



Esta obra está bajo una [Licencia Creative Commons Atribución - 4.0 Internacional \(CC BY 4.0\)](https://creativecommons.org/licenses/by/4.0/)

Vea una copia de esta licencia en <https://creativecommons.org/licenses/by/4.0/deed.es>





FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

Tesis

Reconocimiento de patrones visuales para detección de enfermedades en plantaciones de arroz del sector Mishquiyacu - Morales

Para optar el título profesional de Ingeniero de Sistemas e Informática

Autor:

Gabriela Abigail Sangama Vela

<https://orcid.org/0000-0003-3948-0470>

Asesor:

Ing. Dr. Miguel Ángel Valles Coral

<https://orcid.org/0000-0002-8806-2892>

Tarapoto, Perú

2023



FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

Tesis

**Reconocimiento de patrones visuales para
detección de enfermedades en plantaciones de
arroz del sector Mishquiyacu - Morales**

Para optar el título profesional de Ingeniero de Sistemas e Informática

Autor:

Gabriela Abigail Sangama Vela

Sustentado y aprobado el 13 de octubre del 2023, por los jurados:

Presidente de Jurado
Lic.M.Sc.Edwin Augusto Hernández
Torres

Secretario de Jurado
Ing. John Clark Santa María
Pinedo

Vocal de Jurado
Ing. Carlos Armando Ríos López

Tarapoto, Perú

2023



Universidad Nacional de San Martín
Facultad de Ingeniería de Sistema e Informática
Jr. Vía Universitaria S/Nº - Ciudad Universitaria - Morales
Teléf. (042) 525688 - 524074 - Anexo 109



ACTA DE SUSTENTACIÓN PARA OPTAR EL TÍTULO DE INGENIERO DE SISTEMAS E INFORMÁTICA

En los ambientes del Aula Magna de la Facultad de Ingeniería de Sistemas e Informática de la Universidad Nacional de San Martín, a las 10:00 horas del día viernes 13 de octubre del año 2023, se reunieron los **miembros del Jurado Calificador**, integrado por:

Presidente : LIC. M.Sc. EDWIN AUGUSTO HERNANDEZ TORRES
Secretario : ING. JOHN CLARK SANTA MARIA PINEDO
Vocal : ING. CARLOS ARMANDO RIOS LOPEZ

Para evaluar la Tesis: "RECONOCIMIENTO DE PATRONES VISUALES PARA DETECCIÓN DE ENFERMEDADES EN PLANTACIONES DE ARROZ DEL SECTOR MISHQUIYACU - MORALES" presentada por la Bachiller **GABRIELA ABIGAIL SANGAMA VELA**, participando en calidad de asesor el Ing. Dr. Miguel Ángel Valles Coral.

Los señores miembros del Jurado, después de haber atendido la sustentación y evaluada las respuestas a las preguntas formuladas y terminada la réplica; luego de debatir entre sí, reservada y libremente lo declaran aprobado, por unanimidad, con el calificativo de muy bueno, equivalente a diecisiete (17), en fe de lo cual firmamos la presente acta, siendo las 11:10 horas del mismo día, con lo que se dio por terminado el Acto de Sustentación.

.....
LIC. M.Sc. EDWIN AUGUSTO HERNANDEZ
TORRES
Presidente

.....
ING. JOHN CLARK SANTA MARIA
PINEDO
Secretario

.....
ING. CARLOS ARMANDO RIOS LOPEZ
Vocal

Constancia de asesoramiento

El que suscribe el presente documento, Ing. Dr. Miguel Ángel Valles Coral.

Hace constar:

Que, he revisado la tesis titulada: Reconocimiento de patrones visuales para detección de enfermedades en plantaciones de arroz del sector Mishqiyacu - Morales en fechas del cronograma a fin de optimizar y agilizar la investigación, elaborada por:

Bachiller en Ingeniería de Sistemas e Informática: Gabriela Abigail Sangama Vela.

La que encuentro conforme en estructura y en contenido. Por lo que doy mi conformidad para fines que estime conveniente, y para que conste, firmo en la ciudad de Tarapoto.

Tarapoto, 13 de octubre de 2023.

Atentamente:



Asesor

Ing. Dr. Miguel Ángel Valles Coral

Declaratoria de autenticidad

Gabriela Abigail Sangama Vela, con DNI N° 70170157, egresada de la Escuela Profesional de Ingeniería de Sistemas e Informática Facultad de Ingeniería de Sistemas e Informática de la Universidad Nacional de San Martín, autores de la tesis titulada: Reconocimiento de patrones visuales para detección de enfermedades en plantaciones de arroz del sector Mishqiyacu – Morales.

Declaro bajo juramento que:

1. La tesis presentada es de mi autoría.
2. La redacción fue realizada respetando las citas y referencia de las fuentes bibliográficas consultadas, siguiendo las normas APA actuales.
3. Toda información que contiene la tesis no ha sido plagiada.
4. Los datos presentados en los resultados son reales, no han sido alterados ni copiados, por tanto, la información de esta investigación debe considerarse como aporte a la realidad investigada.

Por lo antes mencionado, asumo bajo responsabilidad las consecuencias que deriven de mi accionar, sometiéndome a las leyes de nuestro país y normas vigentes de la Universidad Nacional de San Martín.

Tarapoto, 13 de octubre de 2023.



Gabriela Abigail Sangama Vela

DNI: 70170157

Autor

Ficha de identificación

<p>Título del proyecto Reconocimiento de patrones visuales para detección de enfermedades en plantaciones de arroz del sector Mishquiyacu - Morales</p>	<p>Área de investigación: Ciencias naturales Línea de investigación: Ciencias de la computación Sublínea de investigación: Inteligencia artificial y recuperación de la información Grupo de investigación: GIIA – Resolución N° 439–2022–UNSM/FISI/CFT Tipo de investigación: Básica <input type="checkbox"/>, Aplicada <input checked="" type="checkbox"/>, Desarrollo experimental <input type="checkbox"/></p>
<p>Autor: Gabriela Abigail Sangama Vela</p>	<p>Facultad de Ingeniería de Sistemas e Informática Escuela Profesional de Ingeniería de Sistemas e Informática https://orcid.org/0000-0003-3948-0470</p>
<p>Asesor: Ing. Dr. Miguel Ángel Valles Coral</p>	<p>Dependencia local de soporte: Facultad de Ingeniería de Sistemas e Informática Escuela Profesional de Ingeniería de Sistemas e Informática Unidad o Laboratorio Ingeniería de Sistemas e Informática https://orcid.org/0000-0002-8806-2892</p>

Dedicatoria

Dedico este proyecto de investigación a mis padres Samuel Sangama Salas y Luanith María Vela Flores, a mis hermanos, sobrinos, amigos y a cada persona que me ha brindado apoyo sincero a lo largo de mi carrera profesional.

Agradecimientos

En primer lugar, dar gracias a Dios por permitirme la vida y la salud hasta esta etapa de mi vida. De igual manera agradezco a mi asesor Dr. Miguel Ángel Valles Coral por la dedicación y el conocimiento compartido para lograr los objetivos deseados. Asimismo, a los docentes de la Facultad de Ingeniería de Sistemas e Informática por haber sido parte de mi formación profesional durante estos últimos cinco años.

Índice general

Ficha de identificación	6
Dedicatoria	7
Agradecimientos	8
Índice general	9
Índice de tablas	11
Índice de figuras	12
RESUMEN	13
ABSTRACT	14
CAPÍTULO I INTRODUCCIÓN A LA INVESTIGACIÓN	15
CAPÍTULO II MARCO TEÓRICO	18
2.1. Antecedentes de la investigación.....	18
2.2. Fundamentos teóricos	20
CAPÍTULO III MATERIALES Y MÉTODOS	30
3.1. Ámbito y condiciones de la investigación.....	30
3.1.1 Contexto de la investigación.....	30
3.1.2 Periodo de ejecución.....	30
3.1.3 Autorizaciones y permisos	30
3.1.4 Control ambiental y protocolos de bioseguridad.....	30
3.1.5 Aplicación de principios éticos internacionales.....	30
3.2. Sistema de variables	30
3.3 Procedimientos de la investigación.....	31
3.3.1 Objetivo específico 1: Identificar las características y patrones visuales de las enfermedades más frecuentes en las plantaciones de arroz del sector Mishqiyacu. 32	
3.3.2 Objetivo específico 2: Construir e implementar un modelo de reconocimiento de patrones visuales para la detección de las enfermedades más frecuentes en plantaciones de arroz del sector Mishqiyacu.	34
3.3.3 Objetivo específico 3: Comparar el rendimiento del modelo de reconocimiento de patrones visuales con dos modelos de arquitecturas comúnmente usadas para la	

detección de las enfermedades más frecuentes en plantaciones de arroz del sector Mishquiyacu.....	35
CAPÍTULO IV RESULTADOS Y DISCUSIÓN	36
4.1 Resultado específico 1.....	36
4.2 Resultado específico 2.....	38
4.3 Resultado específico 3.....	42
CONCLUSIONES	47
RECOMENDACIONES.....	48
REFERENCIAS BIBLIOGRÁFICAS.....	49

Índice de tablas

Tabla 1. Descripción de variable por objetivo general.....	31
Tabla 2. Métricas según grupo de evaluación.....	45
Tabla 3. Resultados de la prueba ANOVA.....	45
Tabla 4. Estadísticos descriptivos.....	46

Índice de figuras

Figura 1. Una arquitectura típica de red neuronal de convolución (Too et al., 2019).....	23
Figura 2. Representación esquemática de Inception V3 (Google Cloud)	23
Figura 3. Diagrama de bloques para la identificación de enfermedades utilizando el modelo de aprendizaje por transferencia.	24
Figura 4. Estructura general de tipos de enfermedades de las plantas.	27
Figura 5. Imágenes de reconocimiento de las manchas marrones.....	28
Figura 6. Imágenes de reconocimiento del tizón del arroz.	28
Figura 7. Imágenes que conforman el dataset	32
Figura 8. Visita realizada al sector Mishquiyacu.....	33
Figura 9. Imágenes del conjunto de datos histórico	33
Figura 11. Enfoque tradicional vs aprendizaje por transferencia	34
Figura 12. Imágenes pre procesadas.....	38
Figura 13. Métricas del entrenamiento	41
Figura 14. Matriz de confusión resultante	43

RESUMEN

Reconocimiento de patrones visuales para detección de enfermedades en plantaciones de arroz del sector Mishquiyacu - Morales

La detección temprana de enfermedades en las plantaciones de arroz es crucial para garantizar la salud de los cultivos y la productividad agrícola. En el presente trabajo, se propone un enfoque basado en un modelo de red convolucional y reconocimiento de patrones visuales para la detección automática de las enfermedades más frecuentes en las plantaciones de arroz. Se utilizó un conjunto de datos compuesto por 1874 imágenes de hojas de arroz, clasificadas en tres categorías: tizón del arroz (blast), manchón marrón (brownspot) y saludable (healthy). Para el entrenamiento y evaluación del modelo se usó un conjunto de datos de acceso abierto en combinación con imágenes tomadas por un smartphone del sector Mishquiyacu, ubicado en el distrito de Morales. Se dividió el conjunto de datos en un 70% para entrenamiento, un 20% para validación y un 10% para prueba. Se empleó la arquitectura de red convolucional InceptionV3, conocida por su capacidad para extraer características relevantes de las imágenes, aplicando transferencia de aprendizaje para conectar dicho modelo preentrenado a una capa de clasificación de 3 categorías. Además, se aplicaron técnicas de preprocesamiento y aumento de datos para mejorar la generalización del modelo. Los resultados obtenidos mostraron una alta precisión en la detección de enfermedades en las plantaciones de arroz. El modelo alcanzó una exactitud del 96.79% en el conjunto de prueba, superior a los modelos de Alexnet y Resnet50. Este resultado fue posteriormente evaluado mediante métricas de evaluación: precisión, exactitud, F1, recall, Kappa de Cohen, y área bajo la curva ROC, demostrando así la eficacia del modelo para clasificar los datos provistos.

Palabras clave: aprendizaje profundo, plantaciones de arroz, red convolucional, reconocimiento de patrones, InceptionV3

ABSTRACT

Visual pattern recognition for disease detection in rice plantations in the Mishquiyacu-Morales sector

Early detection of diseases in rice plantations is crucial to ensure crop health and agricultural productivity. In this study, an approach based on a convolutional neural network and visual pattern recognition is proposed for the automatic detection of the most common diseases in rice plantations. A dataset consisting of 1874 rice leaf images was used, classified into three categories: rice blast, brown spot, and healthy. For model training and evaluation, an open-access dataset was used in combination with images taken by a smartphone in the Mishquiyacu sector, located in the Morales district. The dataset was divided into 70% for training, 20% for validation, and 10% for testing. The InceptionV3 convolutional neural network architecture was employed, known for its ability to extract relevant features from images, applying transfer learning to connect the pretrained model to a classification layer with 3 categories. Additionally, data preprocessing and augmentation techniques were applied to improve the model's generalization. The results obtained showed high accuracy in disease detection in rice plantations. The model achieved an accuracy of 96.79% on the test set, surpassing the AlexNet and ResNet50 models. This result was further evaluated using evaluation metrics such as precision, accuracy, F1 score, recall, Cohen's kappa, and area under the ROC curve, demonstrating the effectiveness of the model in classifying the provided data.

Keywords: deep learning, rice plantations, convolutional neural network, pattern recognition, InceptionV3



CAPÍTULO I

INTRODUCCIÓN A LA INVESTIGACIÓN

La producción de arroz (*Oryza sativa L.*) es actualmente una actividad importante en el mundo (Ashraf & Khan, 2020), ya que más de la mitad de la población mundial incorpora este alimento básico a su estilo de vida (N et al., 2021). En términos de desarrollo y calidad, las plantaciones de arroz se ven afectadas por la ocurrencia de enfermedades en su ciclo de crecimiento; causadas por bacterias, hongos, virus, o en ocasiones, pueden deberse a factores ambientales (Tiwari et al., 2021). En tal sentido, para Wang et al. (2021) es necesario identificarlas de forma rápida y precisa; sin embargo, su diagnóstico requiere especial atención, habilidades y equipos que en muchos casos los agricultores no poseen.

En el Perú, el 74,8% de la producción nacional de arroz se concentra principalmente en 5 departamentos: Amazonas, San Martín, La Libertad, Piura y Lambayeque; ocupando el puesto 21 dentro del ranking de productores (MIDAGRI, 2021a). Se estima que hasta el año 2020, aproximadamente 150 mil productores se dedican al cultivo de arroz, quienes ocupan una superficie de alrededor de 425 mil hectáreas de superficie agrícola (MIDAGRI, 2021b). Por otra parte; según esta misma fuente, a lo largo de los años la producción nacional ha requerido arroz de alta calidad para abastecer a la población. Lo que ha generado el uso inadecuado de fertilizantes, dando como resultado una mayor susceptibilidad de las plantaciones a las enfermedades.

En la región San Martín, la producción de arroz ha crecido positivamente gracias al alto consumo y a las condiciones climáticas favorables de la zona. Según INEI (2021) en diciembre, la producción de arroz alcanzó las 82.172 toneladas; un 16,9% más que las 70.289 toneladas del mismo mes de 2020. Sin embargo, la incidencia de enfermedades sigue amenazando gravemente la seguridad de la producción local de cultivos en algunas áreas subdesarrolladas (Xiong et al., 2020); lo que indica un problema común en la agricultura del arroz ya que los agricultores no siempre están al tanto de estos factores peligrosos o cuando ocurren (Elmitwally et al., 2022).

De lo descrito anteriormente, se define como problema central de esta investigación la deficiencia en la identificación de enfermedades en plantaciones de arroz mediante procesos de inspección visual, para este proyecto se ha considerado las dos enfermedades más comunes que afectan a las plantaciones de arroz en el sector Mishquiyacu; el tizón del arroz (*Magnaporthe grisea*) y la mancha marrón (*Cochliobolus Miyabeanus*). Estos problemas dan como resultado una disminución significativa en el

rendimiento y la calidad del arroz. Por lo tanto, se dificulta el logro del objetivo de hacer frente a la creciente demanda de arroz (Agus et al., 2018).

Una de las causas de este problema es la dificultad para comprender los síntomas visibles de las especies patógenas que afectan al arroz (Picon et al., 2019), ya que los agricultores poseen un conocimiento limitado adquirido a través de la experiencia (N et al., 2021). Es así que incluso los patólogos más experimentados no logran diagnosticar ciertas enfermedades (Atila et al., 2021). O en el peor de los casos no están disponibles, como en áreas remotas (Sharma et al., 2022).

A su vez, los síntomas de las enfermedades varían de planta a planta, y cada una tiene características distintas. En algunos casos el cultivo sano y el cultivo afectado parecen idénticos en etapas iniciales (Sraavan et al., 2021). En el caso de las manchas marrones, los cambios morfológicos provocados por esta enfermedad tienen formas ovaladas mientras que las manchas blásticas presentan formas elípticas que pueden evidenciarse principalmente en sus hojas. Desde este enfoque, para (Chen, Yin, et al., 2020; Sujatha et al., 2021) surge la importancia de reconocer rasgos basados en manchas, formas y colores.

De manera similar ocurre en un estudio realizado por Almasoud et al. (2022) donde mencionan que algunas enfermedades inducen a exhibir colores similares, pero formas distintas; mientras que algunas tienen colores distintos, pero formas similares. Además de las múltiples enfermedades que afectan a un solo espécimen de planta (Barbedo, 2019). A veces, todas estas diferencias generan confusión en los agricultores debido a la estructura compleja y a la similitud entre las enfermedades (Karar et al., 2021).

Lo que genera un diagnóstico incorrecto de la enfermedad, ya que el método de identificación tradicional se basa principalmente en observaciones visuales de la apariencia de las hojas del arroz, realizado por biólogos o agricultores experimentados (Chen, Yin, et al., 2020). De igual manera para KC et al. (2019) el examen visual se considera un método eficaz para la detección temprana de enfermedades; sin embargo, es propenso a errores humanos puesto que la evaluación visual es subjetiva y depende de la perspectiva visual desde el suelo donde se apoyan el observador y el objeto a examinar. Por consiguiente, dicho método puede ser engorroso y confuso para la visión humana (Arinichev et al., 2021).

Otra de las consecuencias es el alto nivel de pérdida de recursos tras un diagnóstico inexacto de la enfermedad, lo que provoca pérdidas económicas significativas (Elmitwally et al., 2022) tanto para la economía nacional como para los hogares de pequeños productores que dependen en gran medida del desarrollo saludable de sus cultivos y de las ganancias netas que genera dicha actividad (Sowmyalakshmi et al., 2021). En

consecuencia, la agricultura debe evaluarse desde una perspectiva socioeconómica debido a que es un método importante para el desarrollo sostenible del sector arrocerero (Wei et al., 2022).

La principal limitación del problema ante una posible solución es la capacidad financiera del agricultor, debido a que la mayoría de familias productoras no disponen una parte de su presupuesto en adquisición de recursos tecnológicos. Y valiéndonos de la investigación realizada por Sujatha et al. (2021) se puede afirmar que el uso de tecnología moderna proporciona técnicas avanzadas para lograr la optimización de la producción agrícola y el desarrollo económico de la agricultura.

Dado que la apariencia externa de las hojas puede ser objeto de procesamiento de imágenes, en esta investigación se implementó el uso de un smartphone para la recolección de imágenes de campo del sector Mishqiyacu. De igual manera se empleó la CNN preentrenada Inception v3 potenciada con aprendizaje por transferencia con el fin de ayudar en la detección de las enfermedades descritas anteriormente; la mancha marrón y el tizón del arroz (Ma et al., 2018). En definitiva, se logró complementar los métodos tradicionales de manejo agrícola con tecnología para mejorar la eficiencia y la perspectiva visual al momento de identificar enfermedades del arroz (Tian et al., 2020), así como reducir el esfuerzo humano en tareas laboriosas y repetitivas (Anuar et al., 2022). Dentro de este marco, se planteó la hipótesis: El reconocimiento de patrones visuales ayuda a detectar las enfermedades más frecuentes que afectan las plantaciones de arroz del sector Misqiyacu.

Asimismo, se propuso como objetivo general detectar las enfermedades más frecuentes que afectan a las plantaciones de arroz a través de reconocimiento de patrones visuales. A continuación, los objetivos específicos fueron: 1) identificar las características y patrones visuales de las enfermedades más frecuentes en las plantaciones de arroz del sector Mishqiyacu; 2) construir e implementar un modelo de reconocimiento de patrones visuales para la detección de las enfermedades más frecuentes en plantaciones de arroz del sector Mishqiyacu; y 3) comparar el rendimiento del modelo de reconocimiento de patrones visuales con dos modelos de arquitecturas comúnmente usadas para la detección de las enfermedades más frecuentes en plantaciones de arroz del sector Mishqiyacu.

CAPÍTULO II

MARCO TEÓRICO

Debido a la relevancia del tema, se han realizado varias revisiones del estado del arte con el objetivo de analizar los trabajos publicados en los últimos años disponibles en la literatura. A continuación, se detallan 6 antecedentes de investigación con mayor semejanza al tema de estudio:

2.1. Antecedentes de la investigación

Internacional

Simhadri & Kondaveeti (2023) presentan en su artículo “Reconocimiento automático de enfermedades de la hoja del arroz mediante aprendizaje por transferencia”, una metodología para detectar y clasificar automáticamente las enfermedades de las hojas de arroz usando el aprendizaje por transferencia sobre modelos pre-entrenados de redes neuronales convolucionales (CNN). El estudio utiliza un conjunto de datos de 10.080 imágenes de diez enfermedades de las hojas de arroz y una clase saludable, y evalúa el rendimiento de 15 modelos pre-entrenados de CNN, como ResNet, Inception, VGG, entre otros. El objetivo es identificar el modelo más efectivo y eficiente para la clasificación de las enfermedades de las hojas de arroz. Los resultados muestran que el modelo InceptionV3 obtiene la mayor precisión media con un 99,64%, seguido por el modelo Xception con un 99,61%. El estudio demuestra la superioridad del aprendizaje por transferencia sobre los modelos pre-entrenados de CNN para el diagnóstico automático de las enfermedades de las hojas de arroz, lo que puede ayudar a mejorar el rendimiento y la calidad del cultivo.

Otra investigación titulada “Clasificación de la enfermedad del arroz Hispa mediante redes neuronales convolucionales” realizada por Sharma et al., (2021) en la que propusieron una red neuronal convolucional (CNN) basada en aprendizaje profundo para detectar la enfermedad del hispa del arroz a partir de imágenes de plantas de arroz. El objetivo del estudio fue desarrollar un sistema automatizado y eficiente para la detección de esta enfermedad que afecta la producción y el consumo de arroz en el mundo. Los resultados mostraron que el modelo CNN propuesto alcanzó una precisión del 94% en la clasificación de imágenes de plantas de arroz sanas y enfermas, utilizando un conjunto de datos en tiempo real recolectado de los campos de arroz de Punjab.

En un estudio de Rahman et al. (2020) titulado “Identificación y reconocimiento de enfermedades y plagas del arroz mediante redes neuronales convolucionales” desarrollaron métodos basados en aprendizaje profundo para detectar plagas y enfermedades a partir de imágenes de plantas de arroz. Adoptaron y mejoraron arquitecturas de última generación a gran escala como VGG16 e InceptionV3 para detectar e identificar plagas y enfermedades del arroz. Los resultados experimentales demostraron la eficacia de estos modelos en conjuntos de datos reales. Dado que las arquitecturas a gran escala no son adecuadas para dispositivos móviles, presentaron una arquitectura CNN pequeña de 2 etapas y compararon con las arquitecturas CNN eficientes en memoria de última generación, como MobileNet, NasNet Mobile y SqueezeNet. Los resultados experimentales mostraron que la arquitectura propuesta puede lograr una precisión prevista del 93,3 % al mismo tiempo que reduce significativamente el tamaño del modelo.

Latif et al. (2022) en su artículo de investigación titulada “Utilización de Deep Learning en Agricultura: Detección de enfermedades de plantas de arroz usando un modelo CNN mejorado” presenta un método basado en aprendizaje profundo para la detección y clasificación de enfermedades de las plantas de arroz usando un modelo mejorado de VGG19. El método propuesto utiliza una técnica de aprendizaje por transferencia para extraer características de las imágenes de las hojas de arroz y clasificarlas en seis clases distintas: sana, mancha parda estrecha, escaldadura foliar, explosión foliar, mancha parda y quemazón bacteriana. Los resultados del método propuesto se evaluaron usando una métrica de precisión media y se compararon con otros métodos similares de la literatura existente. El método propuesto logró una precisión media del 96,08% usando un conjunto de datos aumentado no normalizado. El artículo concluye que el método propuesto ofrece resultados significativamente mejores que los métodos existentes y que puede ser una herramienta útil para los agricultores y los investigadores en el campo de la agricultura.

Nacional

En un estudio titulado “Visión artificial en reconocimiento de patrones para clasificación de frutas en agronegocios” realizado por Sucari-León et al. (2020) presentaron una investigación con la finalidad de especificar la efectividad de la aplicación de visión artificial para la identificación de modelos o patrones de frutas. En este trabajo utilizaron seis tipos de frutas con una base de datos de cincuenta registros, donde consideraron cuatro propiedades para cada una de las frutas además una muestra de veinte frutas. Asimismo, aplicaron el método de identificación automático de patrones mediante un clasificador bayesiano implementado en Octave. Este sistema obtuvo una efectividad de 93.33%. De

esta manera, los autores llegaron a concluir que aplicar visión artificial en procesos donde se debe identificar patrones resulta muy efectivo.

Local

En el ámbito local Santa María-Pinedo et al. (2021) en su investigación “Reconocimiento de patrones de imágenes a través de un sistema de visión artificial en MATLAB” realizaron un estudio para cuantificar el grado de adecuación que existe entre un sistema de visión artificial y la identificación de patrones. La población y muestra que utilizaron es este estudio constaba de ocho modelos de imagen, el método que emplearon fue la hoja de verificación, que aplicaron a dos grupos (experimental y control). La adquisición y procesamiento de las imágenes fue realizado en la facultad de Ingeniería de Sistemas e Informática en un entorno controlado asistido por una cámara de video. Los autores concluyeron que el tiempo de procesamiento fue de 10.75 segundos para el grupo experimental mientras que para el grupo control fue 67.75 segundos; además obtuvieron un porcentaje del 72% con respecto al grado de adecuación entre el sistema de visión artificial y la identificación de patrones.

2.2. Fundamentos teóricos

Visión Artificial

Para Patrício & Rieder (2018) la visión artificial es un área de la inteligencia artificial, que aborda técnicas y/o métodos mediante los cuales se pueden construir sistemas para emplearse de manera razonable en aplicaciones prácticas. Esta área de la informática incluye software, hardware y técnicas de imagen necesarias para esos métodos. Principalmente, los sistemas de visión artificial tienen las siguientes etapas:

- Adquisición de imágenes. Comienza con el uso de dispositivos como cámaras para transmitir señales electrónicas desde sensores a representaciones digitales. Existen principalmente dos tipos de cámaras disponibles que difieren en la forma en que escanean: área o línea. La calidad de las imágenes obtenidas por un sistema de visión artificial se ven directamente afectadas por la iluminación utilizada en la fase de adquisición. De esta forma, cualquier esfuerzo invertido en utilizar una iluminación adecuada aumentará el rendimiento y la fiabilidad del sistema y reducirá la complejidad del software utilizado en la fase de procesamiento.
- Procesamiento de imágenes. Esto incluye tareas realizadas para procesar imágenes digitales, como mejorar la calidad de la imagen, corregir problemas de iluminación o reducir el ruido. Además, el análisis de imágenes se refiere al proceso de separar una región de interés de otras regiones para extraer información.

En solo unas pocas décadas, la visión artificial se ha afianzado en todo, desde el reconocimiento de patrones, el aprendizaje automático, los gráficos por computadora, la reconstrucción 3D, la realidad virtual y aumentada (Kakani et al., 2020). Por esta razón, para Zareiforoush et al. (2016), el diseño y desarrollo de sistemas automáticos basados en visión artificial e inteligencia artificial para clasificar enfermedades en hojas de arroz puede dar como resultado una mayor calidad del producto final.

Redes neuronales

Son modelos matemáticos que simulan los principios generales del funcionamiento del cerebro con las neuronas y sinapsis que los conectan. Su característica principal es su capacidad para capacitarse a través del aprendizaje supervisado (Ferentinos, 2018).

Para Patrício & Rieder (2018) una red neuronal consta de nodos conectados por enlaces dirigidos con pesos asociados que determinan la fuerza y la señal de la conexión. Los pesos son la principal forma de almacenamiento continuo en las redes neuronales, y los algoritmos de aprendizaje a menudo pueden actualizar estos pesos. Los nodos de entrada y salida son los que tienen conexiones con el mundo exterior. Cada nodo es independiente de los demás y, por lo tanto, puede realizar sus cálculos solo a partir de respectivos pesos y valores. Cada unidad recibe los valores de los enlaces entrantes, calcula el nivel de activación y, cuando se activa, envía una señal a todos sus enlaces salientes.

Existen 4 arquitecturas principales de redes neuronales, donde cada una de ellas es única y puede aplicarse para resolver problemas en diferentes campos o áreas (Patayon & Crisostomo, 2021). Las cuales comprenden: Redes sin supervisión pre-formados, redes neuronales convolucionales, red neuronal recursiva, red neuronal recurrente.

Redes neuronales convolucionales

Una red neuronal convolucional (CNN) es una evolución de las redes neuronales artificiales tradicionales, y se enfoca principalmente en aplicaciones con patrones recurrentes en diferentes regiones del espacio de modelado, específicamente el reconocimiento de imágenes. Estas redes toman imágenes como entrada, descubren una serie de características dentro de cada imagen y, en función de estas características, pueden distinguir una imagen de otra; es decir, pueden hacer una clasificación. Su principal ventaja es que, con la metodología utilizada para la estratificación, reducen significativamente el número de elementos estructurales necesarios (número de neuronas artificiales). Para las aplicaciones de reconocimiento de imágenes, se han desarrollado varias arquitecturas de CNN, que se han aplicado con éxito a tareas complejas (Ferentinos, 2018). Como, por ejemplo: AlexNet, GoogleNet, VGG, Inception V3, LeNet, EfficientNet, entre otras.

La arquitectura general de las CNN consta de tres tipos de capas: capas convolucionales, capas de agrupación y capas totalmente conectadas. Cuando estas capas se apilan, surge una arquitectura CNN (O'Shea & Nash, 2015). A continuación, Chen, Chen, et al. (2020) describen cada uno de los tipos de capas:

- Capas convolucionales. es el componente crucial de la CNN, que extrae las características específicas de la imagen mediante los distintos tamaños del núcleo de convolución. Se puede extraer un conjunto de mapas de características de las imágenes de entrada tras aplicar varias veces las capas convolucionales. Supongamos que H_i representa el mapa de características de la capa i -ésima de la CNN, entonces H_i se puede generar de la siguiente manera:

$$H_i = \varphi(H_{i-1}W_i + b_i)$$

Donde H_i denota el mapa de características de la capa actual de la red, H_{i-1} es la característica de convolución de la capa anterior (H_0 es la imagen original), W_i es el peso de la capa i -ésima, b_i es el vector de desplazamiento de la capa i -ésima, y $\varphi(\cdot)$ representa la función de unidad lineal rectificadora (ReLU).

- Capas de agrupación. Su función es reducir la dimensión espacial, lo que puede reducir la complejidad computacional y controlar eficazmente el riesgo de sobreajuste. En la l -ésima capa de agrupación, la característica de salida del j -ésimo campo receptivo local puede calcularse mediante la ecuación:

$$x_j^l = \text{down}(x_j^{l-1}, s)$$

Donde $\text{down}(\cdot)$ representa la función de muestreo descendente, x_j^{l-1} es el vector de características de la capa anterior y s es el tamaño de la agrupación.

- Capas totalmente conectadas. Después de las capas convolucionales y de agrupamiento, hay una o varias capas totalmente conectadas, y el objetivo de estas capas es utilizar las características extraídas para la clasificación de imágenes. La función Softmax suele emplearse para realizar la predicción de clase con las características extraídas de las capas anteriores. Matemáticamente, la función Softmax se define como:

$$\text{softmax}(z)_j = e^{z_j} / \sum_{k=1}^K e^{z_k} (\text{for } j = 1, \dots, K)$$

Donde K representa la dimensión del vector z .

Agregando a lo anterior, la Figura 1 muestra una arquitectura típica de CNN.

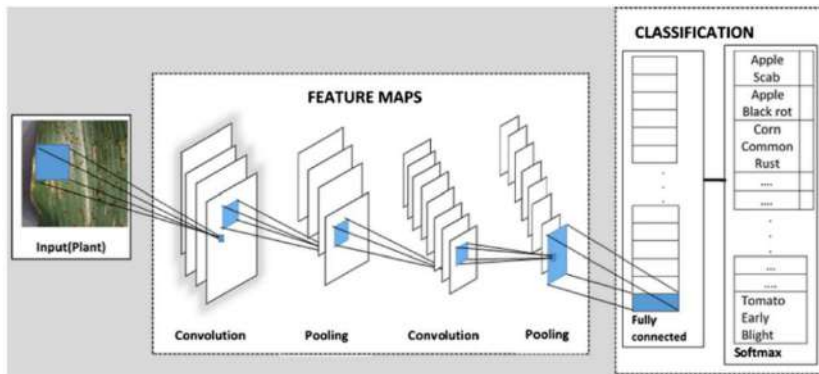


Figura 1

Una arquitectura típica de red neuronal de convolución (Too et al., 2019)

Por consiguiente, este método de transformación simple permite que las CNN transformen la entrada original capa por capa utilizando técnicas de convolución y de reducción de muestreo para generar puntajes de clase con fines de clasificación y regresión.

Inception V3

Desarrollado por Google, Inception V3 es el tercer lanzamiento de la serie Deep Learning Evolutionary Architectures. Después de desarrollar la arquitectura Inception V1, la normalización por lotes se realizó en Inception V2. Posteriormente, la idea de factorización se introdujo en Inception V3. El objetivo principal de la factorización es reducir el número de conexiones y parámetros sin reducir la eficiencia de la red. El modelo en sí consta de bloques de construcción simétricos y asimétricos que contienen convoluciones, agrupación promedio, agrupación máxima, concatenación, abandonos y capas completamente conectadas. La arquitectura Inception V3 con función Softmax en la capa final consta de un total de 42 capas, la capa de entrada captura imágenes de 299x299x3 píxeles (Atila et al., 2021).

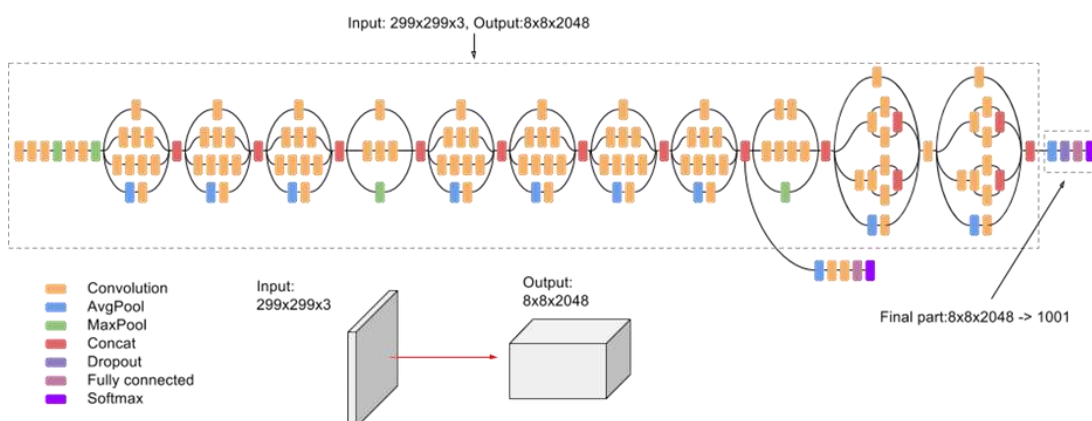


Figura 2

Representación esquemática de Inception V3 (Google Cloud)

Es preciso señalar que, según Albatayneh et al. (2020) el modelo Inception v3 es considerado una de las arquitecturas de redes neuronales convolucionales para la clasificación de imágenes, utiliza el módulo de extracción de características de imagen que se entrenó en ImageNet (una base de datos accesible para imágenes de alta resolución diseñada para desarrolladores e investigadores en el campo del procesamiento de imágenes). En general, el modelo Inception-v3 consta de dos partes principales: (1) la red neuronal convolucional para extraer características de la imagen; y (2) la clasificación de imágenes con softmax y capas totalmente conectadas. La capa softmax se usa como la capa final de un clasificador basado en una red neuronal para proporcionar probabilidades de clase normalizadas para las salidas.

Aprendizaje por transferencia

El aprendizaje por transferencia o transfer learning es un método que permite reutilizar redes neuronales convolucionales preentrenadas para nuevos problemas. Por lo tanto, puede reducir el tiempo de entrenamiento del modelo y mejorar el rendimiento del modelo propuesto en comparación con los modelos desarrollados desde cero. El aprendizaje por transferencia se puede utilizar para crear modelos robustos de aprendizaje automático con relativamente pocos datos de entrenamiento porque el modelo ya está preentrenado (N et al., 2021; Patayon & Crisostomo, 2021).

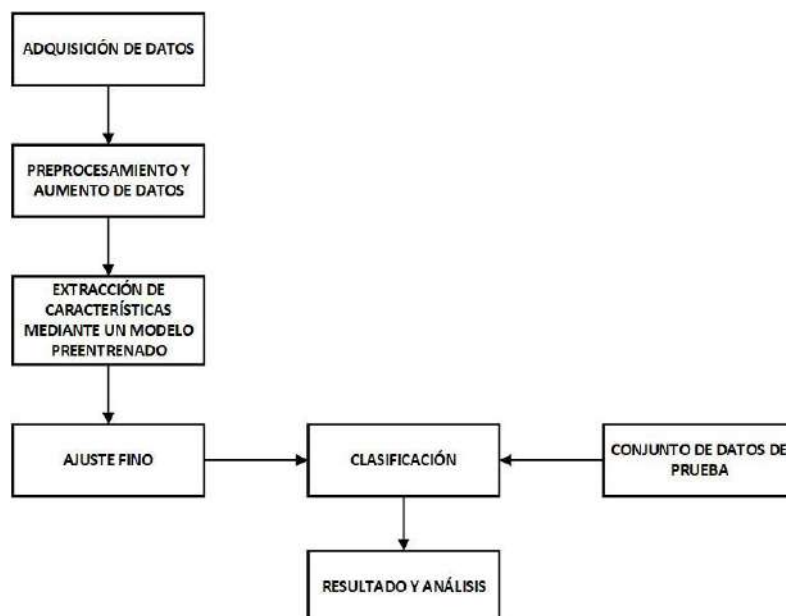


Figura 3

Diagrama de bloques para la identificación de enfermedades utilizando el modelo de aprendizaje por transferencia (N et al., 2021)

No obstante, Kaya et al. (2019) menciona las dos estrategias comunes de aprendizaje por transferencia son la extracción profunda de características y el ajuste fino. En la extracción

profunda de características, los datos de entrada se proporcionan a la red previamente entrenada y los valores de activación de varias capas se almacenan y utilizan como características. En el ajuste fino, la red neuronal se entrena para una tarea similar, en la que el etiquetado es relativamente más fácil. Si bien las primeras capas de la red preentrenada se pueden reparar, se pueden realizar ajustes en las capas finales del modelo para conocer las propiedades del nuevo conjunto de datos. El modelo previamente entrenado se vuelve a entrenar con el nuevo conjunto de datos pequeños y los valores de peso del modelo se actualizan para una nueva tarea. Sin lugar a dudas, aprender a transmitir suele ser más rápido que entrenar una nueva red neuronal desde cero.

TensorFlow

Es una de las bibliotecas de aprendizaje automático más populares, creada por el equipo de Google Brain para construir modelos de aprendizaje profundo que utilizan redes neuronales multicapa. Se puede usar tanto para aplicaciones de producción como de investigación, con un enfoque particular en el entrenamiento de redes neuronales profundas. Tiene funciones bien diseñadas que permiten a los desarrolladores e investigadores crear clasificadores que pueden resolver muchos problemas reservados para los humanos utilizando técnicas más eficientes y económicas (Albatayneh et al., 2020). Esta arquitectura consta de 3 fases:

- Procesamiento previo de los datos.
- Construcción del modelo.
- Entrenamiento y estimación del modelo.

La entrada que se introduce en TensorFlow es una matriz multidimensional, y estas matrices se denominan tensores. El flujo de una matriz multidimensional dio el nombre TensorFlow. Estos tensores ingresan al sistema como entrada en un extremo y pasan por varias operaciones y finalmente producen salida (Joseph et al., 2021).

Usar TensorFlow tiene muchas ventajas sobre otras herramientas de inteligencia artificial, como la visualización de datos y la flexibilidad. Estas dos se consideran las dos ventajas principales que hacen de esta herramienta una de las líderes en su campo. El uso de gráficos y tablas para visualizar lo que sucede en el código ayuda a los investigadores y desarrolladores a comprenderlo por completo (Albatayneh et al., 2020).

Keras

Según Joseph et al. (2021) keras se desarrolló sobre TensorFlow. Por esta razón, antes de explorar keras, es necesario analizar TensorFlow en detalle. Tiene cuatro principios de funcionamiento: Facilidad de uso, modularidad, fácil extensibilidad, trabajar con Python.

Existen once tipos diferentes de capas de Keras disponibles para configurar la red neuronal de aprendizaje profundo y son las siguientes:

1. Capas centrales
2. Capas convolucionales
3. Capas de agrupación
4. Capas conectadas localmente
5. Capas recurrentes
6. Incrustar capas
7. Combinar capas
8. Capas de activación avanzadas
9. Capas de normalización
10. Capas de ruido
11. Envolturas de capas

Matplotlib

Es un paquete de gráficos para la visualización de datos en Python. Matplotlib se ha convertido en una parte central de la pila de ciencia de datos de Python y se integra bien con NumPy y Pandas. El módulo pyplot está estrechamente relacionado con los comandos de trazado de MATLAB. Por lo tanto, los usuarios de MATLAB pueden cambiar fácilmente al uso de Python para el trazado. Matplotlib también se usa para mostrar imágenes. Este procedimiento se utiliza cuando se visualiza un conjunto de datos de píxeles de imagen (Bisong, 2019).

NumPy

Es una biblioteca de Python optimizada para computación numérica. NumPy es la abreviatura de Numerical Python. La principal ventaja de NumPy es su capacidad para crear y manipular matrices de n dimensiones. Esto es especialmente importante para construir modelos de aprendizaje automático y aprendizaje profundo (Bisong, 2019).

Jupyter Notebook

Es una herramienta de navegador de código abierto que se puede usar como un cuaderno de laboratorio virtual para admitir flujos de trabajo, código, datos y visualizaciones que describen el proceso de investigación. Es legible por máquina y por humanos, lo que facilita la interoperabilidad y el intercambio científico. Estos cuadernos pueden existir en repositorios en línea y proporcionar enlaces a objetos de investigación, como conjuntos de datos, código, documentos de métodos, flujos de trabajo y publicaciones ubicadas en otros lugares. Los cuadernos de Jupyter son una herramienta para hacer que la ciencia sea más

abierta. Su importancia para la comunidad JCDL es su interacción con varios componentes de la infraestructura de la biblioteca digital, como identificadores digitales, mecanismos de persistencia, control de versiones, conjuntos de datos, documentación, software y publicaciones (Randles et al., 2017).

Enfermedades en plantaciones de arroz

Las enfermedades en plantaciones son factores que contribuyen a la disminución de la calidad y cantidad de los cultivos agrícolas. En general, las enfermedades pueden atacar a cualquier planta en cualquier momento (Poornappriya & Gopinath, 2020).

Las enfermedades de las hojas en el cultivo del arroz se pueden clasificar principalmente según el tipo de agente causal principal. A continuación, en la Figura 4 se muestra la estructura general de los tipos de enfermedades en plantaciones.

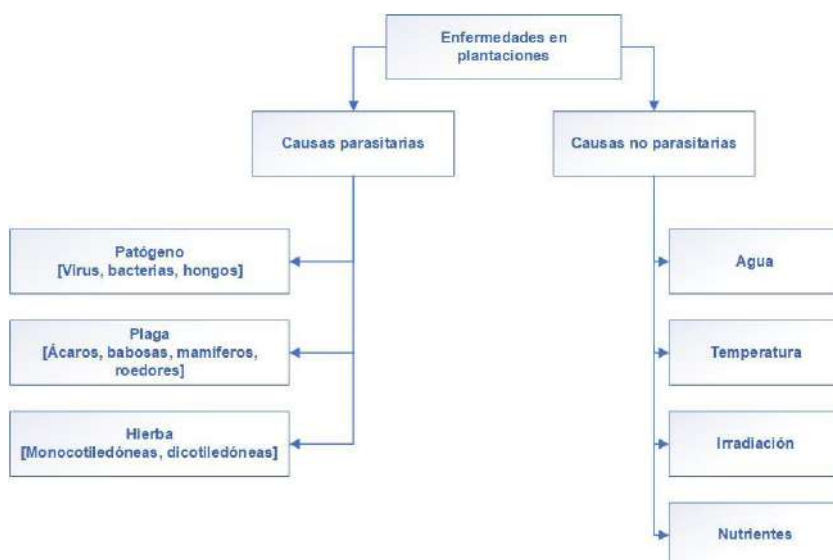


Figura 4

Estructura general de tipos de enfermedades de las plantas (Dhingra et al., 2018)

A continuación, se mencionan las dos enfermedades más comunes del arroz en el sector Mishquiayacu:

- **Manchas marrones:** Caracterizadas por una amplia gama de síntomas. El signo más común de esta infección es una mancha ovalada marrón con un anillo amarillo. A medida que se crecen, se puede formar un centro gris en el centro de la mancha, además se puede visibilizar bordes de color marrón rojizo. Otro síntoma destacado es la decoloración del tallo (Debnath & Saha, 2022).



Figura 5

Imágenes de reconocimiento de las manchas marrones

- Tizón del arroz: Esta enfermedad se caracteriza por presentar manchas de color amarillo a verde claro de forma ovalada. Los márgenes son necróticos y el centro presenta un color grisáceo. A medida que se avanza la enfermedad, las hojas van secándose gradualmente. Este daño es uno de los más devastadores para el arroz (Tian et al., 2021).



Figura 6

Imágenes de reconocimiento del tizón del arroz

Detección de enfermedades en plantaciones de arroz

Una estrategia para lidiar con la pérdida de cultivos debido a la infección por enfermedades es capturar fotografías de las hojas infectadas para obtener más información. Los sistemas de detección de enfermedades del arroz basados en técnicas de redes neuronales convolucionales profundas (CNN) son una buena opción. Las CNN se capacitan para reconocer enfermedades comunes del arroz utilizando conjuntos de datos con fotos reales de plantaciones de arroz dañadas y saludables (Poornappriya & Gopinath, 2020).

La preparación de los datos y el entrenamiento de los modelos son pasos importantes, medir el rendimiento del modelo que se está entrenando es igualmente importante. Lo bien que el modelo generaliza sobre los datos no vistos es lo que define los modelos de aprendizaje automático adaptables frente a los no adaptables (Ghosal & Sarkar, 2020).

De modo que, el uso de diferentes métricas para la evaluación del rendimiento nos brinda la oportunidad de mejorar nuestra percepción general del modelo antes de ponerlo en producción con datos desconocidos. Elmitwally et al. (2022) definen las siguientes métricas de evaluación:

- Precision: La precisión es la proporción de predicciones correctas con un conjunto de datos de prueba. Es la relación entre el número de predicciones correctas y el número total de ejemplos de entrada.

$$Precision = \frac{TP}{TP + FP}$$

- Recall: La métrica de exhaustividad nos muestra la cantidad de verdaderos positivos que el modelo ha clasificado en base al número total de valores positivos.

$$Recall = \frac{TP}{TP + FN}$$

- F1 Score: Es la media armónica de precisión y recall. La puntuación F1 es útil cuando se busca un equilibrio entre precisión y recall.

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

- Accuracy: La exactitud mide el porcentaje de casos que el modelo ha acertado.

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

CAPÍTULO III

MATERIALES Y MÉTODOS

3.1. Ámbito y condiciones de la investigación

3.1.1 Contexto de la investigación

La investigación se llevó a cabo en el sector Mishqiyacu, ubicado en el distrito de Morales, provincia y departamento de San Martín.

3.1.2 Periodo de ejecución

La ejecución del proyecto se llevó a cabo durante los meses de noviembre 2022 a mayo 2023.

3.1.3 Autorizaciones y permisos

Los permisos correspondientes han sido solicitados a propietarios de cultivos de arroz en el sector de Mishqiyacu para facilitar el acceso a los campos y capturar imágenes de las plantaciones que conformarán las muestras de estudio.

3.1.4 Control ambiental y protocolos de bioseguridad

Dado que la captura de las imágenes de las plantaciones de arroz se realizó al aire libre en un espacio abierto, se siguieron protocolos básicos de bioseguridad para garantizar que la salud y el bienestar de los involucrados no corra peligro.

3.1.5 Aplicación de principios éticos internacionales

Los resultados obtenidos de la investigación no perjudican a ninguna persona o grupo de personas en especial. Se espera que la presente investigación sea de utilidad para otros investigadores, de manera que se pueda brindar solución a un problema común en el ámbito de la agricultura.

3.2. Sistema de variables

La operacionalización de la variable se representa en la tabla 1 (ver siguiente página).

Tabla 1.*Descripción de variable por objetivo general*

Objetivo general: Detectar las enfermedades más frecuentes que afectan plantaciones de arroz a través de reconocimiento de patrones visuales				
Variable		Indicadores	Medio de registro	Escala de evaluación
Detección de enfermedades en plantaciones de arroz		Precision	Informe	Numérico
		Recall	Informe	Numérico
		F1 Score	Informe	Numérico
		Accuracy	Informe	Numérico

3.3 Procedimientos de la investigación

a) Tipo y nivel de la investigación

La investigación es de carácter aplicado, ya que se utiliza el conocimiento de las técnicas de procesamiento de imágenes y reconocimiento de patrones para resolver un problema específico; la detección de las enfermedades más frecuentes que afectan al arroz en el sector Mishquiyacu, mediante la construcción de un modelo de reconocimiento de patrones visuales que emplea una CNN existente previamente entrenada.

El alcance de este estudio es de nivel descriptivo y de diseño experimental porque se analizó y observó el comportamiento del modelo, no se manipuló ninguna de las variables, con el fin de conocer la potencialidad del modelo propuesto al detectar las enfermedades más frecuentes que afectan a las plantaciones de arroz.

b) Población y muestra

La población de la presente investigación está conformada por imágenes de un conjunto de datos de acceso abierto, el cual se puede encontrar en el siguiente enlace: <https://archive.ics.uci.edu/ml/datasets/rice+leaf+diseases>, en combinación con imágenes tomadas por un smartphone del sector Mishquiyacu, ubicado en el distrito de Morales; delimitada por 1874 imágenes de plantaciones de arroz sanas y enfermas con sus respectivas etiquetas.

La muestra está conformada por la cantidad de imágenes de la población, ya que es necesario una mayor cantidad de imágenes para lograr la precisión del modelo. El conjunto de datos formado con imágenes de la muestra se divide en dos partes, donde el 70% de las imágenes corresponde a datos de entrenamiento, 20% a datos de validación y el 10% restante a datos de prueba. Para los criterios de selección se utiliza el muestreo no probabilístico, donde la selección de elementos se basa en las características de la investigación, no de la probabilidad. Del conjunto de datos, las imágenes corresponden a 3 clases: Tizón del arroz, Mancha marrón e imágenes de plantaciones de arroz saludables.



Figura 7

Imágenes que conforman el dataset

c) Diseño analítico, muestral y experimental

Se consideró un diseño no experimental y de nivel descriptivo, debido a que se realizó la ejecución del modelo y a su vez se observó el comportamiento del mismo.

Para llevar a cabo el proyecto, se empleó un conjunto de datos compuesto por 1874 imágenes de plantaciones de arroz, estas imágenes son parte de una combinación de imágenes del dataset e imágenes capturadas a través de un smartphone Huawei P Smart 2019 de plantaciones de arroz del sector Mishqiyacu. La iluminación fue crucial para obtener fotos claras y detalladas, las sesiones de fotografía fueron en función de la luz natural, teniendo en cuenta la posición del sol para que la composición de la imagen sea clara y que se destaquen los elementos importantes de la investigación, en un intervalo de 11:00 am a 1:00 pm, durante 3 semanas. Las visitas fueron realizadas en compañía de un ingeniero agrónomo para el etiquetado y reconocimiento de las enfermedades.

Cabe recalcar que no se construyó un modelo nuevo desde cero, sino que se usó un modelo de CNN previamente entrenado desarrollado por Google llamado Inception v3 potenciado por aprendizaje por transferencia para la detección de las enfermedades más frecuentes en plantaciones de arroz, que posteriormente se afinó y entrenó con el conjunto de datos creado previamente.

Se evaluaron los resultados del modelo empleando métricas de evaluación con el fin de ayudarnos a determinar la efectividad y validez del modelo.

3.3.1 Objetivo específico 1: Identificar las características y patrones visuales de las enfermedades más frecuentes en las plantaciones de arroz del sector Mishqiyacu.

1. Recopilación de información del sector y enfermedades a detectar: Para realizar esta actividad acudimos al sector Misqiyacu y hablamos con uno de los miembros que trabaja y forma dicha asociación arrocera. Aprendimos sobre las enfermedades más comunes del arroz y las características de cada enfermedad.



Figura 8

Visita realizada al sector Mishquiyacu

2. Construcción del conjunto de datos histórico: Los datos son la materia prima que alimenta los algoritmos de inteligencia artificial para que puedan tomar decisiones basadas en el análisis del comportamiento pasado y predecir el comportamiento futuro, es así que para la construcción del conjunto de datos se realizó uniendo imágenes de plantaciones de arroz del sector Mishquiyacu capturadas mediante un smartphone, con imágenes de un conjunto de datos disponible públicamente.



Figura 9

Imágenes del conjunto de datos histórico

3. Identificación de patrones visuales y características de las enfermedades: Para ello, nos reunimos con un experimentado productor de arroz del sector Mishquiyacu para identificar y etiquetar las imágenes capturadas con el smartphone, así como para la verificación del dataset de acceso público utilizado en el presente estudio. Seguidamente, organizamos las imágenes en carpetas según la categoría a la que cada imagen pertenecían: “blast_a”, “brownspot_c” y “healthy_b”. Particionamos cada categoría de imágenes en una proporción de 70%, 20% y 10% para los sets de entrenamiento, validación y prueba respectivamente. Finalmente, las imágenes serán preprocesadas a través de métodos de normalización,

redimensionamiento y técnicas de aumento de datos para evitar el sobreajuste del modelo, así como proporcionar una mayor cantidad de datos para el entrenamiento.

3.3.2 Objetivo específico 2: Construir e implementar un modelo de reconocimiento de patrones visuales para la detección de las enfermedades más frecuentes en plantaciones de arroz del sector Mishqiyacu.

1. Recopilación de teoría sobre aprendizaje por transferencia: Se recopiló información sobre el aprendizaje por transferencia de diferentes artículos para poder emplear modelos que han sido previamente entrenados y utilizar parte del conocimiento que han adquirido en su entrenamiento, con el fin de resolver una tarea específica de forma independiente.

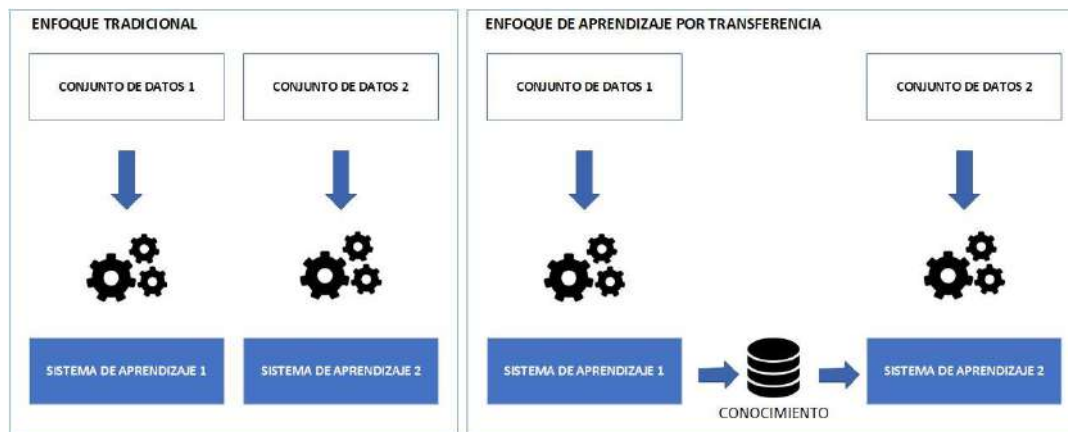


Figura 10

Enfoque tradicional vs aprendizaje por transferencia

2. Recopilación de teoría sobre la arquitectura Inception v3: Se exploraron aspectos de Inception v3 de diversos artículos y se analizaron para la construcción del modelo. Inception v3 utiliza módulos llamados Inception, estos funcionan como múltiples filtros aplicados a un mismo valor de entrada a través de capas convolucionales y de agrupación. Lo que permite aprovechar la extracción de patrones que proporcionan diferentes tamaños en los filtros. Finalmente, el resultado de estos filtros se concatena y se utilizan como valor de salida del módulo. Este modelo tiene la propiedad de aumentar el número de parámetros entrenables y el cálculo requerido, pero la precisión mejora significativamente.

3. Construcción del modelo basado en Inception v3.

4. Inspección y selección de métricas de evaluación: Se seleccionaron las siguientes métricas para la evaluación del modelo:

- **Precision:** Es el porcentaje de predicciones correctas utilizando el conjunto de datos de prueba. Es la relación entre el número de predicciones correctas y el número total de muestras de entrada.
- **Recall:** Se calcula como el cociente entre el número de muestras positivas correctamente clasificadas como positivas y el número total de muestras positivas.

Mide la capacidad del modelo para detectar muestras positivas. Cuanto mayor sea el recall, más muestras positivas se detectaron.

- F1 Score: Es la media armónica de precisión y recall. La puntuación F1 resulta útil cuando se desea encontrar un equilibrio entre precisión y recall.
- Accuracy: Es una métrica que generalmente describe el rendimiento del modelo en todas las categorías. Se calcula a partir de la relación entre el número de predicciones correctas y el número total de predicciones.

3.3.3 Objetivo específico 3: Comparar el rendimiento del modelo de reconocimiento de patrones visuales con dos modelos de arquitecturas comúnmente usadas para la detección de las enfermedades más frecuentes en plantaciones de arroz del sector Mishquiyacu.

1. Construcción de un modelo basado en AlexNet.
2. Construcción de un modelo basado en ResNet50.
3. Aplicación de las métricas de evaluación a cada modelo.
4. Obtención de resultados de las métricas de evaluación con 6 grupos de evaluación.
5. Aplicación de la prueba ANOVA a los indicadores de rendimiento de cada modelo para determinar la existencia de diferencias significativas entre el rendimiento de los distintos modelos.

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

4.1 Resultado específico 1

Con el fin facilitar el procesamiento de las imágenes y la detección de patrones por parte del modelo, es importante iniciar con el preprocesamiento de los datos que serán provistos en las distintas fases del proyecto.

Para ello importamos los recursos y librerías que será utilizados durante la construcción del modelo:

```
# Importación de librerías requeridas
import tensorflow as tf
import numpy as np
import keras
from keras.layers import Input, Dense, Flatten
from keras.models import Model
from keras.callbacks import ModelCheckpoint, EarlyStopping
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications.inception_v3 import InceptionV3
import matplotlib.pyplot as plt
```

Para el caso de Google Colaboratory, es necesario montar la unidad de almacenamiento para cargar los sets de datos:

```
from google.colab import drive
drive.mount('/gdrive')
%cd /gdrive/
%cd MyDrive/PROYECTOS\ ML/
```

Especificamos la localización de cada set de datos necesarios para el entrenamiento, validación y prueba, así como también, establecemos las dimensiones a utilizar en el preprocesamiento:

```
train_data_dir = 'Dataset/train'
validation_data_dir = 'Dataset/validation'
test_data_dir = 'Dataset/test'
img_width, img_height, img_channels = 299, 299, 3
input_shape = (img_width, img_height, 3)
```

Hacemos uso de la clase *ImageDataGenerator* para crear generadores, los cuales se encargan de aplicar una serie de técnicas de transformación y preprocesamiento, tales como cambios de escala, rotación, traslación y aumento, con la finalidad de aumentar la cantidad de datos de entrenamiento y mejorar la capacidad del modelo para generalizar y evitar el sobreajuste.

```
validation_datagen = ImageDataGenerator(rescale=1./255,
                                       shear_range=0.2,
                                       zoom_range=0.2,
                                       horizontal_flip=True)
validation_datagen = ImageDataGenerator(rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)
```

Con ayuda de los generadores instanciados anteriormente, obtuvimos los sets de datos correctamente preparados y estructurados para el entrenamiento, validación y finalmente prueba.

```
train_dataset=validation_datagen.flow_from_directory(train_data_dir,
target_size=(img_width,          img_height),          batch_size=32,
class_mode='categorical');
validation_dataset=
validation_datagen.flow_from_directory(validation_data_dir,
target_size=(img_width,          img_height),          batch_size=32,
class_mode='categorical');
test_dataset=test_datagen.flow_from_directory(test_data_dir,
                                             target_size = (299, 299),
                                             batch_size = 32,
                                             shuffle=False,
                                             class_mode='categorical')
```

Extraemos 10 elementos del primer batch generado aleatoriamente en los pasos anteriores y mostramos el resultado

```
batch = train_dataset.next()
muestra = batch[0][:10]
labels = list(train_dataset.class_indices.keys())
etiquetas = np.argmax(batch[1], axis=1)

plt.figure(figsize=(14, 6))
for i in range(len(muestra)):
    plt.subplot(2, 5, i+1)
    # plt.axis('auto')
    plt.imshow(muestra[i])
    plt.xlabel(labels[etiquetas[i]])
plt.show()
```

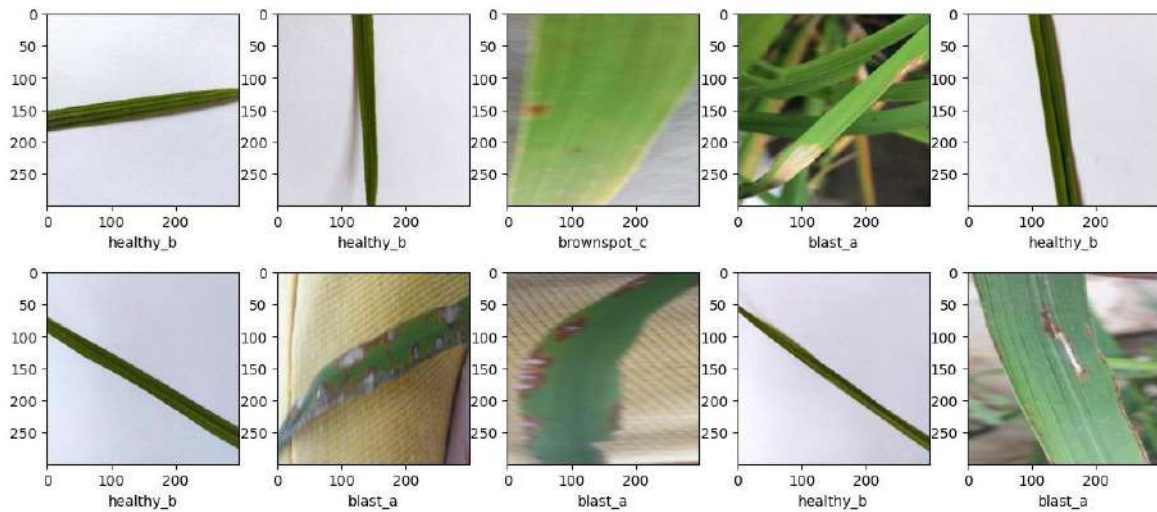



Figura 11

Imágenes pre procesadas

Las imágenes resultantes del preprocesamiento ya se encuentran organizadas e identificadas según su categoría, listas para el proceso de entrenamiento y validación. Este proceso se alinea a trabajos de investigación similares, tales como las de Uppal et al. (2021) los cuales aplicaron métodos y técnicas de redimensionamiento y aumento de datos antes del procesamiento.

4.2 Resultado específico 2

Para construir el modelo clasificación, instanciamos el modelo InceptionV3 en la variable “iv3_model” con el propósito de manipular su estructura.

```
iv3_model = InceptionV3(weights='imagenet', include_top=False,
input_shape=input_shape)
```

Recorremos las capas del modelo con el fin de evitar sobrescribir los pesos del modelo que fue previamente entrenado con la base de datos visual *ImageNet*

```
for layer in iv3_model.layers:
    layer.trainable = False
```

Con la función *output* obtenemos el tensor resultante de la capa convolucional del modelo instanciado, almacenándolo nuevamente en una variable para su posterior integración en un nuevo modelo.

```
res = iv3_model.output
```

El nuevo modelo es generado a partir del tensor anteriormente obtenido, a este le añadimos una capa de aplanamiento (Flatten) conectado a una capa densa (Dense) de 1024 unidades con función de activación “Relu”, la cual estará conectada a una última capa densa (Dense) de 3 unidades con activación “Softmax”. Esta última capa se encargará de la clasificación.

```
res = Flatten()(res)
res = Dense(1024, activation = 'relu')(res)
predictions = Dense(3, activation = 'softmax')(res)
```

Configuramos el modelo con InceptionV3 como capa de entrada, y el modelo almacenado en la variable “predictions” como capa de salida o decisión. Almacenamos el modelo resultante en la variable “model”.

```
model = Model(inputs = iv3_model.input, outputs = predictions)
```

Con la función *compile*, compilamos el modelo utilizando el algoritmo de optimización **Adam** y la entropía cruzada categórica como medida del error durante el entrenamiento de la red. Además, especificamos el monitoreo de la exactitud obtenida por el modelo en cada iteración o época del entrenamiento a través del parámetro *metrics*.

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

Antes de proceder con el entrenamiento, establecimos las callbacks *checkpoint* y *earlyStop*, las cuales son funciones que se ejecutarán durante ciertos momentos del entrenamiento, siendo:

- **Checkpoint:** Almacena el modelo entrenado cada vez que alcanza un punto máximo en el valor de exactitud con el set de validación durante el entrenamiento.
- **EarlyStop:** Detiene el entrenamiento cuando no se ha mejorado el indicador de exactitud con el set de validación durante un rango de iteraciones especificados en el parámetro *patience*, lo que ayuda a prevenir el sobreajuste del modelo.

```
checkpoint = ModelCheckpoint('Rice_InceptionV3_Best.h5',
                             monitor='val_accuracy',
                             save_best_only=True,
                             mode='max',
                             verbose=1)
earlyStop = EarlyStopping(monitor='val_accuracy', patience=5)
```


Para el entrenamiento, ingresamos el set de entrenamiento, el número de lotes que se procesarán en cada época de entrenamiento (*steps_per_epoch*), las épocas de entrenamiento (*epochs*), el set de validación, el número de lotes que se procesarán durante la evaluación (*validation_steps*), y finalmente los callbacks anteriormente explicados.

Con el método `fit`, ejecutamos el entrenamiento, lo que nos devuelve el historial de dicho entrenamiento con las métricas de monitoreo especificadas en la compilación. Almacenamos este historial en la variable “`history`”

```
history = model.fit(train_dataset, steps_per_epoch =
train_dataset.n//train_dataset.batch_size,
epochs = 30, validation_data = validation_dataset,
validation_steps =
validation_dataset.n//validation_dataset.batch_size,
callbacks=[checkpoint, earlyStop])
```

Tras el entrenamiento, corroboramos la exactitud del modelo con el dataset de prueba, el cual ha sido separado y no ha tenido exposición alguna al modelo durante el entrenamiento.

```
score = model.evaluate(test_dataset, steps =
test_dataset.n//test_dataset.batch_size)
print('Pérdida: ', score[0])
print('Precisión: ', score[1])
```

Como resultado, obtuvimos un 96,88% de exactitud con el dataset de prueba.

```
5/5 [=====] - 97s 24s/step - loss: 0.6521 -
accuracy: 0.9688
Pérdida: 0.6521178483963013
Precisión: 0.96875
```

Con ayuda de la librería **matplotlib.pyplot**, iteramos sobre la variable “`history`” para graficar la evolución de la exactitud y la pérdida del modelo a lo largo de las épocas de entrenamiento.

```
import matplotlib.pyplot as plt

# Graficar la evolución de la precisión y pérdida durante el entrenamiento
plt.plot(np.arange(1, 18), history.history['accuracy'],
label='Exactitud')
plt.plot(np.arange(1, 18), history.history['val_accuracy'],
label='Exactitud validación')
plt.plot(np.arange(1, 18), history.history['loss'], label='Pérdida')
```

```
plt.plot(np.arange(1, 18), history.history['val_loss'], label='Pérdida
validación')
plt.title('Métricas del entrenamiento')
plt.xlabel('Épocas')
plt.ylabel('Exactitud/Pérdida')
plt.legend()
plt.show()
```

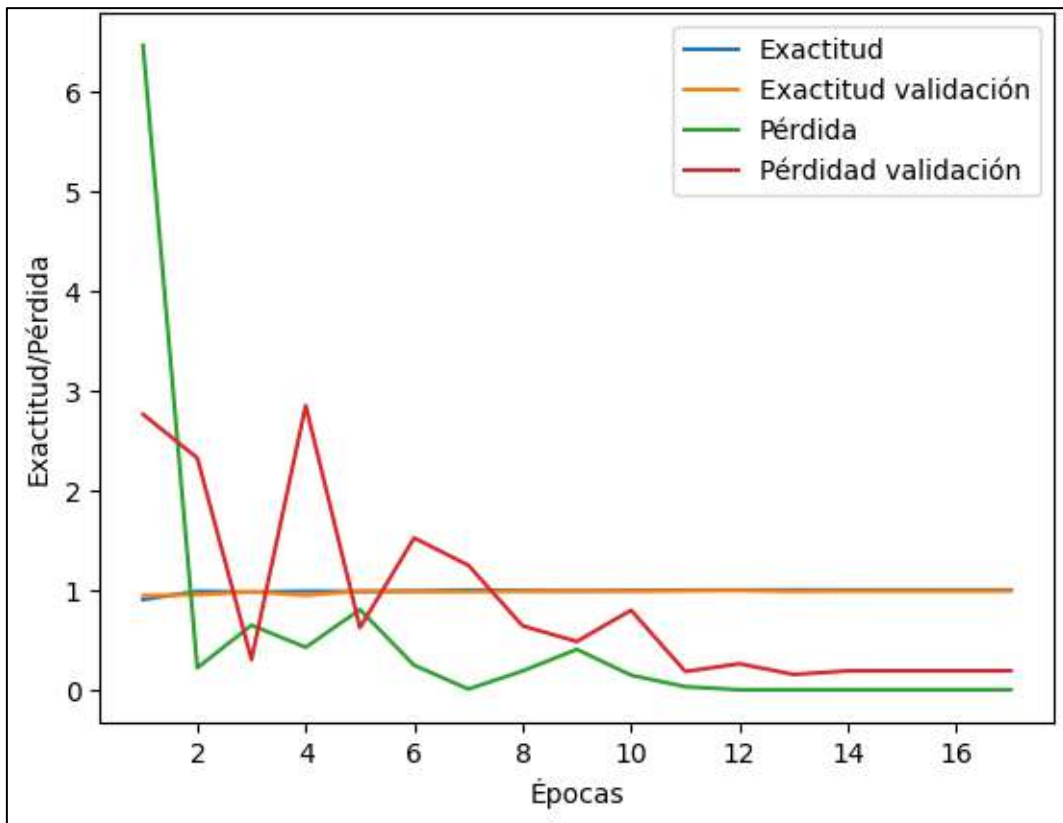


Figura 12

Métricas del entrenamiento

Como se puede observar en el gráfico del proceso de entrenamiento (Figura 13), los valores finales de la métrica de exactitud, tanto en el set de entrenamiento como de validación, estos no varían en demasía, lo que nos indica que no hay presencia de sobreajuste en el modelo.

Sharma et al. (2021) en su trabajo, elabora un modelo para detectar la enfermedad de la “hispa del arroz”. Ellos proponen una red neural convolucional con la que obtienen una exactitud de 96% en el set de entrenamiento y un 95% con el set de validación. En contraste con los resultados obtenidos por el modelo propuesto en esta investigación, podemos evidenciar mejoras, tanto con el set de entrenamiento (100%) y el set de validación (99.43%). Esto puede deberse a la complejidad del modelo preentrenado utilizado en el modelo que proponemos.

4.3 Resultado específico 3

Para determinar la precisión del modelo, instanciamos el modelo previamente entrenado y guardado con ayuda del callback **ModelCheckpoint**.

```
saved_model = keras.models.load_model('Rice_InceptionV3_Best.h5')
```

Generamos la predicción con el dataset de prueba, almacenando los resultados en la variable “prob”, de este último, obtenemos el conjunto de clases predichas y lo almacenamos en la variable “predicted_classes”.

```
prob = saved_model.predict(test_dataset)
predicted_classes = np.argmax(prob, axis=1)
```

Del dataset de prueba, extraemos las etiquetas correspondientes a cada clasificación, y obtenemos las clases a las que pertenece cada uno de los datos.

```
labels = list(test_dataset.class_indices.keys())
from sklearn import metrics
ref_classes = []
for index in range(len(test_dataset)):
    ref_classes.extend(np.argmax(test_dataset[index][1], axis=1))
```

Con los resultados verdaderos en la variable “ref_classes” y los resultados predichos por el modelo en la variable “predicted_classes”, así como las probabilidades obtenidas en la predicción en la variable “prob”, procedemos a someterlos a las métricas de evaluación de desempeño:

- Exactitud
- Coeficiente F1
- Precisión
- Recall
- Kappa de Cohen
- Area bajo de curva ROC (1 vs el resto)
- Area bajo de curva ROC (1 vs 1)
- Matriz de confusión

```
accuracy = metrics.accuracy_score(ref_classes, predicted_classes)
print('Exactitud: %f' % accuracy)

f1 = metrics.f1_score(ref_classes, predicted_classes, average='micro')
print('Coeficiente F1: %f' % f1)
```

```

precision = metrics.precision_score(ref_classes, predicted_classes,
average='micro')
print('Precisión: %f' % precision)

recall = metrics.recall_score(ref_classes, predicted_classes,
average='micro')
print('Recall: %f' % recall)

cohen_kappa = metrics.cohen_kappa_score(ref_classes, predicted_classes)
print('Kappa de Cohen: %f' % cohen_kappa)

roc_ovr = metrics.roc_auc_score(ref_classes, prob, multi_class="ovr")
print('ROC OVR: %f' % roc_ovr)

roc_ovo = metrics.roc_auc_score(ref_classes, prob, average='macro',
multi_class="ovo")
print('ROC OVO: %f' % roc_ovo)

matrix = metrics.confusion_matrix(ref_classes, predicted_classes)
matrix_disp = metrics.ConfusionMatrixDisplay(matrix,
display_labels=labels)
matrix_disp.plot()
plt.show()

```

Dándonos como resultados los siguientes valores:

```

Exactitud: 0.967914
Coeficiente F1: 0.967914
Precisión: 0.967914
Recall: 0.967914
Kappa de Cohen: 0.947538
ROC OVR: 0.992580
ROC OVO: 0.990570

```

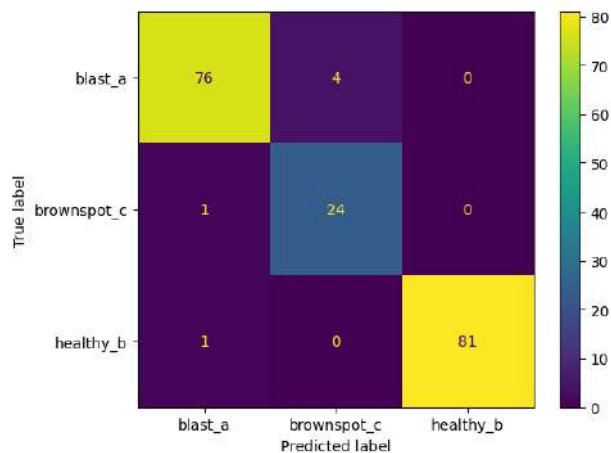


Figura 13

Matriz de confusión resultante

Estos resultados coinciden en gran medida con los resultados reportados por Latif et al. (2022) como puede ser el coeficiente F1, el cual alcanza un valor promedio de 0.9693, lo que resulta ligeramente superior al valor obtenido durante la presente investigación. Sin embargo, en cuanto a la exactitud promedio del modelo, Latif et al. (2022) obtiene un valor del 96.08%, siendo ligeramente inferior al 96.79% obtenida por nuestro modelo. Adicionalmente a estos resultados, se corrieron otros 2 modelos arquitecturas con un flujo de funcionamiento similar.

Para el primero modelo se usó un enfoque de transferencia de aprendizaje como en el presente caso, haciéndose uso del modelo pre entrenado ResNet50. Para el segundo modelo, se utilizó la arquitectura AlexNet, la cual cuenta con 8 capas de profundidad y con la que no se usó un enfoque de transferencia de aprendizaje, sino que se entrenó por completo con el mismo set de entrenamiento y validación.

Para asegurar la consistencia de las métricas de rendimiento, se seccionó el set de prueba en 6 grupos, para posteriormente, correr cada modelo con cada grupo del set de prueba.

Tras ejecutar las predicciones, se llegaron a los siguientes resultados (Tabla 2):

Tabla 2.
Métricas según grupo de evaluación

MODELO	GRUPO	EXACTITUD	F1	PRECISIÓN	RECALL	KAPPA DE COHEN	ROC OVR	ROC OVO
ALEXNET	1	0.9667	0.9667	0.9667	0.9667	0.9459	0.9953	0.9936
	2	0.9667	0.9667	0.9667	0.9667	0.9459	0.9906	0.9872
	3	0.9000	0.9000	0.9000	0.9000	0.8429	0.9811	0.9744
	4	0.9000	0.9000	0.9000	0.9000	0.8324	0.9576	0.9423
	5	0.9000	0.9000	0.9000	0.9000	0.8429	1.0000	1.0000
	6	0.9667	0.9667	0.9667	0.9667	0.9441	0.9844	0.9776
RESNET50	1	0.9667	0.9667	0.9667	0.9667	0.9441	0.9953	0.9936
	2	0.9667	0.9667	0.9667	0.9667	0.9441	1.0000	1.0000
	3	0.9000	0.9000	0.9000	0.9000	0.8324	0.9859	0.9808
	4	0.9333	0.9333	0.9333	0.9333	0.8901	0.9906	0.9872
	5	0.9667	0.9667	0.9667	0.9667	0.9441	0.9764	0.9679
	6	0.9000	0.9000	0.9000	0.9000	0.8266	0.9857	0.9840
INCEPTIONV3	1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	3	0.9667	0.9667	0.9667	0.9667	0.9459	0.9936	0.9936
	4	0.9333	0.9333	0.9333	0.9333	0.8901	0.9775	0.9647
	5	0.9333	0.9333	0.9333	0.9333	0.8936	0.9842	0.9852
	6	0.9667	0.9667	0.9667	0.9667	0.9451	1.0000	1.0000

Para determinar si existe diferencias significativas en los resultados provistos por los modelos propuestos, hicimos uso del método ANOVA (análisis de varianza), con el que se obtuvieron los siguientes resultados:

Tabla 3.
Resultados de la prueba ANOVA

		Suma de cuadrados	GL	Media cuadrática	F	SIG
Exactitud	Entre grupos	.004	2	.002	1.742	.209
	Dentro de grupos	.016	15	.001		
	Total	.020	17			
Coefficiente F1	Entre grupos	.004	2	.002	1.742	.209
	Dentro de grupos	.016	15	.001		
	Total	.020	17			
Precisión	Entre grupos	.004	2	.002	1.742	.209
	Dentro de grupos	.016	15	.001		
	Total	.020	17			
Recall	Entre grupos	.004	2	.002	1.742	.209
	Dentro de grupos	.016	15	.001		
	Total	.020	17			
Kappa de Cohen	Entre grupos	.011	2	.005	1.775	.203
	Dentro de grupos	.044	15	.003		
	Total	.055	17			
ROC OVR	Entre grupos	.000	2	.000	.693	.515
	Dentro de grupos	.002	15	.000		
	Total	.002	17			
ROC OVO	Entre grupos	.000	2	.000	.803	.466
	Dentro de grupos	.004	15	.000		
	Total	.004	17			

Tabla 4.
Estadísticos descriptivos

		N	Media	Desv. Desviación	Desv. Error	95% del intervalo de confianza para la media		Mínimo	Máximo
						Límite inferior	Límite superior		
Exactitud	Alexnet	6	0.93333	0.03651	0.01491	0.89501	0.97165	0.90000	0.96667
	Resnet50	6	0.93889	0.03277	0.01338	0.90450	0.97328	0.90000	0.96667
	InceptionV3	6	0.96667	0.02981	0.01217	0.93538	0.99795	0.93333	1.00000
	Total	18	0.94630	0.03456	0.00815	0.92911	0.96348	0.90000	1.00000
Coefficiente F1	Alexnet	6	0.93333	0.03651	0.01491	0.89501	0.97165	0.90000	0.96667
	Resnet50	6	0.93889	0.03277	0.01338	0.90450	0.97328	0.90000	0.96667
	InceptionV3	6	0.96667	0.02981	0.01217	0.93538	0.99795	0.93333	1.00000
	Total	18	0.94630	0.03456	0.00815	0.92911	0.96348	0.90000	1.00000
Precisión	Alexnet	6	0.93333	0.03651	0.01491	0.89501	0.97165	0.90000	0.96667
	Resnet50	6	0.93889	0.03277	0.01338	0.90450	0.97328	0.90000	0.96667
	InceptionV3	6	0.96667	0.02981	0.01217	0.93538	0.99795	0.93333	1.00000
	Total	18	0.94630	0.03456	0.00815	0.92911	0.96348	0.90000	1.00000
Recall	Alexnet	6	0.93333	0.03651	0.01491	0.89501	0.97165	0.90000	0.96667
	Resnet50	6	0.93889	0.03277	0.01338	0.90450	0.97328	0.90000	0.96667
	InceptionV3	6	0.96667	0.02981	0.01217	0.93538	0.99795	0.93333	1.00000
	Total	18	0.94630	0.03456	0.00815	0.92911	0.96348	0.90000	1.00000
Kappa de Cohen	Alexnet	6	0.89238	0.05815	0.02374	0.83136	0.95340	0.83240	0.94595
	Resnet50	6	0.89692	0.05629	0.02298	0.83784	0.95599	0.82659	0.94413
	InceptionV3	6	0.94579	0.04837	0.01975	0.89502	0.99655	0.89011	1.00000
	Total	18	0.91170	0.05686	0.01340	0.88342	0.93997	0.82659	1.00000
ROC OVR	Alexnet	6	0.98482	0.01503	0.00613	0.96905	1.00059	0.95758	1.00000
	Resnet50	6	0.98897	0.00827	0.00338	0.98029	0.99765	0.97643	1.00000
	InceptionV3	6	0.99254	0.00965	0.00394	0.98241	1.00266	0.97747	1.00000
	Total	18	0.98878	0.01115	0.00263	0.98323	0.99432	0.95758	1.00000
ROC OVO	Alexnet	6	0.97917	0.02045	0.00835	0.95771	1.00063	0.94231	1.00000
	Resnet50	6	0.98558	0.01106	0.00451	0.97397	0.99718	0.96795	1.00000
	InceptionV3	6	0.99059	0.01394	0.00569	0.97596	1.00522	0.96474	1.00000
	Total	18	0.98511	0.01547	0.00365	0.97742	0.99280	0.94231	1.00000

Según los valores de significancia observados en la tabla 3, y en términos de la prueba ANOVA utilizando un intervalo de confianza del 99%, se determinó que no existe diferencia significativa en las métricas de rendimiento obtenidas por los distintos modelos. Sin embargo, tras analizar los valores de los estadísticos descriptivos, específicamente la media y el error estándar de la media, es posible determinar que InceptionV3 alcanza en promedio valores más altos los cuales se desvían menos entre sí a comparación de los demás modelos. Esto se alinea a los resultados obtenidos por Simhadri & Kondaveeti (2023) en los que se observa un mejor valor de la exactitud promedio del modelo basado en InceptionV3 respecto a otros 15 modelos pre-entrenados en los cuales se incluyen Alexnet y Resnet50.

CONCLUSIONES

1. Logramos detectar las enfermedades más frecuentes que afectan las plantaciones de arroz mediante el reconocimiento de patrones visuales. A través del uso del modelo de red convolucional InceptionV3 y el análisis de un dataset de 1874 imágenes.
2. Identificamos exitosamente las características y patrones visuales de las enfermedades más frecuentes presentes en las plantaciones de arroz del sector Mishqiyacu. Mediante un exhaustivo análisis de imágenes, además identificamos rasgos distintivos asociados a cada enfermedad, lo que nos permitió optimizar el desarrollo de un modelo de reconocimiento eficiente.
3. Construimos e implementamos un modelo de reconocimiento de patrones visuales para la detección de las enfermedades más frecuentes en las plantaciones de arroz del sector Mishqiyacu. Haciendo uso de la transferencia de aprendizaje de la arquitectura de red convolucional InceptionV3, reconocida por su capacidad de extraer características relevantes de las imágenes. El modelo fue entrenado con un extenso dataset de 3 categorías y se aplicaron técnicas de preprocesamiento para mejorar los resultados.
4. Tras construir otros 2 modelos de reconocimiento de patrones visuales con arquitecturas AlexNet y ResNet50, comparamos los indicadores de rendimiento de los 3 modelos de reconocimiento de patrones visuales en la detección de las enfermedades más frecuentes en las plantaciones de arroz del sector Mishqiyacu. Después de evaluar sus indicadores de rendimiento con la prueba ANOVA, no se pudieron encontrar diferencias significativas entre los distintos grupos. Sin embargo, luego de analizar los valores estadísticos descriptivos y que, dada la media y el error estándar de la media, se determinó que InceptionV3 alcanza en promedio valores más altos los cuales se desvían menos entre sí a comparación de los demás modelos.

RECOMENDACIONES

1. Para futuros trabajos, recomendamos recopilar un conjunto de datos lo suficientemente grande y representativo de las enfermedades que se desea detectar en las plantaciones de arroz. Realiza un exhaustivo proceso de limpieza y preprocesamiento de los datos para garantizar la calidad y consistencia de las imágenes.
2. Recomendamos la utilización y evaluación diferentes arquitecturas de redes convolucionales para determinar cuál se ajusta mejor a las necesidades y al problema específico abordado. Considerando métricas de evaluación del rendimiento similares a la expuestas durante la presente investigación.
3. Recomendamos experimentar con diferentes valores de hiperparámetros, como tasas de aprendizaje, tamaño de lote y número de capas, para encontrar la configuración óptima que maximice el rendimiento del modelo.
4. Con el fin de aumentar los tipos de enfermedades que el modelo es capaz de detectar, recomendamos extender el set de datos con nuevos datos relacionados a enfermedades en las plantaciones de arroz que no hayan sido abarcadas en este trabajo.

REFERENCIAS BIBLIOGRÁFICAS

- Agus, F., Ihsan, Muh., Marisa Khairina, D. & Candra, K. P. (2018). ESforRPD2: Expert System for Rice Plant Disease Diagnosis. *F1000Research*, 7, 1902. <https://doi.org/10.12688/f1000research.16657.1>
- Albatayneh, O., Forslöf, L. & Ksaibati, K. (2020). Image Retraining Using TensorFlow Implementation of the Pretrained Inception-v3 Model for Evaluating Gravel Road Dust. *Journal of Infrastructure Systems*, 26(2). [https://doi.org/10.1061/\(asce\)is.1943-555x.0000545](https://doi.org/10.1061/(asce)is.1943-555x.0000545)
- Almasoud, A. S., Abdelmaboud, A., Eisa, T. A. E., Al Duhayyim, M., Elnour, A. A. H., Hamza, M. A., Motwakel, A. & Zamani, A. S. (2022). Artificial Intelligence-Based Fusion Model for Paddy Leaf Disease Detection and Classification. *Computers, Materials and Continua*, 72(1), 1391–1407. <https://doi.org/10.32604/cmc.2022.024618>
- Anuar, M. M., Halin, A. A., Perumal, T. & Kalantar, B. (2022). Aerial Imagery Paddy Seedlings Inspection Using Deep Learning. *Remote Sensing*, 14(2). <https://doi.org/10.3390/rs14020274>
- Arinichev, I. V., Polyanskikh, S. V., Volkova, G. V. & Arinicheva, I. V. (2021). Rice Fungal Diseases Recognition Using Modern Computer Vision Techniques. *International Journal of Fuzzy Logic and Intelligent Systems*, 21(1), 1–11. <https://doi.org/10.5391/IJFIS.2021.21.1.1>
- Arnal Barbedo, J. G. (2019). Plant disease identification from individual lesions and spots using deep learning. *Biosystems Engineering*, 180, 96–107. <https://doi.org/10.1016/j.biosystemseng.2019.02.002>
- Ashraf, T. & Khan, Y. N. (2020). Weed density classification in rice crop using computer vision. *Computers and Electronics in Agriculture*, 175. <https://doi.org/10.1016/j.compag.2020.105590>
- Atila, Ü., Uçar, M., Akyol, K. & Uçar, E. (2021). Plant leaf disease classification using EfficientNet deep learning model. *Ecological Informatics*, 61. <https://doi.org/10.1016/j.ecoinf.2020.101182>
- Bisong, E. (2019). Building Machine Learning and Deep Learning Models on Google Cloud Platform. In *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Apress. <https://doi.org/10.1007/978-1-4842-4470-8>

- Chen, J., Chen, J., Zhang, D., Sun, Y. & Nanehkaran, Y. A. (2020). Using deep transfer learning for image-based plant disease identification. *Computers and Electronics in Agriculture*, 173. <https://doi.org/10.1016/j.compag.2020.105393>
- Chen, J., Yin, H. & Zhang, D. (2020). A self-adaptive classification method for plant disease detection using GMDH-Logistic model. *Sustainable Computing: Informatics and Systems*, 28. <https://doi.org/10.1016/j.suscom.2020.100415>
- Debnath, O. & Saha, H. N. (2022). An IoT-based intelligent farming using CNN for early disease detection in rice paddy. *Microprocessors and Microsystems*, 94.
- Dhingra, G., Kumar, V. & Joshi, H. D. (2018). Study of digital image processing techniques for leaf disease detection and classification. *Multimedia Tools and Applications*, 77(15), 19951–20000. <https://doi.org/10.1007/s11042-017-5445-8>
- Elmitwally, N. S., Tariq, M., Khan, M. A., Ahmad, M., Abbas, S. & Alotaibi, F. M. (2022). Rice Leaves Disease Diagnose Empowered with Transfer Learning. *Computer Systems Science and Engineering*, 42(3), 1001–1014. <https://doi.org/10.32604/csse.2022.022017>
- Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, 145, 311–318. <https://doi.org/10.1016/j.compag.2018.01.009>
- Ghosal, S. & Sarkar, K. (2020). Rice Leaf Diseases Classification Using CNN With Transfer Learning. *IEEE Calcutta Conference (CALCON)*, 230–236.
- INEI. (2021). *Perú: Panorama Económico Departamental*. <https://m.inei.gob.pe/media/MenuRecursivo/boletines/02-informe-tecnico-panorama-economico-departamental-dic-2021.pdf>
- Joseph, F. J. J., Nonsiri, S. & Monsakul, A. (2021). Keras and TensorFlow: A Hands-On Experience. In *EAI/Springer Innovations in Communication and Computing* (pp. 85–111). Springer Science and Business Media Deutschland GmbH. https://doi.org/10.1007/978-3-030-66519-7_4
- Kakani, V., Nguyen, V. H., Kumar, B. P., Kim, H. & Pasupuleti, V. R. (2020). A critical review on computer vision and artificial intelligence in food industry. In *Journal of Agriculture and Food Research* (Vol. 2). Elsevier B.V. <https://doi.org/10.1016/j.jafr.2020.100033>
- Karar, M. E., Alsunaydi, F., Albusaymi, S. & Alotaibi, S. (2021). A new mobile application of agricultural pests recognition using deep learning in cloud computing system. *Alexandria Engineering Journal*, 60(5), 4423–4432. <https://doi.org/10.1016/j.aej.2021.03.009>

- Kaya, A., Keceli, A. S., Catal, C., Yalic, H. Y., Temucin, H. & Tekinerdogan, B. (2019). Analysis of transfer learning for deep neural network based plant classification models. *Computers and Electronics in Agriculture*, 158, 20–29. <https://doi.org/10.1016/j.compag.2019.01.041>
- KC, K., Yin, Z., Wu, M. & Wu, Z. (2019). Depthwise separable convolution architectures for plant disease classification. *Computers and Electronics in Agriculture*, 165. <https://doi.org/10.1016/j.compag.2019.104948>
- Latif, G., Abdelhamid, S. E., Mallouhy, R. E., Alghazo, J. & Kazimi, Z. A. (2022). Deep Learning Utilization in Agriculture: Detection of Rice Plant Diseases Using an Improved CNN Model. *Plants*, 11(17). <https://doi.org/10.3390/plants11172230>
- Ma, J., Du, K., Zheng, F., Zhang, L., Gong, Z. & Sun, Z. (2018). A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network. *Computers and Electronics in Agriculture*, 154, 18–24. <https://doi.org/10.1016/j.compag.2018.08.048>
- MIDAGRI. (2021a). *Marco Orientador De Cultivos 2021*. https://siea.midagri.gob.pe/portal/phocadownload/datos_y_estadisticas/informacion_especializada/marco_orientador_de_cultivos_2021-2022.pdf
- MIDAGRI. (2021b). *Observatorio de las Siembras y Perspectivas de la Producción - Arroz*. <https://bit.ly/37hCBDJ>
- N, K., Narasimha Prasad, L. V., Pavan Kumar, C. S., Subedi, B., Abraha, H. B. & Sathishkumar, V. E. (2021). Rice leaf diseases prediction using deep neural networks with transfer learning. *Environmental Research*, 198. <https://doi.org/10.1016/j.envres.2021.111275>
- O'Shea, K. & Nash, R. (2015). *An Introduction to Convolutional Neural Networks*. <http://arxiv.org/abs/1511.08458>
- Patayon, U. B. & Crisostomo, R. V. (2021). Automatic Identification of Abaca Bunchy Top Disease using Deep Learning Models. *Procedia Computer Science*, 179, 321–329. <https://doi.org/10.1016/j.procs.2021.01.012>
- Patrício, D. I. & Rieder, R. (2018). Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review. In *Computers and Electronics in Agriculture* (Vol. 153, pp. 69–81). Elsevier B.V. <https://doi.org/10.1016/j.compag.2018.08.001>
- Picon, A., Seitz, M., Alvarez-Gila, A., Mohnke, P., Ortiz-Barredo, A. & Echazarra, J. (2019). Crop conditional Convolutional Neural Networks for massive multi-crop plant disease

classification over cell phone acquired images taken on real field conditions. *Computers and Electronics in Agriculture*, 167. <https://doi.org/10.1016/j.compag.2019.105093>

Poornappriya, T. S. & Gopinath, R. (2020). Rice plant disease identification using artificial intelligence approaches. *International Journal of Electrical Engineering and Technology (IJEET)*, 11(10), 392–402. <https://doi.org/10.34218/IJEET.11.10.2020.050>

Rahman, C. R., Arko, P. S., Ali, M. E., Iqbal Khan, M. A., Apon, S. H., Nowrin, F. & Wasif, A. (2020). Identification and recognition of rice diseases and pests using convolutional neural networks. *Biosystems Engineering*, 194, 112–120. <https://doi.org/10.1016/j.biosystemseng.2020.03.020>

Randles, B. M., Paschetto, I. V., Golshan, M. S. & Borgman, C. L. (2017). Using the Jupyter Notebook as a Tool for Open Science: An Empirical Study. *2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, 29(2). <https://doi.org/10.1016/j.future.2011.08.004>

Santa María Pinedo, J. C., Ríos López, C. A., Rodríguez Grández, C. & García Estrella, C. W. (2021). Reconocimiento de patrones de imágenes a través de un sistema de visión artificial en MATLAB. *Revista Científica de Sistemas e Informática*, 1(2), 15–26. <https://doi.org/10.51252/rcsi.v1i2.131>

Sharma, Rahul, Singh, A., Kavita, Jhanjhi, N. Z., Masud, M., Jaha, E. S. & Verma, S. (2022). Plant disease diagnosis and image classification using deep learning. *Computers, Materials and Continua*, 71(2), 2125–2140. <https://doi.org/10.32604/cmc.2022.020017>

Sharma, Rishabh, Kukreja, V. & Kadyan, V. (2021). Hispa rice disease classification using convolutional neural network. *2021 3rd International Conference on Signal Processing and Communication, ICPSC 2021*, 377–381. <https://doi.org/10.1109/ICSPC51351.2021.9451800>

Simhadri, C. G. & Kondaveeti, H. K. (2023). Automatic Recognition of Rice Leaf Diseases Using Transfer Learning. *Agronomy*, 13(4). <https://doi.org/10.3390/agronomy13040961>

Sowmyalakshmi, R., Jayasankar, T., Ayyem Pillai, V., Subramaniyan, K., Pustokhina, I. V., Pustokhin, D. A. & Shankar, K. (2021). An optimal classification model for rice plant disease detection. *Computers, Materials and Continua*, 68(2), 1751–1767. <https://doi.org/10.32604/cmc.2021.016825>

Sravan, V., Swaraj, K., Meenakshi, K. & Kora, P. (2021). A deep learning based crop disease classification using transfer learning. *Materials Today: Proceedings*. <https://doi.org/10.1016/j.matpr.2020.10.846>

- Sucari León, R., Aroquipa Durán, Y., Quina Quina, L. D., Quispe Yapo, E., Sucari León, A. & Huanca Torres, F. A. s. (2020). Visión artificial en reconocimiento de patrones para clasificación de frutas en agronegocios. *PURIQ*, 2(2). <https://doi.org/10.37073/puriq.2.2.76>
- Sujatha, R., Chatterjee, J. M., Jhanjhi, N. Z. & Brohi, S. N. (2021). Performance of deep learning vs machine learning in plant leaf disease detection. *Microprocessors and Microsystems*, 80. <https://doi.org/10.1016/j.micpro.2020.103615>
- Tian, H., Wang, T., Liu, Y., Qiao, X. & Li, Y. (2020). Computer vision technology in agricultural automation —A review. In *Information Processing in Agriculture* (Vol. 7, Issue 1, pp. 1–19). China Agricultural University. <https://doi.org/10.1016/j.inpa.2019.09.006>
- Tian, L., Xue, B., Wang, Z., Li, D., Yao, X., Cao, Q., Zhu, Y., Cao, W. & Cheng, T. (2021). Spectroscopic detection of rice leaf blast infection from asymptomatic to mild stages with integrated machine learning and feature selection. *Remote Sensing of Environment*, 257. <https://doi.org/10.1016/j.rse.2021.112350>
- Tiwari, V., Joshi, R. C. & Dutta, M. K. (2021). Dense convolutional neural networks based multiclass plant disease detection and classification using leaf images. *Ecological Informatics*, 63. <https://doi.org/10.1016/j.ecoinf.2021.101289>
- Too, E. C., Yujian, L., Njuki, S. & Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture*, 161, 272–279. <https://doi.org/10.1016/j.compag.2018.03.032>
- Uppal, A., Kumar Trivedi, S. & Tewari, G. (2021). LEAF DISEASE DIAGNOSIS MODEL OF RICE PLANT USING IMAGE PROCESSING AND DEEP CNN ARCHITECTURE. *International Journal of Mechanical Engineering*, 6(3).
- Wang, Y., Wang, H. & Peng, Z. (2021). Rice diseases detection and classification using attention based neural network and bayesian optimization. *Expert Systems with Applications*, 178. <https://doi.org/10.1016/j.eswa.2021.114770>
- Wei, J., Cui, J., Xu, Y., Li, J., Lei, X., Gao, W. & Chen, Y. (2022). Social Life Cycle Assessment of Major Staple Grain Crops in China. *Agriculture*, 12(4), 535. <https://doi.org/10.3390/agriculture12040535>
- Xiong, Y., Liang, L., Wang, L., She, J. & Wu, M. (2020). Identification of cash crop diseases using automatic image segmentation algorithm and deep learning with expanded dataset. *Computers and Electronics in Agriculture*, 177. <https://doi.org/10.1016/j.compag.2020.105712>

Zareiforush, H., Minaei, S., Alizadeh, M. R. & Banakar, A. (2016). Qualitative classification of milled rice grains using computer vision and metaheuristic techniques. *Journal of Food Science and Technology*, 53(1), 118–131. <https://doi.org/10.1007/s13197-015-1947-4>

Reconocimiento de patrones visuales para detección de enfermedades en plantaciones de arroz del sector Mishquiyacu - Morales

por Gabriela Sangama

Fecha de entrega: 16-abr-2024 11:21a.m. (UTC-0500)

Identificador de la entrega: 2351788914

Nombre del archivo: sis_Corregio_Gabriela_Abigail_Sangama_Vela_V2_OFICIAL_16-04.docx (2.97M)

Total de palabras: 12262

Total de caracteres: 72920

Reconocimiento de patrones visuales para detección de enfermedades en plantaciones de arroz del sector Mishquiyacu - Morales

INFORME DE ORIGINALIDAD



FUENTES PRIMARIAS

1	tesis.unsm.edu.pe Fuente de Internet	6%
2	repositorio.unsm.edu.pe Fuente de Internet	1%
3	repositorio.umsa.bo Fuente de Internet	1%
4	repository.unad.edu.co Fuente de Internet	1%
5	docs.microsoft.com Fuente de Internet	1%
6	stackoverflow.com Fuente de Internet	1%
7	Submitted to Pontificia Universidad Catolica del Ecuador - PUCE Trabajo del estudiante	<1%
8	www.pcm.gob.pe Fuente de Internet	<1%