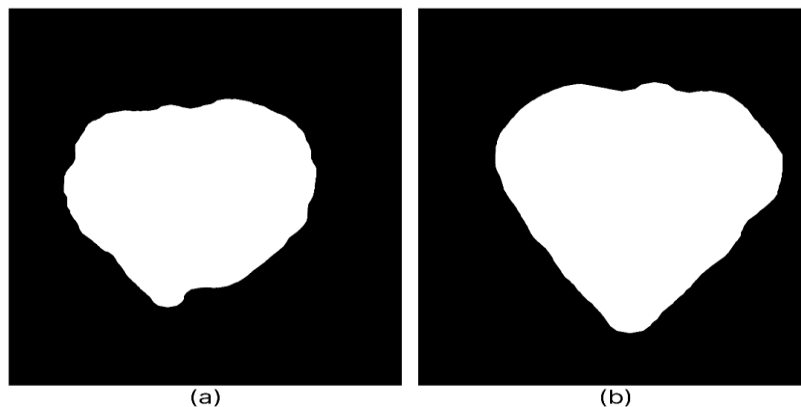


Fig. 22. Resultados de la segmentación utilizando la morfología matemática.

a) Hoja afectada por la plaga. b) Hoja no afectada por la plaga.

- **Detección de Contornos:** En esta etapa, se emplea un enfoque basado en algoritmos matemáticos para identificar y representar los bordes de las formas

presentes en la imagen segmentada de la hoja. Para lograrlo, se utilizan conceptos derivados del cálculo diferencial y la teoría de curvas. Formalmente, el contorno de una región en una imagen se define como una curva cerrada, cuyas coordenadas (x, y) varían continuamente a lo largo de la curva, denotada como $C(t) = (x(t), y(t))$, donde t es el parámetro del contorno. En términos más precisos, se busca una secuencia discreta de puntos que se aproximen al contorno real de la región. Este proceso involucra operaciones matemáticas de derivación para determinar la dirección y el grado de cambio en las intensidades de píxeles a lo largo de la curva. La detección de contornos se realiza utilizando la función `cv2.findContours` de la biblioteca OpenCV, que implementa algoritmos de seguimiento de bordes como el algoritmo de detección de Canny. El resultado final es una lista de puntos discretos que representan de manera aproximada el contorno de la hoja, permitiendo una representación matemática adecuada para su análisis subsiguiente.

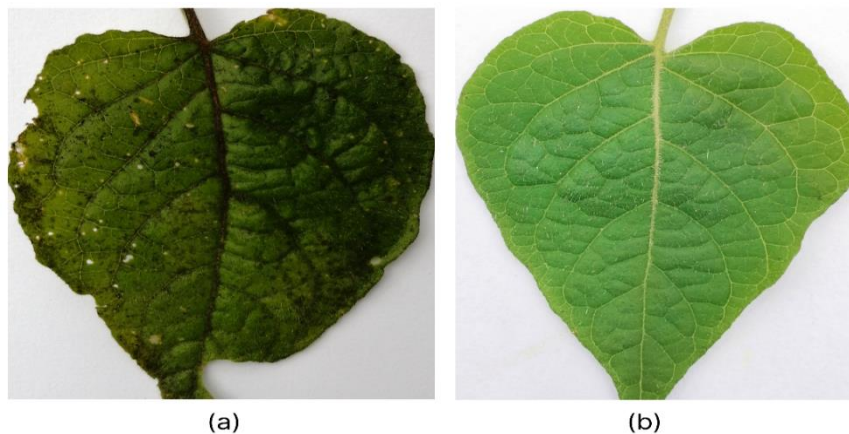


Fig. 23. Resultados de la detección de contorno con el recorte de la región de la hoja. a) Hoja afectada por la plaga. b) Hoja no afectada por la plaga.

- Después de la detección de contornos, se obtiene una imagen binaria en blanco y negro donde los contornos son blancos y el fondo es negro. Para convertir esta imagen binaria en color, se utilizó un proceso de superposición

(overlay) con la imagen original en color. Esto implica tomar la imagen original en color y asignar colores específicos a las regiones donde se detectaron contornos en la imagen binaria y con la superposición se ha logrado mediante operaciones de máscara y superposición de píxeles.

- **Recorte de la Región de la Hoja:** Se utiliza el rectángulo delimitador del contorno más grande para recortar la región de la hoja de la imagen original. Esto asegura que solo se analice la parte relevante de la imagen en la etapa de reconocimiento (Figura 26).

Tabla 8. Se muestra el código de la función que realiza el preprocesamiento para preparar el conjunto de entrenamiento y validación.

```
def my_preprocessing_func(img):  
  
    # Convertir la imagen a escala de grises  
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)  
    gray = gray.astype(np.uint8)  
  
    # Aplicar el umbral de Otsu para separar la hoja del fondo  
    thresh = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY_INV  
        cv2.THRESH_OTSU)[1]  
    kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (15,15))  
    otsu = cv2.morphologyEx(thresh, cv2.MORPH_CLOSE, kernel,  
        iterations = 25)  
    otsu = cv2.morphologyEx(otsu, cv2.MORPH_OPEN, kernel, iterations =  
        25)  
  
    # Encontrar los contornos en la imagen en escala de grises  
    contours, _ = cv2.findContours(gray_result, cv2.RETR_TREE,  
        cv2.CHAIN_APPROX_SIMPLE)  
  
    # Ordenar los contornos por área y mantener el más grande  
    contours = sorted(contours, key=cv2.contourArea, reverse=True)[:1]  
  
    leaf = None  
    # Iterar sobre los contornos y encontrar el rectángulo delimitador para  
    cada contorno  
    if contours:  
        for contour in contours:  
            x, y, w, h = cv2.boundingRect(contour)  
  
    # Usar el rectángulo delimitador para recortar la hoja de la imagen  
    leaf = img[y:y+h, x:x+w]  
    if leaf is None:
```

```
return img
```

```
return leaf
```

Fuente: *Elaboración propia.*

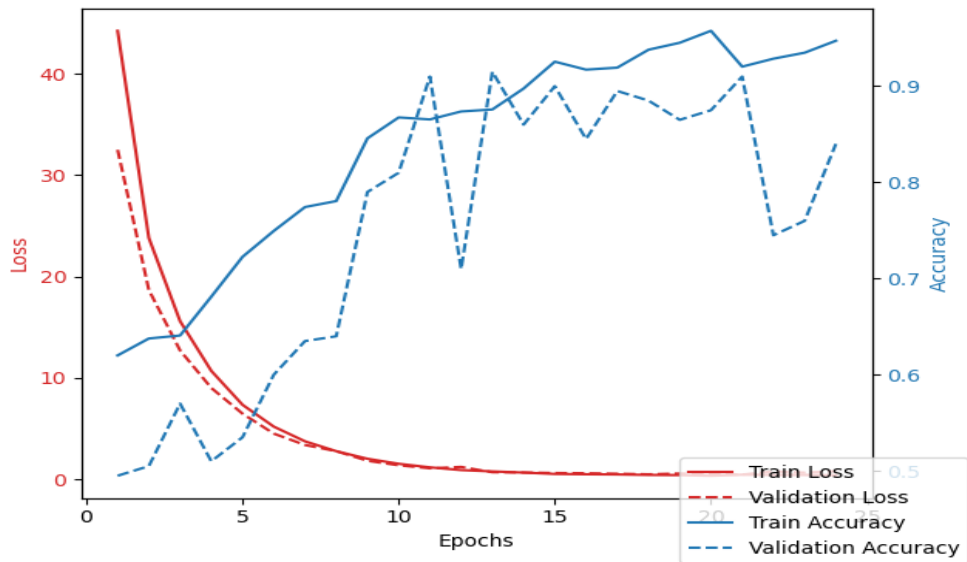


Fig. 24. Entrenamiento de la red neuronal.

3.3.2. Red neuronal profunda.

Se implementó la red neuronal convolucional profunda en el IDE Anaconda el cual permite realizar proyectos en Python de inteligencia artificial o de ciencia de datos y el editor Jupyter ya que permitió trabajar línea a línea la fase de entrenamiento y realizar las pruebas para llegar a un buen entrenamiento. Se utilizó Transfer Learning para las redes neuronales convolucionales VGG16, VGG19 y MobileNet, ya que son un modelo pre-entrenado. A continuación, explicamos en detalle el procedimiento.

- a. **Librerías.** Luego de elaborar el dataset se exportaron las librerías a utilizar tensorflow, keras, pandas, numpy, mlxtend, os e itertools en el IDE Anaconda y el editor Jupyter.

Tabla 9. Lista de Librerías exportadas en el editor Jupyter del IDE Anaconda.

```

import pandas as pd
import numpy as np
import os
import keras
import matplotlib.pyplot as plt
import tensorflow as tf
from keras import applications
from keras.utils import to_categorical
from keras.layers import Dense, GlobalAveragePooling2D
from keras.layers.convolutional import Conv2D
from keras.layers import Dense, Dropout, Convolution2D,
MaxPooling2D
from keras.layers.convolutional import MaxPooling2D
from keras.layers import Flatten
from keras.applications import MobileNet
from keras.preprocessing import image
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Model, Sequential
from keras.optimizers import Adam
from mlxtend.evaluate import confusion_matrix
from keras import backend as K
import itertools
import cv2
from keras.callbacks import EarlyStopping
from keras.optimizers import Adam
import matplotlib.pyplot as plt

```

Fuente: *Elaboración propia.*

b. Arquitectura de la red neuronal

La arquitectura de la red que se ha diseñado es una red neuronal convolucional (CNN) profunda que consta de varias capas:

Tabla 10. *Arquitectura de la red neuronal propio.*

```

from keras.layers import BatchNormalization
from keras.regularizers import l2, l1

seed = 5

tam = 224

model = Sequential()

# Convolutional Layer 1
model.add(Convolution2D(32, (3, 3), padding='same',
activation='relu', input_shape=(tam, tam, 3)))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))

# Convolutional Layer 2

```

```

model.add(Convolution2D(64, (3, 3), padding='same',
activation='relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))

# Convolutional Layer 3
model.add(Convolution2D(128, (3, 3), padding='same',
activation='relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))

# Convolutional Layer 4
model.add(Convolution2D(256, (3, 3), padding='same',
activation='relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))

# Convolutional Layer 5
model.add(Convolution2D(512, (3, 3), padding='same',
activation='relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))

# Convolutional Layer 6
model.add(Convolution2D(1024, (3, 3), padding='same',
activation='relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))

# Flattening the layers
model.add(Flatten())

# Fully connected layer 1
model.add(Dense(512, activation='relu',
kernel_regularizer=l2(0.01)))
model.add(Dropout(0.5))

# Fully connected layer 2
model.add(Dense(256, activation='relu',
kernel_regularizer=l1(0.01)))
model.add(Dropout(0.5))

# Output Layer
model.add(Dense(1, activation='sigmoid'))

```

Fuente: *Elaboración propia.*

- **Capas Convolucionales:** Se diseñaron 6 capas convolucionales con un número creciente de filtros (32, 64, 128, 256, 512, 1024). Cada capa convolucional utiliza un filtro de 3x3, 'same' padding (lo que significa que la salida tiene la misma dimensión que la entrada) y función de activación ReLU. Después de cada capa convolucional,

se aplica una capa de normalización de lotes (BatchNormalization) seguida de una capa de MaxPooling con un tamaño de pool de 2x2.

- **Capa de Aplanamiento:** Después de las capas convolucionales, se aplanan las salidas antes de pasarlas a las capas completamente conectadas.
- **Capas Completamente Conectadas (Dense):** Se diseñaron dos capas completamente conectadas, la primera con 512 neuronas y la segunda con 256 neuronas. Ambas utilizan la función de activación ReLU. También se añadió regularización L2 y L1 a estas capas, respectivamente, para ayudar a prevenir el sobreajuste. Después de cada capa densa, se aplica una capa de abandono (Dropout) con una tasa de 0.5.
- **Capa de Salida:** La capa de salida es una capa densa con una sola neurona y utiliza la función de activación sigmoide. Esto sugiere que se está realizando una clasificación binaria.

En resumen, esta arquitectura parece adecuada para una tarea de clasificación de imágenes. La normalización de lotes ayudará a acelerar el entrenamiento y la regularización y el Dropout ayudarán a prevenir el sobreajuste. Sin embargo, hay algunas cosas que debemos considerar:

- **Regularización:** Se ha utilizado regularización L1 y L2 en las capas completamente conectadas, lo cual es una buena práctica. Esto ayuda a evitar el sobreajuste en la red.
- **Tamaño de la Imagen:** Se utilizó un tamaño de imagen de 224x224, lo cual es bastante estándar.

Preparando el conjunto de datos con la función de preprocesamiento para alimentar a la red neuronal.

- **ImageDataGenerator**: Esta es una clase de Keras que genera lotes de datos de imágenes con aumento de datos en tiempo real. El aumento de datos es una técnica que se utiliza para aumentar la cantidad de datos de entrenamiento. Esto se hace aplicando varias transformaciones (como rotaciones, cambio de escala, etc.) a las imágenes existentes para crear nuevas imágenes. Esto ayuda a hacer el modelo más robusto y generalizar mejor.
- **rescale=1./255**: Esto es para normalizar las imágenes. Las imágenes están compuestas por valores de píxeles que van de 0 a 255. Al dividir cada píxel por 255, los valores de los píxeles se normalizan para estar en el rango de 0 a 1. Esto ayuda a acelerar el entrenamiento y reducir el riesgo de problemas de optimización.
- **my_preprocessing_func**: Esto indica que se debe aplicar la función de preprocesamiento personalizada a cada imagen.
- **flow_from_directory**: Este método toma el path a un directorio y genera lotes de datos aumentados. Los datos se cargan en lotes de un tamaño 10 desde el directorio especificado.
- **target_size=(tam,tam)**: Esto redimensiona todas las imágenes a un tamaño específico, en este caso, 224x224 píxeles.
- **color_mode='rgb'**: Esto indica que las imágenes son en color y deben cargarse con tres canales (rojo, verde, azul).

- **batch_size=10**: Esto especifica el número de imágenes que deben incluirse en cada lote. En este caso, cada lote tendrá 10 imágenes.
- **class_mode='binary'**: Esto indica que se trata de un problema de clasificación binaria, es decir, cada imagen pertenece a una de dos clases.
- **shuffle=True**: Esto indica que los datos deben ser mezclados antes de cada época.
- **seed=seed**: Esto establece la semilla para la mezcla de los datos. Esto es importante para garantizar la reproducibilidad de los resultados.
- **Generador de Datos de Validación**: Es similar al generador de datos de entrenamiento, pero se aplica al conjunto de datos de validación. La validación es importante para evaluar cómo el modelo está aprendiendo y ajustar los hiperparámetros si es necesario.
- **print(train_generator.class_indices)**: Esto imprime un diccionario que mapea las etiquetas de clase (como strings) a los números enteros que el modelo utilizará internamente. Esto es útil para saber cómo se han codificado las clases.

En resumen, este código carga y preprocesa las imágenes antes de entrenar el modelo.

Tabla 11. Se muestra el código sobre el procesamiento de los datos para alimentar a la red neuronal.

```
train_datagen=ImageDataGenerator(rescale=1./255,
preprocessing_function=my_preprocessing_func)

train_generator=train_datagen.flow_from_directory(data_entrenamiento,
target_size=(tam,tam),
color_mode='rgb',
batch_size=10,
class_mode='binary',
shuffle=True,
```

```
seed=seed)
```

```
validacion_datagen = ImageDataGenerator(rescale=1./255,  
preprocessing_function=my_preprocessing_func)  
  
valid_generator = validacion_datagen.flow_from_directory(  
    data_validacion,  
    target_size=(tam,tam),  
    color_mode='rgb',  
    batch_size=10,  
    class_mode='binary',  
    shuffle=True,  
    seed=seed)
```

```
print(train_generator.class_indices)
```

Fuente: *Elaboración propia.*

Definimos la tasa de aprendizaje que permite controlar la magnitud de los ajustes realizados en los pesos de la red neuronal durante el proceso de entrenamiento y el número total de épocas indica cuántas veces el conjunto de datos completo se utilizará para entrenar la red. Utilizamos el optimizador **Adam** que es responsable de ajustar los pesos de la red neuronal en función de la función de pérdida durante el entrenamiento.

En la línea **model.compile** permite configurar y compilar el modelo, se especifica el optimizador Adam, la función de pérdida que es “**binary_crossentropy**” que es adecuado para problemas de clasificación binaria y las métricas que se utilizarán para evaluar el rendimiento durante el entrenamiento “precisión”.

Calculamos el número de pasos por épocas para el conjunto de entrenamiento y validación; se crean instancias de dos devoluciones de llamada **callbacks** y **EarlyStopping**, esto permite monitoriar la función de pérdida y controlar el sobreajuste, en caso no mejora durante un número determinado de épocas (en este caso, 3) se detiene y termina el entrenamiento. Así mismo, guardamos el modelo en un archivo con formato **h5**. Finalmente, en la línea **fit** iniciamos el proceso de entrenamiento real donde colocamos las variables definidas.

Tabla 12. Número de pasos por épocas para el conjunto de entrenamiento y validación.

```
# Define una tasa de aprendizaje personalizada
TOTAL_EPOCAS_ENTRENAMIENTO = 30
learning_rate = 0.001

# Crea una instancia del optimizador Adam con tu tasa de aprendizaje
optimizer = Adam(learning_rate=learning_rate)
model.compile(optimizer=optimizer , loss='binary_crossentropy',
metrics=['accuracy'])

step_size_train=train_generator.n//train_generator.batch_size
step_size_valid=valid_generator.n//valid_generator.batch_size

early_stopping = EarlyStopping(monitor='val_loss', patience=3,
restore_best_weights=True)
checkpoint_cb = ModelCheckpoint("nuevadata.h5", save_best_only=True)

H=model.fit(x=train_generator,
            steps_per_epoch=step_size_train,
            epochs=TOTAL_EPOCAS_ENTRENAMIENTO,
            validation_data=valid_generator,
            validation_steps=step_size_valid,
            callbacks=[early_stopping, checkpoint_cb])
```

Fuente: *Elaboración propia.*

IV. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

Durante el desarrollo de esta investigación, se lograron avances significativos en la adquisición y procesamiento de imágenes para el análisis de las hojas del cultivo de aguaymanto. A continuación, se resaltan los principales hallazgos:

- **Protocolo de Captura de Imágenes:** Se estableció un protocolo de adquisición de imágenes que considera las condiciones ambientales y técnicas. En entornos no controlados, se determinó que las capturas deben realizarse en días soleados entre las 10:00 y las 11:00 am, a una distancia de 10 a 50 cm, utilizando una cámara de 16 megapíxeles. Para entornos controlados, se diseñó una caja de 25 x 30 cm para mantener las hojas a una distancia de 20 cm y a un ángulo de 90 grados, igualmente empleando una cámara de 48 megapíxeles.
- **Base de Datos de Imágenes:** Se recopilaron un total de 1666 imágenes de hojas de aguaymanto en condiciones controladas, tanto afectadas como no afectadas, con las cuales se constituyó una valiosa base de datos para el análisis.
- **Procesamiento de Imágenes:** Se exploraron y aplicaron diversas técnicas de procesamiento de imágenes, como el redimensionamiento, la obtención de canales de imagen, la conversión a escala de grises, la normalización, la manipulación del tamaño y el zoom de la imagen, así como operaciones de espejo y rotación. Estos procedimientos se revelaron esenciales para preparar las imágenes antes del análisis.
- **Selección de Algoritmos y Resultados:** Entre los algoritmos de clasificación evaluados, las redes neuronales convolucionales (CNN) y las redes neuronales profundas se destacaron por su eficacia. Los resultados obtenidos revelaron una precisión del 88% para las redes neuronales profundas y un 81-83% para las redes neuronales convolucionales en la tarea de clasificación.
- **Herramientas y Bibliotecas Utilizadas:** La implementación de los algoritmos se realizó en el entorno Python 3.9, utilizando el IDE Anaconda y el editor Jupyter notebook. Se

emplearon diversas bibliotecas, incluyendo tensorflow, keras, pandas, numpy, mlxtend, os e itertools.

4.2. Recomendaciones

Basándonos en los hallazgos de esta investigación, se formulan las siguientes recomendaciones:

- **Desarrollo de Herramientas Computacionales:** Dado el potencial de la visión artificial en el sector agrícola, se recomienda continuar con el desarrollo de herramientas computacionales para el análisis de imágenes de cultivos. Esto permitirá una mayor precisión y eficiencia en la detección temprana de enfermedades y el monitoreo de la salud de las plantas.

REFERENCIAS

- 1] FAO, «Oficina Regional de la FAO para América Latina y el Caribe,» 14 Setiembre 2018. [En línea]. Available: [view-source:http://www.fao.org/americas/noticias/ver/es/c/1152160/#:~:text=Empleo-,FAO%3A%20Hambre%20aumenta%20en%20el%20mundo%20y%20en%20Am%C3%A9rica%20Latina,la%20desnutrici%C3%B3n%20infantil%20sigue%20disminuyendo..](http://www.fao.org/americas/noticias/ver/es/c/1152160/#:~:text=Empleo-,FAO%3A%20Hambre%20aumenta%20en%20el%20mundo%20y%20en%20Am%C3%A9rica%20Latina,la%20desnutrici%C3%B3n%20infantil%20sigue%20disminuyendo..)
- 2] FAO, «FAO EN PERÚ,» 21 Enero 2020. [En línea]. Available: <http://www.fao.org/peru/fao-en-peru/peru-en-una-mirada/es/>.
- 3] H. Nagamani y D. Saroja, «Métodos de investigación para la detección de sanidad vegetal mediante visión artificial,» *Revista Internacional de Investigación de Ingeniería Aplicada* , vol. 14, nº 7, pp. 1-6, 2019.
- 4] . L. Rouhiainen, Inteligencia artificial, España: Editorial Planeta, 2018.
- 5] Ikusmen, Introducción a la visión artificial, Mexico: Cognex, 2019.
- 6] Real Academia de la Lengua Española, «Real Academia de la Lengua Española,» 24 Junio 2020. [En línea]. Available: <https://dle.rae.es/fruta>.
- 7] Ministerio de Agricultura, «MINAGRI,» 12 Julio 2020. [En línea]. Available: <https://www.minagri.gob.pe/portal/32-sector-agrario/frutas#:~:text=Frutas,or%C3%ADgenes%20y%20periodos%20de%20comercializaci%C3%B3n..>
- 8] D. Guerrero, C. Sandoval, N. Coronado, C. Rodriguez y K. Saavedra, «DISEÑO DE LA LÍNEA DE PRODUCCIÓN PARA LA OBTENCIÓN Y ENVASADO DE NÉCTAR DE AGUAYMANTO,» Piura, Repositorio Institucional Universidad de

Piura, 2012.

9] MINAGRI, Estudio de pre factibilidad para la producción y comercialización de aguaymanto en condiciones de valles andinos, Perú: MINAGRI, 2018.

10] H. F. Vega Huerta, Redes neuronales para el reconocimiento de la calidad morfológica de mangos exportables para la empresa Biofruit del Perú S.A.C., Lima-Perú: Repositorio de la Universidad Nacional Federico Villarreal, 2011.

11] C. A. Olvera Olvera y C. E. Galván Tejada, Comparación de arquitecturas de redes neuronales convolucionales para la clasificación de enfermedades en tomate, Zacatecas: Repositorio de la Universidad Autónoma de Zacatecas, 2019.

12] C. A. Olvera Olvera y C. E. Galván Tejada, Comparación de arquitecturas de redes neuronales convolucionales para la clasificación de enfermedades en tomate, Zacatecas: Repositorio de la Universidad Autónoma de Zacatecas, 2019.

13] I. Noor, R. Zaman Khan y H. Mokhtar Mohammed, «Comprensión de los modelos de color: una revisión,» *ARNP Revista de Ciencia y Tecnología*, p. 2, 2012.

14] H. Ñaupas Pitán , «Metodoloiga de la investigación y asesoramiento de tesis,» Lima- Perú, Gráfica Reta S.A.C., 2009, p. 62.

15] G. . T. CASTRO ALVAREZ, Aplicación de algoritmos inteligentes para el reconocimiento automático de enfermedades foliares de cultivo de palta, Ilo- Perú: Repositorio de la Universidad de Moquegua, 2019.

16] C. E. Galvan Tejada y C. A. Olvera Olvera, Comparación de arquitecturas de redes neuronales convolucionales para la clasificación de enfermedades en tomate, Zacatecas: Universidad Autónoma de Zacatecas, 2019.

17] F. L. Becerra Suárez, «PATRONES DE CONDUCTA FACIAL, PARA IDENTIFICAR ACCESOS INFORMÁTICOS NO AUTORIZADOS,» Pimentel - Perú, 2019.

I. Mejia, R. Ramirez, H. Jimenez y J. Rosas, «A new method a architecture

- 18] entreprise,» *Conference IEEE bussines*, pp. 200-215, 2019.
- H. Jimenez, R. Rodriguez y J. Tiparra, Diagnóstico de TEA, Madrid:
19] Latinoamérica SA, 1978.
- P. SZNAJDLEDER, Java a fondo - estudio del lenguaje y desarrollo de
20] aplicaciones - 2a ed., México: Alfaomega, 2012.
- I. Mejia, M. Tuesta y M. Forero, «A new method of enterprise archicture small
21] organizations,» *Computer Science Techology*, pp. 150-170, 2020.
- J. Palma y R. Marín, Inteligencia Artificial, Madrid: McGrawHill, 2008, pp. 3-8.
22]
- K. Rojas, «Identificación de efectos negativos de la TEA en el aprendizaje,»
23] *IEEE conference Techology children especial*, pp. 200-215, 2018.
- H. Nagamani y D. Saroja, «Métodos de investigación para la detección de
24] sanidad vegetal mediante visión artificial,» de *Revista Internacional de Investigación
de Ingeniería Aplicada*, India, 2019.
- MIDAGRI, «Frutas,» 2015. [En línea]. Available:
25] <https://www.midagri.gob.pe/portal/32-sector-agrario/frutas>.
- S. Cecilia, C. Molina, C. Rodríguez y T. Fernández, «Detección de
26] enfermedades en hojas de arándano usando computadora Visión y técnicas de
aprendizaje automático,» *Revista Internacional de Aprendizaje Automático y
Computación*, pp. 1-6, 2019.
- L. Yin y L. Changhua, «Investigación sobre el método de aprendizaje profundo
27] de la enfermedad de los cultivos,» *Association for Computing Machinery*, pp. 1-6,
2019.
- S. Davinder, J. Naman, J. Pranjali, K. Pratik, K. Sudhakar y B. Nipun, «Un
28] conjunto de datos para la detección visual de enfermedades de plantas,» de *En
Séptimo ACM IKDD CoDS y 25º COMAD (CoDS COMAD 2020)*, Nueva York, NY, EE.

UU, 2020.

Á. O. González Frayre, C. A. Guerrero Méndez, C. A. Olvera Olvera, F. E.
29] López Monteagudo, D. López Betancur, V. Maeda Gutiérrez y S. Villagrana Barraza,
«Estudio y comparativa de algoritmos de detección de objetos con redes neuronales
artificiales convolucionales para la detección de enfermedades en hojas,» *Unidad
Académica de Ingeniería Eléctrica, Universidad Autónoma de Zacatecas*, pp. 1-6,
2019.

J. Madhu, A. Rashi , S. Raghuraj y K. Shilpa D. , «Una revisión sobre las
30] técnicas de procesamiento de imágenes digitales para la detección, cuantificación e
identificación automáticas de Enfermedades de las plantas,» *Revista internacional de
aplicaciones informáticas (0975 - 8887)*, pp. 1-6, 2017.

P. Jagadeesh D. , Y. Rajesh y S. B. Abdulmunaf, «Clasificación basada en
31] SVM y ANN de enfermedades de las plantas utilizando la técnica de reducción de
características,» *Revista Internacional de Multimedia Interactiva e Inteligencia
Artificial*, pp. 1-9, 2016.

G. Bhagyashri S. y B. Gajanan K. , «Computer vision enfoque basado en
32] detectar enfermedades de las hojas de arroz utilizando descriptores de textura y
color,» de *Actas de la Conferencia Internacional sobre la Inventiva Computación e
Informática (ICICI 2017) IEEE Xplore Cumbre*, Maharashtra, India., 2017.

A. Halimatu Sadiyah , S. Ray E. y F. Mahieddine, «Convolución red neuronal
33] en la agricultura de precisión para el reconocimiento de la planta y clasificación de
imágenes,» *Universidad de Bradford*, pp. 1-8, 2017.

A. Arib Alfarisy, Q. Chen y M. Guo, «Clasificación Aprendizaje Basado en lo
34] Profundo de Arroz Plagas y enfermedadesReconocimiento,» *Association for
Computing Machinery*, pp. 1-5, 2018.

S. Apurva, P. Shikha, S. Swati, K. Khushboo y S. S. Sitanshu, «Detección de

35] plagas mediante la segmentación de imágenes basada en colores,» de *2da Conferencia Internacional sobre Comunicación Inventiva y Tecnologías Computacionales (ICICCT 2018)* , China, 2018.

H. Ñaupas Paitan, «Metodología de la investigación científica y asesoramiento de tesis,» Lima-Perú, Gráfica Reatai S.A, 2009, p. 215.

E. D. Lucero Carrillo y H. A. Saldaña Saldaña, Utilización de técnicas de visión artificial para la detección automática de defectos externos del mango, Chiclayo - Perú: Repositorio de la Universidad Señor de Sipan, 2016.

S. Cecilia, C. Molina, C. Rodríguez y T. Fernández, «Detección de enfermedades en hojas de arándano usando computadora Visión y técnicas de aprendizaje automático,» *Revista Internacional de Aprendizaje Automático y Computación*, pp. 1-6, 2019.

N. Velázquez López, Y. Sasaki, K. Nakano, J. M. Mejía-Muñoz⁴ y E. Romanchik Kriuchkova, «Detección de cenicilla en rosa usando procesamiento de imágenes por computadora,» *Revista Chapingo Serie Horticultura*, pp. 1-10, 2011.

C. Flórez, C. Andrés, A. Hurtado, S. Ramos y O. Lucía, «Procesamiento de imágenes para reconocimiento de daños Procesamiento de imágenes para reconocimiento de daños (flor de azúcar),» de *Acta Agronómica*, Colombia , 2015.

J. Ardepolla, M. J. Reymar Cortez, A. L. Escorpion y J. J. Adtoon, «Identificación y clasificación de la calidad de exportación Carabao Mangos con procesamiento de imágenes,» *Association for Computing Machinery*, pp. 1-5, 2019.

J. Berger, C. Preussler, J. P. Agostini y , «Identificación de síntomas de Huanglongbing en hojas de cítricos mediante técnicas de deep learning,» *Congreso Argentino de AgroInformática*, pp. 1-14, 2016.

V. Cañedo y J. Kroschel, Manejo integrado de plagas/agroecología, Peru: Fontagro, 2016 .

J. R. Chancafe Sirlopu y G. A. Mazabel Quijandria , «Deteccion automática de
44] caries utilizando reconocimiento de patrones en placas radiográficas,» Chiclayo - Perú,
2016.

M. A. M. Villegas Vega y W. J. Chapoñan Santisteban , «RECONOCIMIENTO
45] DE IMPUREZAS DE AISLADORES ELÉCTRICOS EN IMÁGENES DIGITALES,»
Pimentel – Perú, 2018.

ANEXOS

Anexo 1. Resolución de aprobación del proyecto de investigación



FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO RESOLUCIÓN N°0746-2023/FIAU-USS

Pimentel, 14 de noviembre de 2023

VISTO:

El Acta de reunión N°11011-2023 del Comité de investigación de la INGENIERÍA DE SISTEMAS remitida mediante oficio N° 0270-2023/FIAU-IS-USS de fecha 13 de noviembre de 2023, y;

CONSIDERANDO:

Que, de conformidad con la Ley Universitaria N° 30220 en su artículo 48º que a letra dice: "La investigación constituye una función esencial y obligatoria de la universidad, que la fomenta y realiza, respondiendo a través de la producción de conocimiento y desarrollo de tecnologías a las necesidades de la sociedad, con especial énfasis en la realidad nacional. Los docentes, estudiantes y graduados participan en la actividad investigadora en su propia institución o en redes de investigación nacional o internacional, creadas por las instituciones universitarias públicas o privadas.";

Que, de conformidad con el Reglamento de grados y títulos en su artículo 21º señala: "Los temas de trabajo de investigación, trabajo académico y *tesis* son *aprobados por el Comité de Investigación* derivados a la facultad o Escuela de Posgrado, según corresponda, para la emisión de la resolución respectiva. El *periodo de vigencia de los mismos será de dos años*, a partir de su aprobación. En caso un tema perdiera vigencia, el Comité de Investigación evaluará la ampliación de la misma.

Que, de conformidad con el Reglamento de grados y títulos en su artículo 24º señala: La tesis es un estudio que debe denotar rigurosidad metodológica, originalidad, relevancia social, utilidad teórica y/o práctica en el ámbito de la escuela profesional. Para el grado de doctor se requiere una tesis de máxima rigurosidad académica y de carácter original. Es individual para la obtención de un grado; *es individual o en pares para obtener un título profesional*. Asimismo, en su artículo 25º señala: "El tema debe responder a alguna de las líneas de investigación institucionales de la USS S.A.C.".

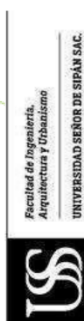
Que, mediante documentos de vistos, el Comité de investigación de la referida Escuela profesional acordó aprobar tema de las tesis, que se detallan en el Acta de reunión N° 1409 - 2023, de la línea de investigación de CIENCIAS DE LA INFORMACIÓN COMO HERRAMIENTAS MULTIDISCIPLINARES Y ESTRATÉGICAS EN EL CONTEXTO INDUSTRIAL Y DE ORGANIZACIONES, a cargo de los estudiantes y /o egresados del Programa de estudios INGENIERÍA DE SISTEMAS , hasta la fecha que indica la presente resolución.

Estando a lo expuesto, y en uso de las atribuciones conferidas y de conformidad con las normas y reglamentos vigentes;

SE RESUELVE:

ARTÍCULO 1: APROBAR, tema de Tesis a cargo de los estudiantes y /o egresados del Programa de estudios de INGENIERÍA DE SISTEMAS que se detallan en el anexo de la presente Resolución.

ARTÍCULO 2: DEJAR SIN EFECTO, toda Resolución emitida por la Facultad que se oponga a la presente Resolución.



**FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO
RESOLUCIÓN N°0746-2023/FIAU-USS**

Pimentel, 14 de noviembre de 2023

ANEXO

APROBAR TEMA DE TESIS

	APellidos y Nombres	TESIS
1	BURGOS VARGAS DANTE ATILANO	detección y clasificación de plaga pulguilla en el cultivo de aguaymanto mediante redes neuronales
2	HUAMAN CORTEZ OSCAR JULIO	Evaluación de calidad del software de licencias de funcionamiento según norma ISO/IEC 9126 en la municipalidad provincial de Chiclayo
3	RABINES PANDURO JHANET	Evaluación de métodos de usabilidad para medir calidad en uso en aplicaciones web de pequeñas empresas
4	PISFIL CHUMIOQUE LUIGI JAIR	Modelo de gestión de riesgos de seguridad de la información para reducir vulnerabilidades en el grupo VESERGENPERU E.I.R.L.
5	RIVADENEIRA PURISACA KARINA MERCEDES	modelo de gestión de la calidad de los procesos de prueba de software basado en la norma ISO 29119-2

REGÍSTRESE, COMUNÍQUESE Y ARCHÍVESE





Dr. Víctor Alexi Tuesta Monteza
Decano (E) / Facultad de Ingeniería,
Arquitectura y Urbanismo
UNIVERSIDAD SEÑOR DE SIPÁN SAC.




Dr. Halyn Alvarez Vásquez
Secretario Académico Facultad de
Ingeniería, Arquitectura y Urbanismo
UNIVERSIDAD SEÑOR DE SIPÁN SAC.

Cc: Interesado, Archivo

Anexo 2. Carta de aceptación de la institución para la recolección de datos.

 UNIVERSIDAD SEÑOR DE SIPÁN	SISTEMA DE GESTIÓN DE LA CALIDAD - USS		
	GUÍA	Código:	F-PC-USS
	DE PRODUCTOS ACREDITABLES DE LAS ASIGNATURAS DE INVESTIGACIÓN	Versión:	00
		Hoja:	53 de 52

AUTORIZACIÓN PARA EL RECOJO DE INFORMACIÓN

Ciudad de Catilluc, 17 de noviembre del 2023

Quien suscribe:

Sr. JAIME BECERRA ROMERO – DIRECTOR

Representante Legal – I.E. “Túpac Amaru”

AUTORIZA: Permiso para recojo de información pertinente en función del proyecto de investigación, denominado: “Detección y clasificación de plaga pulguilla en el cultivo de aguaymanto mediante redes neuronales”

Por el presente, el que suscribe, señor Prof. JAIME BECERRA ROMERO, representante legal de la Institución Educativa Secundaria “Túpac Amaru”, AUTORIZO al alumno Dante Atilano Burgos Vargas, identificado con DNI N°46999383, estudiante de la Escuela Profesional de Ingeniería de Sistemas, y autor del trabajo de investigación denominado “Detección y clasificación de plaga pulguilla en el cultivo de aguaymanto mediante redes neuronales”, al uso de dicha información que conforma el expediente técnico así como hojas de memorias, cálculos entre otros como planos para efectos exclusivamente académicos de la elaboración de tesis de investigación, enunciada líneas arriba de quien solicita se garantice la absoluta confidencialidad de la información solicitada.

Atentamente.


MINISTERIO DE EDUCACIÓN
DIRECCIÓN REGIONAL DE EDUCACIÓN
UGEL SAN MIGUEL
Lic. Jaime Becerra Romero
DIRECTOR
I.E. "TUPAC AMARU"

NOMBRE DEL TRABAJO

Burgos Vargas Dante Atilano-Turnitin.docx

AUTOR

Dante Burgos Vargas

RECuento DE PALABRAS

10614 Words

RECuento DE CARACTERES

60391 Characters

RECuento DE PÁGINAS

61 Pages

TAMAÑO DEL ARCHIVO

4.0MB

FECHA DE ENTREGA

Jan 11, 2024 11:01 AM GMT-5

FECHA DEL INFORME

Jan 11, 2024 11:02 AM GMT-5

● **19% de similitud general**

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

- 14% Base de datos de Internet
- Base de datos de Crossref
- 16% Base de datos de trabajos entregados
- 4% Base de datos de publicaciones
- Base de datos de contenido publicado de Crossref

● **Excluir del Reporte de Similitud**

- Material bibliográfico
- Coincidencia baja (menos de 8 palabras)
- Material citado