



Universidad
Señor de Sipán

**FACULTAD DE INGENIERÍA, ARQUITECTURA Y
URBANISMO**

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

TESIS

**Evaluación de la eficiencia de técnicas de aprendizaje
automático para la identificación de enfermedades en los
cultivos de caña de azúcar**

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

Autor:

Bach. Reupo Atto, Paul Gianmarco

ORCID: <https://orcid.org/0000-0002-7633-298X>

Asesor:

Mg. Bances Saavedra, David Enrique

ORCID: <https://orcid.org/0000-0002-7164-8918>

Línea de Investigación:

Infraestructura, Tecnología y Medio Ambiente

Pimentel – Perú 2023

**EVALUACIÓN DE LA EFICIENCIA DE TÉCNICAS DE APRENDIZAJE
AUTOMÁTICO PARA LA IDENTIFICACIÓN DE ENFERMEDADES EN LOS
CULTIVOS DE CAÑA DE AZÚCAR**

Aprobación del jurado

DR. FORERO VARGAS, MANUEL GUILLERMO
Presidente del Jurado de Tesis

MG. MEJIA CABRERA, HEBER IVAN
Secretario del Jurado de Tesis

MG. TUESTA MONTEZA, VÍCTOR ALEXCI
Vocal de Jurado

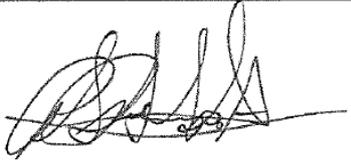
DECLARACIÓN JURADA DE ORIGINALIDAD

Quien suscribe la **DECLARACIÓN JURADA**, soy egresado del Programa de Estudios de la Escuela de Ingeniería de Sistemas de la Universidad Señor de Sipán S.A.C, declaro bajo juramento que soy autor del trabajo titulado:

**EVALUACIÓN DE LA EFICIENCIA DE TÉCNICAS DE APRENDIZAJE
AUTOMÁTICO PARA LA IDENTIFICACIÓN DE ENFERMEDADES EN LOS
CULTIVOS DE CAÑA DE AZÚCAR.**

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 (CIEI USS) conforme a los principios y lineamientos detallados en dicho documento, en relación a las citas y referencias bibliográficas, respetando al 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:

Reupo Atto, Paul Gianmarco	DNI: 70895786	
----------------------------	------------------	---

Pimentel, 14 de abril de 2023.

Dedicatoria

A mis padres, porque gracias a su esfuerzo he podido lograr cada uno los objetivos que me he planteado en la vida, brindándome su apoyo y afecto tiempos grises que a veces nos toca enfrentar durante nuestra estadía en la Tierra.

Agradecimientos

Agradezco a todos los docentes y maestros que han aportado en mi instrucción durante mi formación académica, desde el jardín de infancia hasta hoy, porque ellos complementaron las enseñanzas de mis padres, llegando hasta este punto de mi vida donde finalmente culmino mi formación universitaria.

Índice	
Dedicatoria	iv
Agradecimientos	v
Resumen	vii
Abstract	viii
I. INTRODUCCIÓN	9
1.1 Realidad Problemática.	9
1.2 Formulación del problema	21
1.3 Hipótesis	21
1.4 Objetivos	21
1.5 Teorías relacionadas al tema	22
II. MATERIALES Y MÉTODO	37
1.1 Tipo y Diseño de Investigación.	37
1.2 Variables, Operacionalización	37
1.3. Población y muestra.	39
1.4 Técnicas e instrumentos de recolección de datos, validez y confiabilidad.	40
1.5 Procedimiento de análisis de datos.	41
1.6. Criterios éticos.	43
III. RESULTADOS.	43
2.2 Discusión de resultados.	49
2.3. Aporte práctico.	50
IV. CONCLUSIONES Y RECOMENDACIONES	66
3.1. Conclusiones.	66
3.2 Recomendaciones.	67

Resumen

La agricultura en el Perú está ampliamente diversificada al haber diferentes regiones naturales, teniendo como consecuencia una gran variedad de cultivos, cuyos frutos surten los mercados o son transformados en otros productos por la actividad industrial, siendo componente importante de la economía nacional, brindando empleo a una buena parte de la población económicamente activa. Uno de estos cultivos es la caña de azúcar, históricamente relevante en la economía nacional, existiendo antes de la reforma agraria, extensas haciendas en el territorio nacional, en el caso de Lambayeque las haciendas de Tuman, Pomalca, Cayaltí, etc. Dentro de las actividades realizadas para la producción de azúcar, está el cultivo de *saccharum officinarum*, conocida coloquialmente como caña azúcar, para lo cual se debe dar vital importancia al control de plagas y enfermedades de las plantaciones, las cuales se ven amenazadas de forma constante, siendo una de las consecuencias la erradicación de buena parte de la plantación, ocasionando cuantiosas pérdidas económicas para los agricultores y disminución de la producción para las azucareras. El aprendizaje automático ha sido utilizado para resolver diversos problemas, en los cuales se requiera el análisis de grandes volúmenes de información, por ello podría ayudar en la temprana detección de estas amenazas. Entre las técnicas de aprendizaje automático que han demostrado gran eficiencia para estos problemas relacionados a visión por ordenador. Por esto esta investigación se centró en abordar este problema, integrando estas técnicas, para lograr el margen de error más pequeño posible en la clasificación de imágenes entre plantas sanas y enfermas, para ello se utilizó imágenes de muestra de las hojas de diferentes ejemplares, sanos y enfermos, de la especie *saccharum officinarum*, las cuales se emplearán en el entrenamiento y prueba de estos algoritmos, los cuales son tres redes convolucionales ampliamente usadas en visión por computador MobileNet, VGGNET16, ResNet50 y una propuesta.

Palabras Clave: Aprendizaje automático, visión por ordenador, enfermedad, caña de azúcar, redes neuronales convolucionales, clasificación de imágenes.

Abstract

Agriculture in Peru is widely diversified due to its different natural regions, resulting in a wide variety of crops, whose fruits are populated by markets or transformed into other products by industrial activity, being an important component of the national economy, allowing the employment of a large part of the economically active population. That is why it's important to adapt the technology currently available, in order to optimize production and reduce costs. One of these crops is sugar cane, historically relevant in the national economy, existing before land reform, extensive farms in the national territory, in the particular case of Lambayeque the farms of Tuman, Pomalca, Cayaltí, El Choloque and Batan Grande. Each counting on their respective factory, which remain to this day. Among the activities carried out for sugar production, is the cultivation of sugar cane (*saccharum officinarum*), for which special attention should be paid to the control of pests and diseases of plantations, which are constantly threatened, being one of the consequences the eradication of much of the plantation, causing large economic losses for farmers and decrease in production for sugar mills. Machine learning has been used to solve a variety of problems, where large volumes of information are required, so it could assist in early detection of these threats. Machine learning techniques that have proven high efficiency for these computer vision-related problems include convolutional neural networks and support vector machine. For this reason this research focused on addressing this problem, integrating these techniques, to achieve the smallest possible margin of error in the classification of images between healthy and sick plants, for this purpose sample images of the leaves of different specimens, healthy and sick, of the species *saccharum officinarum*, which will be used in the training and testing of these algorithms

Keywords: Machine Learning, convolutional neural network, computer visión, sugarcane, disease, image classification.

I. INTRODUCCIÓN

1.1 Realidad Problemática.

Para el año 2019, se pronosticó que la producción de caña de azúcar en 13.2 millones de toneladas métricas, excediendo en un 17 % a la que hubo en años anteriores. La producción de la caña de azúcar en el Perú, está focalizada en su mayoría en los valles de la costa norte, siendo la producción en el departamento de La Libertad un 52 %, le sigue el departamento de Lambayeque, con un 30 % y luego el departamento de Lima, con un 17 %. En condiciones de clima favorables, la producción de caña en el Perú puede darse durante todo el año. En promedio, el precio de la caña de azúcar por tonelada métrica es 1004.86 soles, siendo este el precio dado por los productores. [1]

La caña de azúcar es un cultivo que es altamente afectado por las enfermedades si no se tienen medidas de contingencia, por ser generalmente sembrada a modo de monocultivo extensivo de una misma variedad, esto tiene como consecuencia, que, si la plantación no tiene resistencia a un agente patógeno, toda esta se verá afectada por el. [2]

El control de las posibles plagas y las enfermedades en la industria del agro, es un tema de suma importancia. Existen sistemas inteligentes con visión de computadora integrados, para la identificación de plagas y enfermedades. Por otro lado, las tecnologías que involucran aprendizaje automático enfrentan ciertos problemas, como limitaciones en la capacidad de información con la que trabajan, elevados costos en la arquitectura de las redes y entradas de muy alta calidad para su análisis. [3]

Entre las enfermedades más dañinas a este cultivo, está el “carbón”, presentándose esta infección en el extremo superior pudiéndose extender por toda la planta. Esta plaga es causada por el hongo *ustilago scitaminea*. También están “la mancha del anillo”, siendo el causante el hongo *leptosphaeria sacchari*, apareciendo en hojas viejas pequeñas manchas que, con el tiempo, aumentan de tamaño, cubriendo toda la hoja; la “mancha del ojo”, producida por el hongo *bipolaris sacchari*, caracterizada por la aparición de manchas rojizas en las hojas de la planta; el “cogollo retorcido”, causada por el hongo *gibberella*

moniliformis, la cual ocasiona una pudrición grave y posterior muerte de la planta. Estas enfermedades afectan a gran parte de los países productores de este cultivo. [4]Estas enfermedades comparten la característica de presentar síntomas visibles en hojas y tallos, lo cual puede ser aprovechado para su detección temprana y posterior tratamiento.

Siendo el caso que en el departamento de Lambayeque hay ingenios productores de azúcar, es importante utilizar la tecnología disponible en esta actividad productiva, permitiendo mejora de la calidad y disminución de los costos. El aprendizaje automático puede ayudar en la detección mediante la clasificación de enfermedades presentes en los cultivos de caña, a través de algoritmos de clasificación de imágenes, sin incurrir en el costo de contratar a un experto que evalúe todas las plantaciones en búsqueda de posibles enfermedades o plagas. Los algoritmos de clasificación, permiten procesar y clasificar rápidamente gran cantidad de información a diferencia de un experto humano, que le tomaría varios días, hasta semanas. Entre las dificultades que podemos encontrar al utilizar el aprendizaje automático, están el tiempo, pudiendo tardar mucho en el entrenamiento de la red neuronal y evaluación de resultados, además que esto depende mucho del hardware que se esté empleando para esta tarea. Requiriendo una buena unidad de procesamiento para aliviar esta labor. Por lo que es necesario determinar que algoritmo es el más eficiente, teniendo en cuenta no solo lo apegado a la realidad de sus resultados de clasificación, sino el costo en cuanto a procesador y memoria para ello.

Con este trabajo de investigación se tuvo por objetivo encontrar una alternativa tecnológica, para la detección de las enfermedades en los cultivos de caña de azúcar utilizando algoritmos de aprendizaje automático, la cual debe tener un margen de error lo más pequeño posible. Para ello se seleccionó cuatro algoritmos de aprendizaje automático, siendo estos Support vector machine y tres redes neuronales convolucionales, siendo estas MobileNet, VGGNET, ResNet, además de una arquitectura propia, estos fueron seleccionados por conveniencia, al haber mostrado buen desempeño en la documentación revisada.

F.O. Sweetwilliams, V.O. Matthews, E. Adetiba, D.T. Babalola, V. Akande, (2019), realizaron la investigación "Detection of Sigatoka Disease in Plantain Using IoT and Machine Learning Techniques", en Nigeria. El problema que enfrentaron fue la detección en fases incipientes de la enfermedad de la sigatoka en plantaciones de plátano, mediante la obtención de indicadores obtenidos de sensores que utilizan tecnología de internet de las cosas. Para ello se capturaron cien imágenes de hojas, la mitad sana y la mitad con signos de enfermedad, luego pasan por un pre procesamiento reduciendo su resolución a 250x150 píxeles, después se les realiza la extracción de características con scalable vector descriptor (SVD) y histogram oriented gradient (HOG), estos se utilizan para obtener la distribución espacial del color, luego se clasifica como saludable o no saludable con la ayuda de la red neuronal artificial - perceptron multicapa, siendo su estructura de 16 capas de entradas para el SVD y 81 capas para el HOG, de 10 a 100 capas ocultas de neuronas, dos capas de neuronas de salida finalmente el output es enviado al sistema integrado de internet de las cosas. El desempeño se evaluó utilizando el error cuadrático medio y la exactitud. El mejor resultado se obtuvo con el HOG, siendo este de un 98 % de exactitud y el error cuadrático medio de 0.0208 a 40 neuronas en la capa oculta.

G. Sambasivam y Geoffrey Duncan Opiyo, (2020), realizaron la investigación "A predictive machine learning application in agriculture: cassava disease detection and classification with imbalanced dataset using convolutional neural networks", en Egipto, tratando el problema de la utilización de machine learning para la detección de enfermedades de la yuca. Para resolver este problema, eligieron usar redes neuronales convolucionales. Para ello se utilizó 10,000 imágenes etiquetadas de un dataset existente, que fue recolectado durante una encuesta en Uganda. La arquitectura del modelo fue de tres capas convolucionales y cuatro capas totalmente conectadas. Para las capas de salida, se consideraron cuatro enfermedades/plagas presentes en la yuca, las cuales son: el ácaro verde (AV), larvas de mosca blanca (LMB), la enfermedad del mosaico (EM) y la plaga

bacterial (PB). Para evaluar el desempeño del modelo, se utilizó la precisión, recall, valor F y exactitud, obteniendo un resultado promedio de 94%.

RuhuaLu, YalanLi, PanYang y WenfenZhang, (2021), realizaron la investigación titulada "Facial recognition based on convolutional neural network", en China. La problemática a tratar aquí fue el reconocimiento de expresiones faciales, siendo el movimiento de los músculos faciales un factor que podría ser de ayuda en esta tarea. Utilizaron los datasets de expresiones faciales FER2013, compuesto de treinta y cinco mil ochocientas ochenta y seis imágenes, conteniendo expresiones como "furioso", "disgustado", "temeroso", "triste", "acongojado", "relajado", etc. El segundo es el dataset JAFFE, liberado en mil novecientos noventa y ocho, conteniendo doscientos trece archivos de imagen, divididos en siete expresiones y por último el dataset CK+, lanzado en el año dos mil diez, siendo este un recurso libre. Se cambió la resolución de todas las imágenes a un estándar de 48x48 píxeles. Posteriormente usaron la red convolucional VGGnet, utilizando un total de 2071 imágenes en el entrenamiento y 168 en la evaluación modelo. Se utilizó kernel de convulsión pequeño de 3x3, capa de pool máximo de 2x2, teniendo al final esta red dieciséis y diecinueve capas. Al final se obtuvo una precisión del 83.57%, habiendo sido entrenado doscientas veces.

Suvajit Dutta Shalva Rai and Vijayarajan V, (2017), realizaron el estudio comparative titulado "A Comparative Study of Deep Learning Models for Medical Image Classification", en India. Aquí el problema que abordan es la clasificación de imágenes obtenidas de tomografías computarizadas (TC) de enfisemas y retinopatía diabética (RD), las cuales son complicadas de clasificar por métodos simples. Se empezó realizando procesamiento de las imágenes, acompañado de agrupamiento por K-medios, logrando obtener los niveles de gravedad de las enfermedades, posteriormente, usaron redes neuronales artificiales para conseguir resultados de clasificación primordiales, estos resultados son comparados con los obtenidos a través de redes neuronales profundas. Inicialmente tuvieron que tratar con

problemas en las imágenes, como desenfoco, ruido o alto contraste, esto complica la obtención de las peculiaridades requeridas derivando esto en una incorrecta clasificación. Como contramedida, optaron por cambiar las imágenes a escala de grises y la detección de bordes, lo cual es finalmente usada como entrada para los algoritmos de aprendizaje automático. Para lograr una validación del dataset, se utilizó como método de agrupamiento K-medias. Compararon los resultados obtenidos de redes neuronales profundas (RNP), redes neuronales profundas (RNP) y red neuronal prealimentada (RNPA), obteniendo en un principio una exactitud de 62.7 % (RNPA), 89.6 % (RNP) y 76.4 % (RNC), para RD y 42% (RNPA), 86.3 % (RNP) y 78 % (RNC) para TC.

Yunfei Lai, (2019), realizó la investigación titulada "A Comparison of Traditional Machine Learning and Deep Learning in Image Recognition", en China. El objetivo de su investigación fue el reconocimiento de imágenes de números escritos a mano con algoritmos de aprendizaje automático, como lo son máquina de vector de soporte (MSV) y red neuronal profunda (RNP). Para realizar su investigación utilizó como dataset Mnist dataset, el cual contiene imágenes digitalizadas de escritura a mano. Estos dígitos fueron redimensionados a 28x28 píxeles, habiendo utilizado 60,000 de las 70,000 imágenes contenidas en el dataset. El primer algoritmo utilizado fue MSV, el cual es comúnmente usado en problemas de clasificación lineal, tardando el entrenamiento del modelo 11.63 minutos, con exactitud entre 90.6 % y 98 %, notando al final que MSV es deficiente para ciertos números especiales, como el caso de los números 4 y 9, 3 y 5, 7 y 9. Luego utilizó RNP, el cual utiliza agrupamiento, convolución y otros recursos, logrando obtener de forma eficiente las peculiaridades de las imágenes, en este caso particular. Al realizar el experimento, este modelo tardó 11 horas, 50 minutos y 41.2 segundos, ejecutando 5,000 iteraciones en los datos, teniendo una tasa de exactitud que logró un 100 %.

Chunyan Cui, Shunxin Wang, Jian Zhou, Haojiang Li y Lizhi Liu, (2019), realizaron la investigación titulada "Machine Learning Analysis of Image Data Based on Detailed MR

Image Reports for Nasopharyngeal Carcinoma Prognosis”, en China. En esta investigación abordaron el uso de algoritmos de clasificación, pertenecientes al aprendizaje automático, en pacientes con carcinoma nasofaríngeo según sus resonancias magnéticas, siendo el caso que el sistema tradicional para la clasificación de estados de carcinoma nasofaríngeo que se utilizaba omitía características que podrían ayudar a un mejor diagnóstico. Para ello Se clasificó primero a los pacientes como: patológicamente diagnosticado, no evidencia de futura metástasis y paciente en tratamiento, para utilizar sus resonancias y obtener las características de sus estadios, utilizando el juicio de expertos que diagnostiquen según su criterio, siendo estos datos necesarios para verificar la precisión del algoritmo. La información de los pacientes se dividió usando validación cruzada de forma aleatoria. Utilizaron el algoritmo de AutoML, el cual viene incluido dentro del paquete H2O de R, fijando un tiempo de 7,200 segundos para el entrenamiento. También se utilizó curva ROC para mostrar el desempeño del modelo de clasificación, llegando a la conclusión que se logró un mejor diagnóstico que el método que se estaba empleando al emplear AutoML.

Indri Neforawati, Nanna Suryana Herman, Othman Mohd, (2018), realizaron la investigación “Precision agriculture classification using convolutional neural networks for paddy growth level”, en Indonesia. Aquí abordan el tema de la agricultura de precisión, siendo este el pilar de la agricultura moderna, en este caso, aplicado en cultivos de arroz. El tema de la clasificación, tanto del cultivo, fruto y semillas, ha sido tratado por numerosos investigadores, utilizando para ello modelos de clasificación de machine learning. En su investigación, la clasificación en cultivos de arroz fue realizada mediante datasets de imágenes y redes neuronales convolucionales. En su método propuesto, usaron un dataset Open Source de la FAO aunado con otros datasets, incluyendo uno propio, pre procesando las imágenes, mejorando su calidad y lograr que su lectura sea más sencilla, separaron el dataset en tres partes: validacion, entrenamiento y prueba, luego se utiliza una red neuronal convolucional como tecnica de clasificación el cual determinará el nivel de crecimiento del arrozal, siendo su clasificación recién sembrado, trasplantado, en florecimiento y maduro.

Para la evaluación del desempeño usaron F-score, recall y precisión. También usaron máquina de vector de soporte, con la finalidad de tenerlo como punto de referencia a comparar con las redes neuronales convolucionales de su método. Como resultado general obtuvieron un 82 % de exactitud con las redes neuronales convolucionales, en la clasificación del nivel de crecimiento del arrozal. En el resultado de la máquina de vector de soporte se obtuvo una exactitud, relativamente baja, del 25 %.

Harish Babu Arunachalam, Rashika Mishra, Ovidiu Daescu, Kevin Cederberg, Dinesh Rakheja, Anita Sengupta, David Leonard y Rami Hallac, (2018), realizaron la investigación titulada “Viable and necrotic tumor assessment from whole slide images of osteosarcoma using machine-learning and deep-learning models”, en Estados Unidos. En esta investigación abordan el tema del uso de algoritmos de aprendizaje automático y aprendizaje profundo para la clasificación de imágenes de osteosarcoma, el cual es un tipo de cáncer. Para el dataset, se empleó imágenes digitalizadas de tejido de cincuenta pacientes tratados de osteosarcoma entre 1995 y 2015. De estas imágenes, dos patólogos, seleccionaron cuarenta imágenes, las cuales fueron recortadas en cuadrados de 1024x1024 píxeles, obteniendo alrededor de 1,200 imágenes, de estas se eliminaron cincuenta y seis, las cuales no aportaban ningún dato relevante para la investigación. Estas imágenes fueron usadas como input para el modelo de aprendizaje profundo, para ello se coloreo las partes de la imagen que podrían aportar datos necesarios para su evaluación por el algoritmo. En la clasificación se dividió en tumor viable (TV), no tumor (NT) y tumor necrótico (TN), siendo 536 (47 %), 263 (23 %) y 345 (30 %) su cantidad. Las características de las células del tejido que fueron evaluadas se dividen en propiedades de forma (área nuclear, redondez y perímetro) y densidad (conteo de núcleos). En cuanto a los métodos de aprendizaje automático utilizados, tenemos a los árboles de decisión (AD) y máquinas de vector de soporte (MSV), para realizar los experimentos, aprovecharon las herramientas disponibles en el paquete de aprendiz de clasificación de Matlab. También utilizaron aprendizaje profundo, en este caso, red neuronal convolucional (RNC), contando la arquitectura con tres

capas de convolución, tres capas de sub muestreo y dos perceptrones multi capa totalmente conectados. Para el procesado de las imágenes, se removió los parches con información poco relevante, transformando la imagen a binario. Para las métricas de desempeño se utilizó validación cruzada k-fold y exactitud, obteniendo como resultados promedio para TV 95.3%, para TN 92.7 % y para NT 91.9%, en el caso de redes neuronales convolucionales; para TV 77 %, para TN 75 % y para NT 87 %, en el caso de árboles complejos; para MSV los resultados fueron 91 % TV, 87 % TN y no tumor 91 %.

Eman N. Marzban, Ayman M. Eldeib, , Inas A. Yassine y Yasser M. Kadah, (2020), realizaron la investigación titulada "Alzheimer's disease diagnosis from diffusion tensor images using convolutional neural networks", en Estados Unidos. En ella abordan el tema de análisis de imágenes de resonancias cerebrales para diagnóstico de la enfermedad de Alzheimer con la ayuda de algoritmos de aprendizaje automático. Para este caso, utilizaron un dataset constituido por datos relativos a discapacidad cognitiva leve (DCL) y enfermedad de Alzheimer (EA) a través del control de la salud (CS) de 406 pacientes. La arquitectura de las RNC es de una capa de entrada para los datos de las imágenes, filtros de convolución, una capa de normalización por lotes, otra capa de agrupamiento máximo, una capa totalmente conectada y una capa para la clasificación. El desempeño fue evaluado con la ayuda de la sensibilidad, área bajo la curva, f1-score, exactitud y especificidad. La clasificación deberá darse entre la DCL y la EA, tomando en cuenta el CS. En el caso de DCL contra CS, se obtuvieron valores de 93.5 %, 93.9 % y 0.94 para la exactitud, especificidad y área bajo la curva respectivamente. Por otro lado, el caso de EA contra CS los valores fueron 81.8%, 71.1 % y 0.68 para la especificidad, la exactitud y área bajo la curva respectivamente.

Xiaowen Liang, Jinsui Yu, Jianyi Liao y Zhiyi Chen, (2020), realizaron la investigación de título ""Convolutional Neural Network for Breast and Thyroid Nodules Diagnosis in Ultrasound Imaging", en China. Su objetivo era el diagnóstico de nódulos en

seno y tiroides usando imágenes de ultrasonido con la ayuda de redes neuronales convolucionales, teniendo como dificultad el hecho que en comparación con la resonancia magnética y la tomografía computarizada, la calidad de una imagen de ultrasonido es muy influenciada gracias a muchos factores. Para el dataset, emplearon 537 imágenes provenientes de ultrasonidos de pacientes, las cuales estaban en formato Portable Networks Graphics (PNG). Las enfermedades se encontraban en dos grandes grupos: tumores de seno y nódulos tiroideos, los que a su vez se subdividían en fibro adenoma y carcinoma invasivo para el primero, y cotos nodulares y carcinoma papilar, para el segundo. Primero se pre procesaron las imágenes, para ello se recortaron, focalizando las áreas afectadas, en cuadrados de 315x315 de resolución. Para resaltar los límites de la lesión, se ayudaron de líneas rojas. Para el entrenamiento se fijó una tasa de aprendizaje del 0.001 y 200 periodos. Para evaluar el desempeño se usó la sensibilidad, la exactitud y especificidad. Finalmente se obtuvo como resultado que el desempeño de la RNC para los nódulos de seno y tiroides, comparado con el de expertos, fue superior, con 92 % para tiroides y 91 % para seno. También notaron que la falta de segmentación en el tejido puede causar un diagnóstico errado.

Lijun Luo, Yan Ren, Ling Gu, Tianjiao Wang, Fenglin Zhou, (2019), realizaron la investigación "Major crop pests identification research based on Convolutional Neural Network", en China. Su investigación se basó en la identificación de plagas en los variados cultivos presentes en la agricultura de China. Para combatir el problema de las plagas se suele utilizar diferentes tipos de pesticidas, lo cual tiene efectos colaterales que pueden afectar el ecosistema si son usados de forma indiscriminada, por lo que es necesario poder focalizar el área afectada. Por lo que optaron a encontrar una forma automatizada de resolver este problema, siendo que con los algoritmos de reconocimientos de patrones disponibles en las técnicas de aprendizaje automático, eligiendo a GoogLeNet como modelo y el dataset de entrenamiento ImageNet. Como resultados se logró la identificación de siete especies de animales causantes de plagas, siendo el promedio de este un 80%, pero la

excepción fue la especie *cnaphalocrocis medinalis* con la cual se obtuvo un 97% de precisión,

Y. Nagaraj, A Hema Sai Teja y A.V. Narasimhadhan, (2018), realizaron la investigación "Automatic Segmentation of Intima Media Complex in Carotid Ultrasound Images Using Support Vector Machine", en India. Ahí abordan el diagnóstico de enfermedades carótidas por medio de imágenes digitales de ultrasonido de la íntima y media carotídea con la ayuda de máquina de vector de soporte como modelo de clasificación. Un factor que podría influir de forma negativa en la correcta clasificación, es el hecho que, en las imágenes de ultrasonido de carótida, el ruido degrada la calidad de la imagen. El grosor de la íntima y la media (GIM) de la carótida es un medio para la detección de anomalías cardíacas, las cuales de no ser tratadas pueden tener un desenlace fatal. Para ello, en su método propuesto, uno de los puntos relevantes es la extracción de la "Región de Importancia" (RDI) en la detección de límites en un área de la imagen procesada, por lo que recurrieron a la librería de Matlab que incluye MSV para construir el modelo y obtener una predicción de los resultados. La función de base radial (FBR) se utiliza en la segmentación del GIM en las imágenes de entrada del dataset. Durante el proceso de aprendizaje el dataset fue dividido a la mitad entre datos de prueba, monitoreando el desempeño del modelo, y datos de entrenamiento. Como resultados obtuvieron un 92 % de precisión, usando como métrica de evaluación el coeficiente de correlación R, frente a un 90 % de redes neuronales, 89 % con programación dinámica y 85% con segmentación utilizando contornos activos.

T. Subetha, K. Rashmita y M. S. Christo (2020), realizaron la investigación "A comparative analysis on plant pathology classification using deep learning architecture – Resnet and VGG19", en India. Aquí los investigadores se clasificaron las imágenes de forma binaria, como sanos y enfermos. usando detección de bordes basado en imagen y técnicas de segmentación abocadas a la detección de bordes. para esta investigación de

diagnóstica temprano de enfermedades en hojas de manzanas, optaron por comparar dos algoritmos de aprendizaje profundo, estos son VGG19 y ResNet50. el dataset usado es Kaggle, teniendo como tarea clasificar entre saludable, con costras, enfermedades combinadas y moho. las imágenes están capturadas en ambientes no controlados, con variaciones de nivel de luz.

para el pre procesado de las imágenes, se optó por el tamaño de 224x224 pixeles, realizando rotaciones de las mismas por los ejes X e Y, también cambiando la saturación y el matiz. para el pre procesado de las imágenes, se optó por el tamaño de 224x224 pixeles, por ser esta la entrada máxima permitida para las arquitecturas seleccionadas, realizando rotaciones de las mismas por los ejes X e Y, también cambiando la saturación y el matiz. Finalmente obtuvieron una accuracy de entrenamiento y de validacion de 99.88 y de 96.17% de resnet50, el de VGG19 fue de 99.94 % y 96.72%, siendo este último mejor que el anterior.

G. Sachdeva, P. Singh, P. Kaur (2021), realizaron la investigación "Plant leaf disease classification using deep Convolutional neural network with Bayesian learning", en India. En su investigación, utilizaron una arquitectura de red neuronal convolucional, sumado al aprendizaje con el teorema de bayes, con eso los síntomas de las enfermedades presentes en las hojas lograron ser clasificadas. El teorema de bayes es usada en la capa superior de la red residual, todo ello con la finalidad de remarcar la dependencia entre los píeles. La información utilizada se trató de imágenes de hojas provenientes del dataset PlantVillage, el cual contiene 20.639 imágenes en la fecha de la realización de dicha investigación, siendo divididas en quince diferentes clases, comprendiendo imágenes digitalizadas de hojas sanas y enfermas de pimiento, papa y tomate. Se comparó redes convolucionales usadas en trabajos previos con la arquitectura de red convolucional propuesta, en la cual utilizaron cuatro capas de poolig, convolucion, dense y finalmente ReLU. seguido de la obtención de las características, aprendizaje bayesiano. Finalmente están la capa de quince clases entre imágenes de hojas sanas y enfermas. Se utilizó un padding para mantener el tamaño de la

imagen con cada convolucion. la entrada de la misma fue de 64x64x3, siendo las dos primeras las dimensiones de las imágenes y la tercera los valores de rojo, verde y azul, al ser el formato RGB. el padding fue de dos limites de ceros. la capa de convolución aplicó un kernel de 5x5x3..usando los parámetros de exactitud, precisión, F-score. Se entrenó con lotes de 16, tasa de dropout de 20% y cincuenta épocas. con aprendizaje bayesiano, obteniendo como resultado en la precisión de 98.99%.

A.Umit, U. Murat, A. Kemal, U. Emine (2020), realizaron la investigación “Plant leaf disease classification using EfficientNet deep learning model”, en Turquía. en la investigación utilizaron el dataset disponible en kaggle llamado PlantVillage, el cual consiste en 54305 imágenes, divididas en 38 clases, pertenecientes a 14 plantas distintas. de estas clases, 26 corresponden a plantas enfermas y 12 a sanas. con la finalidad de incrementar de forma artificial la cantidad de muestras disponibles, aplicaron seis métodos de Data Augmentation, ascendiendo la cantidad a 61486 imágenes. en este caso particular, se dividió para el entrenamiento un 90%, 7% para la validación y finalmente un 3% del total de las imágenes disponibles para las pruebas del modelo. en la ultima capa se cambio de 1000 clases a 39, para que coincidan con las clases del dataset, todas las capas se ajustaron para que sean entrenadas. en el caso de la activación se utilizó la función softmax. la función de costo fue de entropia categorica cruzada, un patience de 5 y un cambio minimo en la perdida de $1e-3$. en el caso de VGG61 utiliza el metodo de activación SGD, los demas la optimizacion Adam. la tasa de aprendizaje fue puesta en 0.001 y 0.01 para Adma y para SGD respectivamente. los pasos de validacion fue puesta en 1. Finalmente obteniendo un 98.31 % de precisión con el dataset sin alterar y un 99.38% de precisión utilizando data augmentation.

Gran parte de la agricultura en el Perú, se da forma tradicional teniendo cierta resistencia a la integración de soluciones de tecnologías de información (TI). Esto limita el aprovechamiento de los datos que constantemente se reciben por no darle la relevancia

adecuada. Otro factor sería la limitada cantidad de herramientas que utilicen TI, no habiendo en algunos casos la “solución” a problemas específicos. Este proyecto de investigación tuvo como objetivo implementar una herramienta que permita la clasificación de enfermedades de caña de azúcar con bajo costo computacional, un cultivo ampliamente difundido en el territorio peruano, siendo la industria azucarera un factor importante en la economía local y nacional. Las enfermedades existentes en los cultivos de caña de azúcar, puede generar grandes pérdidas económicas, muchas de ellas teniendo como consecuencia la muerte de la planta y la rápida propagación del agente infeccioso por todo el cultivo, no habiéndose encontrado literatura similar al problema que se aborda.

1.2 Formulación del problema

¿Cómo identificar de manera automática las enfermedades en cultivos de caña de azúcar basados en síntomas de las hojas?

1.3 Hipótesis

Mediante el uso de técnicas de clasificación de aprendizaje profundo, utilizando para ello imágenes digitales de las hojas de caña de azúcar, se podrá detectar y clasificar las enfermedades que les afecten.

1.4 Objetivos

Objetivo general.

Comparar redes neuronales convolucionales para identificar enfermedades en cultivos de caña de azúcar mediante imágenes digitales de las hojas y encontrar la más eficiente.

Objetivos específicos.

- Elaborar el dataset con imágenes digitales de las hojas de caña de azúcar.
- Pre procesar las imágenes digitales de las hojas de caña de azúcar.
- Implementar redes neuronales convolucionales estándares y personalizada para el procesamiento de las imágenes del dataset.
- Realizar las pruebas y generar resultados

1.5 Teorías relacionadas al tema

1.5.1 Aprendizaje automático.

Es la investigación de algoritmos computacionales que tienen la particularidad de mejorar su desempeño a través de la experiencia. Está fuertemente ligado a la computación estadística, en la cual se realizan predicciones con la ayuda de computadoras. También está emparentado con la minería de datos cuyo objetivo es la detección de características ignotas en grandes volúmenes de datos, utilizando para ello aprendizaje no supervisado. [5]

1.5.2 Inteligencia artificial.

Es un campo de la ciencia de datos, en la cual, las máquinas muestran un comportamiento inteligente de forma artificial, pudiendo ejecutar tareas complejas, imitando las capacidades cognitivas de los seres humanos, como la absolución de problemas y el aprendizaje a través de la experiencia. En este se investiga a entes inteligentes, los cuales perciben el ambiente que los rodean y ejecutan tareas con la finalidad de alcanzar ciertas metas requeridas. La inteligencia artificial (IA), es utilizada en diversas áreas de las actividades humanas, como son salud, deporte, milicia, economía, video juegos, auditoria, arte, etc. [6]

1.5.3 Aprendizaje profundo.

Es parte del amplio repertorio de métodos de aprendizaje automático, en el cual el término profundo se debe al hecho de utilizar múltiples capas en la red. Surgió como resultado de las limitaciones del perceptrón lineal como clasificador universal. [7]

El aprendizaje profundo o deep learning está contenido dentro del campo del aprendizaje automático. En este se busca desarrollar modelos que puedan representar el mundo que los rodea, a través de la interacción, logrando obtener una versión simplificada y fácil de entender de la información. [8]

En este tipo de aprendizaje, cada nivel convierte los datos de entrada en información cada vez más abstracta conforme se transforma. Este puede ser utilizado tanto para el aprendizaje supervisado como para el no supervisado. [9]

1.5.4 Red neuronal artificial.

Es un modelo matemático con características similares a algunas halladas en la función cerebral. En estas se busca simular sistema nervioso de los organismos, pero hay una discusión sobre si en realidad es una simulación del sistema nervioso de los organismos vivos.

Las neuronas artificiales tienen (nodos) que forman una red por acoplamiento sináptico cambiando los pesos de la unión a sinapsis a través del aprendizaje, dándole la capacidad de resolver problemas. Las neuronas artificiales tienen un comportamiento similar a una forma simplificada de las neuronas de organismos vivos. Estas son comúnmente usadas en minería de datos, clasificación de imágenes, detección de patrones, entre otros. [10]

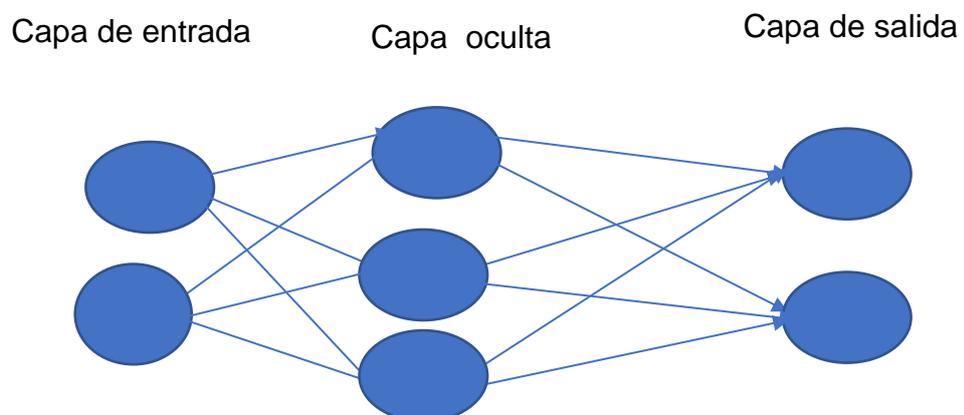
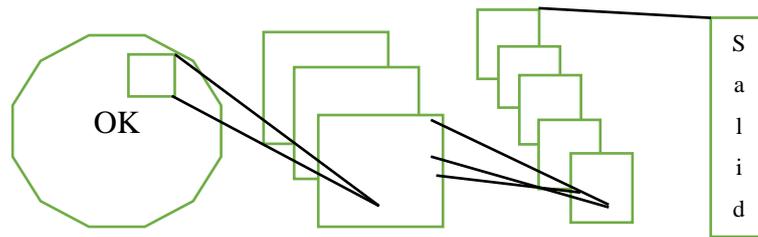


Figura 1. Representación de una red neuronal.

Fuente: Elaboración propia

1.5.5 Red neuronal convolucional

Este es un tipo de red neuronal profunda artificial, propagada directamente. Es un modelo utilizado para el reconocimiento de vídeos e imágenes. Utiliza una variación de perceptrón multicapa y está diseñado para requerir un mínimo pre procesamiento de datos. La forma en que están unidas las neuronas en la red de convolución se basan en la corteza visual de los animales. Los campos receptivos de diferentes neuronas están superpuestos para poder cubrir el campo de visión en su totalidad. Realiza un pre procesamiento pequeño si es comparado con otros algoritmos de clasificación por imagen. Es aplicada al procesamiento del lenguaje natural, reconocimiento de video e imagen y al sistema de recomendación. [11]



Tamaño del filtro = $f^{[l]}$	$entrada = n_h^{[l-1]} \times n_n^{[l-1]} \times n_c^{[l-1]}$
Padding = $p^{[l]}$	$cada\ filtro = f^{[l]} \times f^{[l]} \times n_c^{[l-1]}$
Paso = $S^{[l]}$	$pesos = f^{[l]} \times f^{[l]} \times n_c^{[l-1]} \times n_c^{[l]}$
	$salida = n_h^{[l]} \times n_w^{[l]} \times n_c^{[l]}$
	$sesgo = 1 \times 1 \times 1 \times n_c^{[l]}$

Figura 2. Representación de una red neuronal convolucional.

Fuente: *Elaboración propia basado en [11]*

1.5.6 Capa de Pooling.

Dentro de las redes neuronales convolucionales, esta el concepto de “capas de pooling”, que es básicamente como tomar una muestra de la imagen que ha pasado por una capa convolucional anterior. Para obtener esta “muestra” se puede emplear el Pooling promedio, tomando valores medios entre los elementos que encierra el filtro de pooling; por otro lado, también esta el Max Pooling, el cual solo toma los valores máximos dentro del filtro de pooling. El objetivo de esta capa es disminuir la resolución de la imagen, pero no por ello perdiendo las características de la misma, reduciendo el costo computacional de la red neuronal. Después de cada convolucion, la imagen se reduce con una capa de pooling, en la cual se pasa de tener detalles de la imagen cada vez más pequeños a más grandes y generalizados.

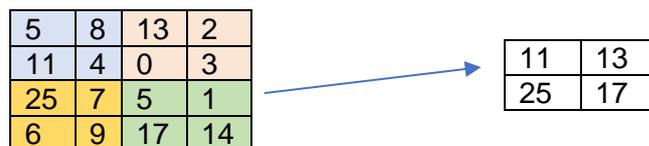


Figura 3. Representación del uso de un filtro de Max Pooling 2x2. Fuente: Elaboración propia



Figura 4. Representación del uso de un filtro de Average Pooling 2x2. Fuente: Elaboración propia

1.5.7 Flattening.

El flattening (aplanamiento) consiste en convertir la información recibida de una anterior capa en un vector de una dimensión. Está conectado con la capa final del modelo.



Figura 5. Flattening de una imagen de dimensiones 2x2 . Fuente: Elaboración propia

1.5.8 Aprendizaje supervisado.

En este tipo, la labor de aprendizaje se da utilizando ejemplos que servirán de base, siendo llamados “datos de entrenamiento”, logrando una generalización que utilizara cuando enfrente datos que anteriormente no había recibido. Se debe considerar las siguientes tareas al momento de tratar de resolver un problema que requiera este tipo de aprendizaje: evaluar qué tipo de ejemplos serán usados para el entrenamiento, luego se deberá reunir los datos necesitados, después se debe considerar la función de aprendizaje que según la entrada que se le dé, podrá obtener la salida esperada, luego la arquitectura de la función de aprendizaje, finalmente evaluar el desempeño de la solución propuesta. [12]

1.5.9 Agrupamiento.

Es un tipo de método de análisis de datos y también un algoritmo de machine learning no supervisado que clasifica datos, también se dice que es un método para clasificar automáticamente determinados datos sin criterios externos. [13]

Existen métodos en los que se clasifica datos jerárquicamente y métodos no jerarquizados para clasificar los datos en un número específico de agrupadores. [14]

1.5.10 Support vector machine.

En español máquina de vector de soporte, para este algoritmo se busca un hiper plano que pueda clasificar datos con características similares en un espacio de N-dimensiones. Esta separación se da a través de la selección de un plano con margen máximo. Este un modelo de aprendizaje que cuenta con un gran rendimiento de reconocimiento si lo comparamos con las técnicas existentes. Esta presenta una eficiente

taza de reconocimiento debido al alto rendimiento de identificación para los datos no aprendidos. [15]

1.5.11 Regresión lineal.

En un concepto de la estadística, posibilita el modelar la relación entre variables dependientes y variables independientes. El modelado es mediante funciones de predicción lineales con parámetros inciertos, los cuales son estimados utilizando los datos. La regresión se centra en la distribución de probabilidad condicional de la respuesta con un valor previsto. [16]

1.5.12 Regresión logística.

Es un tipo de modelo de regresión estadística, siendo este un modelo lineal generalizado (MLG). Es una regresión de probabilidad y se utiliza principalmente para la clasificación de estadísticas. La regresión lineal es el primer tipo de análisis de regresión rigurosamente estudiado y es ampliamente utilizado en aplicaciones prácticas. El modelo que depende de la alineación con parámetros desconocidos es más fácil de ajustar que el modelo que no está linealmente relacionado con el parámetro, porque es fácil determinar las características estadísticas del valor estimado del resultado. [17]

1.5.13 Naive bayes.

Es un conjunto de clasificadores basados en el concepto del teorema de Bayes con uso de grandes suposiciones de independencia entre funciones. Sin embargo, si combina con la estimación de la densidad de kernel, es posible lograr un mayor nivel de precisión. El clasificador de Naive Bayes es muy escalable y requiere muchos parámetros para el número de variables (características/predictores) de problemas de aprendizaje. Se puede realizar un entrenamiento de máxima similitud evaluando expresiones cerradas, tardando un tiempo lineal en lugar de aproximaciones iterativas caras utilizadas en otras clases de clasificadores. [18]

El modelo de Naive Bayes posee muchos nombres, como Bayes simple y Bayes Independiente, por mencionar algunos. Todos se basan en el uso de las reglas de decisión del clasificador, presentes en el teorema de Bayes. [5]

1.5.14 Análisis de la discriminante lineal.

Es una técnica utilizada en aprendizaje automático, estadísticas y reconocimiento de patrones para encontrar una separación en objetos dos o más clases usando combinaciones lineales de entidades. El resultado de la combinación puede ser utilizado para la reducción dimensional. Está estrechamente relacionado con el análisis de regresión, representando una variable dependiente como una combinación lineal de otras medidas o características. La regresión logística presenta variables de categoría por los valores de variables independientes. [19]

Está relacionado con el análisis de las partes esenciales y el análisis de factores, que buscan configuraciones lineales de variables, describiendo los datos de forma más eficiente. Este modela las particularidades entre clases de datos. Es diferente al análisis de factores, ya que no se vale de interdependencias, se debe diferenciar las variables independientes y dependientes. Funciona cuando se toman mediciones para las variables independientes para toda observación son cantidades consecutivas. El método equivalente es, cuando son variables independientes, un análisis que reconozca la diferenciación.

1.5.15 K-nearest neighbours.

es un tipo de aprendizaje fundamentado en instancias, donde las funciones se aproximan de manera local y siendo el cálculo aplazado hasta que la función sea evaluada. Puede mejorarse la precisión al normalizar la información de entrenamiento, por el hecho que este algoritmo, para la clasificación, emplea la distancia. La ponderación a las contribuciones vecinas otorga un enfoque favorable, aportando al promedio mucho más que los vecindarios lejanos. Se puede representar como un conjunto de entrenamiento para un algoritmo, pero no necesita de pasos de entrenamiento explícitos. [20]

1.5.16 Aprendizaje no supervisado.

Es un tipo de método de análisis de datos y también un algoritmo de machine learning no supervisado que clasifica datos, también se dice que es un método para clasificar automáticamente determinados datos sin criterios externos.

Existen métodos en los que se clasifica datos jerárquicamente y métodos no jerarquizados para clasificar los datos en un número específico de agrupadores. [14]

1.5.17 Procesamiento de imágenes digitales.

Es la utilización de algoritmos para procesar imágenes, esto permite disminuir características no deseadas como ruido, desenfoque o distorsión. Esto es utilizado en tecnologías de aprendizaje automático como en reconocimiento de patrones, análisis de señales, clasificación, redes neuronales, extracción de características, etc. También es utilizado en dispositivos electrónicos como cámaras digitales, permitiendo obtener mejores fotografías o videos al utilizar hardware y software dedicado. [21]

1.5.18 Segmentación.

Es la separación de áreas de interés en una imagen, teniendo para ello que discriminar los componentes cuyas características no correspondan con las buscadas. La segmentación se puede realizar separando regiones con formas aparentemente relevantes, únicamente separando una figura en específico del fondo o agrupando utilizando algo similar a “súper píxeles” regiones vecinas que sean muy similares. [22]

1.5.19 Reconocimiento de patrones.

Es la ejecución de un proceso en el cual se utiliza como insumo información obtenida a través de los sentidos para encontrar similitudes con lo que se percibe en el ambiente. En el área del aprendizaje automático se utiliza un dataset de entrenamiento, que le servirá al algoritmo como guía en la búsqueda de similitudes con la información nueva

obtenida. Para esta tarea es muy común el uso de inferencias probabilística, etiquetando con mayor precisión a una etiqueta dada.

1.5.20 Data Augmentation.

Se utiliza este término al referirse a la creación nuevos datos a partir de los datos que ya se tienen, realizando para ello algunas modificaciones, como por ejemplo voltear una imagen en su eje vertical o en su eje horizontal, rotándola en una cierta dirección y ángulo, realizar un zoom, etc. Al incrementar la cantidad de datos en el entrenamiento del modelo se puede evitar el sobre entrenamiento. Como ejemplo se puede tomar las imágenes de rostros humanos, las cuales si el dataset está compuesto enteramente de rostros mirando a una sola dirección, al momento de determinar si una imagen de rostro que mira en la dirección contraria entra o no en la clasificación de “rostro humano”, termine cayendo en la categoría de falso negativo. [23]

1.5.21 Estandarización.

Consiste en escalar un valor basado en la distribución normal estándar, a continuación su fórmula:

$$z = \frac{X - \mu}{\sigma}$$

Donde:

X = valor a estandarizar.

μ = Promedio de los valores dividido entre la desviación estándar.

σ = Desviación estándar.

1.5.22 Normalización.

Es escalar un valor entre 0 y 1, la fórmula es:

$$X_{normalizado} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Donde:

X = valor a normalizar.

Xmax = el máximo valor dentro del conjunto de datos disponibles.

Xmin = el mínimo valor dentro del conjunto de datos disponibles.

En el caso de imágenes digitales, los valores de los píxeles van de 0 a 255 píxeles, teniendo como resultado al aplicar la fórmula, el valor del píxel dividido entre 255.

1.5.23 Detección de anomalías.

Usado ampliamente en minería de datos, procesamiento de señales, finanzas, estadísticas, etc. Se encarga de identificar en un conjunto de elementos que no encajan con las características presentes en la mayoría. Las anomalías son patrones que no se ajustan a lo que está definido de forma concisa como normal. Estas anomalías pueden también interpretarse como problemas como comportamiento errado, fraude financiero, defectos estructurales, problemas médicos, etc. En el aprovechamiento de vulnerabilidades, estas también se presentan como actividad inusual súbita. [24]

1.5.24 Python.

Es un lenguaje de programación diseñado por Guido Van Rosum, quien tuvo como lineamiento que sea fácilmente entendible y sencillo de aprender. Es popularmente usado en Ciencia de datos por tener librerías que permiten fácilmente ejecutar programas basados en análisis matemático, estadística, algebra; además contando con librerías para mostrar los resultados obtenidos mediante gráficas. [25]

1.5.25 Numpy.

Esta librería de Python está especializada para permitir la óptima manipulación de objetos representados en arrays o arreglos multidimensionales. Ahora, estos arrays no funcionan del mismo modo que las listas. [26]

1.5.26 Matplotlib.

Es una librería de Python la misma que permite realizar gráficos, lo que facilita, en el caso particular de la investigación realizada; observar el comportamiento de las métricas de los modelos utilizados. Entre los gráficos que pueden ser usados están los de líneas, gráficos de barras, piecharts o gráfico de torta, boxplots (representación de datos atípicos) y gráficos de dispersión. [25]

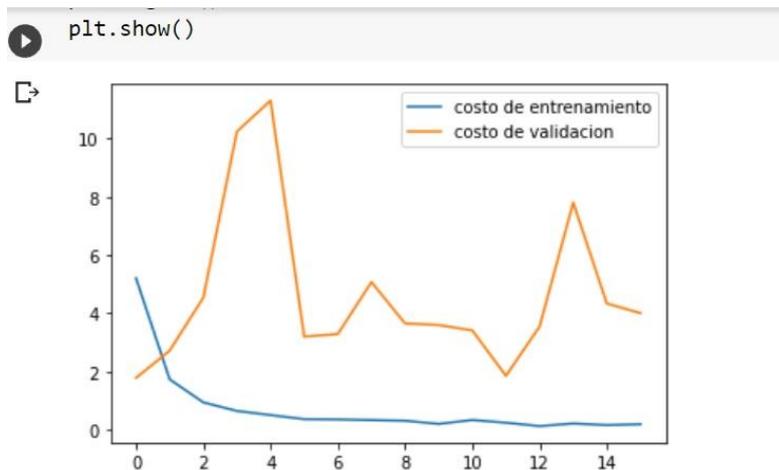


Figura 6. Ejemplo de un gráfico de líneas realizado en google collab, utilizando la librería matplotlib.

Fuente: Elaboración propia

1.5.27 Tensorflow.

Está dentro de las tecnologías disponibles de código abierto. Es una plataforma que se enfoca en el aprendizaje automático, teniendo un ambiente y herramientas que facilitan la tarea de realizar aplicaciones de este campo. Dentro de las funcionalidades que pueden ser usadas están la creación, despliegue entrenamiento y evaluación de modelos

1.5.28 Caña de azúcar.

Es una planta alta emparentada con el césped, cuyo nombre científico es *Saccharum Officinarum*, la cual se utiliza en la industria para la obtención de azúcar, etanol,

panela e incluso el bagazo es utilizado para la producción de papel. La costa norte de Perú es una zona que cuenta con las condiciones climáticas ideales para su siembra, encontrándose en varias provincias de esta área en particular, fábricas, conocidas también como ingenios, que se encargan de obtener productos derivados a partir de su procesamiento. [27]



Figura 7. Planta de caña de azúcar.

Fuente: *(Food Bussines Food, 2019)*

1.5.29 Xanthomonas albilineans.

Es una proteobacteria que suele infectar a una gran variedad de cultivos. Para el caso particular de la caña de azúcar, los síntomas visibles en las hojas son franjas de color amarillo. Este agente patógeno en su fase más avanzada causa necrosis en las hojas, causando como ultima instancia, la muerte de la planta. El tiempo que toma desde la infección no es fijo, puede ser desde semanas a meses, esto es debido a que algunas variaciones de caña de azúcar tienen resistencia a esta bacteria [28]



Figura 8. Xanthomonas albilineans en hoja de caña de azúcar.

Fuente: Elaboración propia

1.5.30 Acidovorax Avenae.

También es conocida en algunos países como *Acidovorax Citrulli*, es una bacteria. Un sintoma visible de cuando infecta a la caña de azúcar, es la aparición de líneas rojas en las hojas y al interior del tallo, siendo el color mas intenso en fases muy avanzadas, causando necrosis en el tejido estructural de la planta y su posterior muerte. [29]

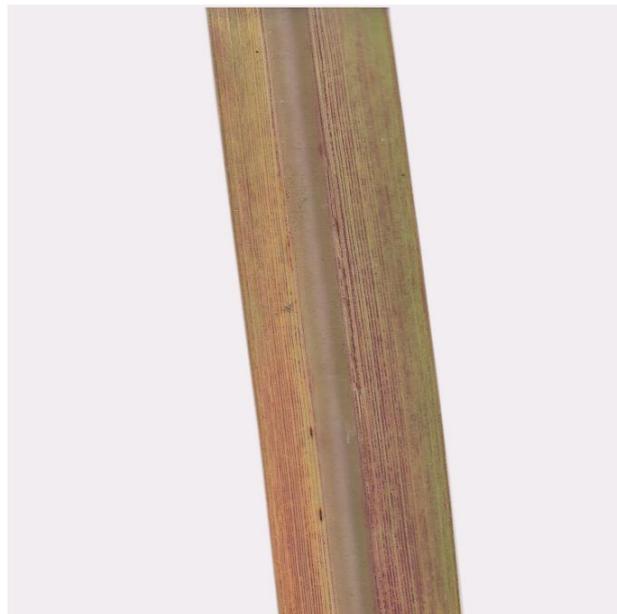


Figura 9. Acidovorax Avenae en hoja de caña de azúcar.

Fuente: Elaboración propia

1.5.31 *Bipolaris sacchari*.

Es un hongo, el cual infecta mayormente a la caña de azúcar, esta enfermedad se produce por el contacto de la planta con esporas, siendo el síntoma más observable la aparición de manchas de centro color rojo en las hojas. Es conocida de forma coloquial como “mancha de ojo”. [1]



Figura 10. Hoja infectada con Bipolaris Sacchari.

Fuente: Elaboración propia

1.5.32 Partes de la caña de azúcar.

Hojas

Las hojas surgen a partir de los nodos presentes en el tallo, estando estas dispuestas en pares opuestos. Teniendo en un inicio una orientación hacia el firmamento, la cual va decayendo conforme la planta madura, adquiriendo la característica forma de arco. [30]



Figura 11. Hoja sana de caña de azúcar.

Fuente: Elaboración propia

Canutos

Son segmentos que conforman el tallo, los cuales son en un inicio de base gruesa y extremo superior delgado. Conforme la planta va creciendo, nuevos canutos se van agregando, quedando los más antiguos con un grosos similar en ambos extremos.

Rizoma.

Se encuentra en la base del tallo, siendo sus funciones la de proveer soporte a la planta y la de producir raíces.

II. MATERIALES Y MÉTODO

1.1 Tipo y Diseño de Investigación.

Es un tipo de investigación cuantitativa, porque se obtendrá resultados numéricos medibles, los cuales sirven como indicadores del desempeño del algoritmo.

El diseño de la investigación es cuasi experimental, en esta se comparará entre 3 algoritmos ya existentes con uno propuesto por el investigador.

1.2 Variables, Operacionalización

1.2.1 Variable dependiente.

Clasificación de enfermedades de caña de azúcar.

1.2.2 Variable independiente.

Algoritmos de clasificación de aprendizaje automático.

1.2.3 Operacionalización de Variables.

Tabla 1

Operacionalización de variables

Variable	Indicador	Ítem	Técnica e instrumentos de recolección de datos
Variable Dependiente: Clasificación de enfermedades de caña de azúcar.	Precisión,	$Precisión = \frac{VP}{VP + FN}$	Registro en medios electrónicos
	Recall	$Recall = \frac{VP}{VP + FN}$	
	F1-Score	$F1\ score = \frac{2}{recall^{-1} + precision^{-1}}$ $= 2 \cdot \frac{precision \cdot recall}{precision + recall}$	
	Exactitud	$exactitud = \frac{VN + VP}{VP + FP + VN + FN}$	

Fuente: Elaboración propia

Variable	Indicador	Ítem	Técnica e instrumentos de recolección de datos
Variable Independiente:	Tiempo de Respuesta	$TR = TF - TI$	Registro en medios electrónicos
Algoritmos de clasificación de aprendizaje automático	Consumo de GPU	$GPU = GPU_f - GPU_0$	
	Consumo de RAM	$CRAM = \frac{TM - UM}{TM}$	

Fuente: Elaboración propia

1.3. Población y muestra.

La población está conformada por los algoritmos de aprendizaje automático de clasificación, en este caso Naive Bayes, máquina de soporte vectorial, la discriminante lineal de Fisher, redes neuronales convolucionales, bosque aleatorio, árbol de decisión, regresión logística, vecinos cercanos K y red neuronal artificial y para la muestra se eligió a Naive Bayes, redes neuronales convolucionales, vecinos cercanos K y máquina de vector de soporte.

La muestra comprende redes neuronales convolucionales MobileNet, VGGNET16, ResNet50 seleccionadas por conveniencia al haber obtenido buenos resultados en anteriores investigaciones, además de un modelo propio propuesto.

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

1.4.1 Observación.

La observación permite recopilar datos sin interactuar directamente con el objeto de estudio, evitando la modificación artificial de la información obtenida. Esta información se puede almacenar tanto en medios digitales como en físicos. Con la información obtenida se tiene un panorama más claro de la situación problemática.

Para la validez y confiabilidad de la información obtenida, se utilizará indicadores de desempeño, en este caso la precisión, la exactitud, recall, F1-score y matriz de confusión.

1.4.2 Instrumentos

Para este trabajo de investigación, en el cual uno de los principales instrumentos es un ordenador, siendo este mecánico y electrónico, en el cual se obtendrán resultados del tipo positivo y negativo, se consideró el uso de una matriz de confusión donde se plasmó estos resultados, observando en cuales casos se dieron Verdaderos Positivos (VP), Verdaderos Negativos (VN), Falsos Positivos (FP) y Falsos Negativos (FN).

Tabla 3.

Ejemplo de matriz de confusión

Población total	Predicción Positiva	Predicción Negativa
	Clase A (Positiva)	Verdadero Positivo (VP)
Clase B (Negativa)	Falso Negativo (FN)	Verdadero Negativo (VN)

Fuente: [31]

1.5 Procedimiento de análisis de datos.

En esta tarea se utilizó los estos indicadores:

1. Consumo de GPU

Con esta métrica medimos que tanto tiempo tomó procesar la ejecución de los algoritmos a evaluar.

2. Consumo de RAM

La memoria RAM se encarga de guardar de forma temporal la información necesaria para la ejecución de procesos que se están ejecutando, permitiendo que se ejecuten de forma rápida.

3. Tiempo de respuesta

Mide el intervalo de tiempo en la ejecución de un proceso.

4. Precisión

Con la precisión se observa la proporción entre los resultados positivos obtenidos y si fueron correctos.

$$\text{Precisión} = \frac{VP}{VP + FP}$$

Donde:

VP= Verdaderos positivos

FP= Falsos positivos

5. Recall

También conocido como exhaustividad, con esta se mide la proporción entre los verdaderos positivos y la suma de los falsos negativos con los verdaderos positivos o tasa de verdaderos positivos.

$$\text{Recall} = \frac{VP}{VP + FN}$$

6. Exactitud

Con este indicador obtenemos la tasa de verdaderos negativos, siendo este indicador de la proporción de los negativos evaluados de forma correcta.

$$\text{Exactitud} = \frac{VN}{VN + FP}$$

7. Accuracy

Con este indicador obtenemos la proporción entre los resultados correctamente clasificados y el total de resultados proporcionados obtenidos.

$$\text{Accuracy} = \frac{VN + VP}{VP + FP + VN + FN}$$

8. F1-Score

Con este indicador obtenemos la media armónica de la precisión y la exhaustividad o recall,

$$F1\ score = \frac{2}{recall^{-1} + precision^{-1}} = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

1.6. Criterios éticos.

Respeto a la propiedad intelectual:

No se atribuirá información proveniente de aportes científicos como propios. La propiedad intelectual tiene como fin, proteger los bienes inmateriales creados a partir de la reflexión humana, esta protección motiva a la creación de nuevo conocimiento el cual podrá ser utilizado para la obtención de más conocimiento.

Confidencialidad de la información obtenida:

La información que involucre a terceros será tratada garantizando el total anonimato, evitando cualquier tipo de infidencia.

I.2. Criterios de Rigor Científico.

Replicación:

Los procedimientos realizados en esta investigación serán detallados de tal forma que podrá ser utilizada y replicada por terceros para nuevas investigaciones.

Confiabilidad:

Los resultados obtenidos serán consistentes y confiables, por lo que, si se siguen las condiciones planteadas en este documento, se obtendrán resultados muy cercanos a los expuestos.

Validez:

La información obtenida tendrá conclusiones obtenidas a través de la investigación realizada, procurando que las mediciones de las variables sean las más precisas.

III. RESULTADOS.

2.1 Resultados en Tablas y Figuras.

En la investigación realizada, se comparó la eficiencia en la clasificación de enfermedades presentes en las hojas de la caña de azúcar, siendo estas *Acidovorax Avenae*, *Bipolaris Sacchari* y *Xanthonomas Albillinealis*, incluyendo la comparación con ejemplares sanos; de cuatro redes neuronales convolucionales, tres ampliamente usadas en investigaciones relacionadas a visión por computar, siendo estas MobileNet1, VGG16, RESNET50; además de una red neuronal convolucional propuesta por el investigador. Por eso se utilizaron métricas, las cuales están especificadas en el punto “2.5 Procedimiento de análisis de datos”.

Para todas las redes se fijo un número de epochs de 140 y lotes de 4 . Siendo los resultados los siguientes para cada arquitectura:

Resultados obtenidos al ejecutar la red neuronal convolucional propuesta

Tabla 4

Matriz de confusión contenida de resultados de validación en red propuesta

Síntoma visible	A. Avenae	B. Sacchari	Hoja Sana	X. Albillineans
A. Avenae	44	3	8	5
B. Sacchari	12	25	2	21
Hoja Sana	2	1	46	11
X. Albillineans	0	0	3	57

Fuente: Elaboración propia

Tabla 5

Indicadores de desempeño de resultados de validación en red propuesta

Indicadores	B. Avenae	B. Sacchari	Hoja Sana	X. Albillineans
F1 - Score	0.7458	0.5618	0.7731	0.7403
Precisión	0.7333	0.4167	0.7667	0.9500
Exactitud	0.8515	0.8152	0.8643	0.8113
Recall	0.7586	0.8621	0.7797	0.6064
Exactitud	0.8889	0.8077	0.9000	0.9746
Tiempo de respuesta total				1468 s
RAM usada				1.75 GB
GPU usada				1.77 GB

Fuente: Elaboración propia

Resultados obtenidos al ejecutar la red neuronal convolucional MobileNet1.

Tabla 6

Matriz de confusión contenida de resultados de validación en red MobileNet1

Síntoma visible	A. Avenae	B. Sacchari	Hoja Sana	X. Albillineans
A. Avenae	45	4	2	9
B. Sacchari	8	28	0	24
Hoja Sana	8	0	42	10
X. Albillineans	2	4	0	54

Fuente: Elaboración propia

Tabla 7

Indicadores de desempeño de resultados de validación en red MobileNet1

Indicadores	B. Avenae	B. Sacchari	Hoja Sana	X. Albillineans
F1 - Score	0.7317	0.5833	0.8077	0.6879
Precisión	0.7500	0.4667	0.7000	0.9000
Exactitud	0.8366	0.8086	0.8942	0.7752
Recall	0.7143	0.7778	0.9545	0.5567
Exactitud	0.8921	0.8150	0.8759	0.9504
Tiempo de respuesta total				1151 s
RAM usada				1.92 GB
GPU usada				7.88 GB

Fuente: Elaboración propia

Resultados obtenidos al ejecutar la red neuronal convolucional ResNet50.

Tabla 8

Matriz de confusión contenida de resultados de validación en red ResNet50

Síntoma visible	A. Avenae	B. Sacchari	Hoja Sana	X. Albillineans
A. Avenae	50	4	2	4
B. Sacchari	4	42	2	12
Hoja Sana	1	0	56	3
X. Albillineans	0	5	3	52

Fuente: Elaboración propia

Tabla 9

Indicadores de desempeño de resultados de validación en red ResNet50

Indicadores	B. Avenae	B. Sacchari	Hoja Sana	X. Albillineans
F1 - Score	0.8696	0.7568	0.9106	0.7939
Precisión	0.8333	0.7000	0.9333	0.8667
Exactitud	0.9302	0.8811	0.9479	0.8811
Recall	0.9091	0.8235	0.8889	0.7324
Exactitud	0.9375	0.8977	0.9730	0.9487
Tiempo de respuesta total				1579 s
RAM usada				1.92 GB
GPU usada				7.51 GB

Fuente: Elaboración propia

Resultados obtenidos al ejecutar la red neuronal convolucional VGG16.

Tabla 10

Matriz de confusión contenida de resultados de validación en red VGG16.

Síntoma visible	A. Avenae	B. Sacchari	Hoja Sana	X. Albillineans
A. Avenae	53	6	0	1
B. Sacchari	6	47	3	4
Hoja Sana	2	5	51	2
X. Albillineans	7	13	5	35

Fuente: Elaboración propia

Tabla 11

Indicadores de desempeño de resultados de validación en red VGG16

Indicadores	B. Avenae	B. Sacchari	Hoja Sana	X. Albillineans
F1 - Score	0.8281	0.7176	0.8571	0.6863
Precisión	0.8833	0.7833	0.8500	0.5833
Exactitud	0.8942	0.8341	0.9163	0.8532
Recall	0.7794	0.6620	0.8644	0.8333
Exactitud	0.9500	0.9145	0.9375	0.8580
Tiempo de respuesta total				2755 s.
RAM usada				3.92 GB
GPU usada				5.26 GB

Fuente: Elaboración propia

2.2 Discusión de resultados.

Utilizando las técnicas de redes convolucionales, se realizó pruebas con los cuatro algoritmos utilizados, obteniendo los siguientes resultados:

Tabla 12.

Resultados de algoritmos

Red	Precisión	Accuracy	Tiempo de respuesta	Uso de GPU	USO de RAM
CNN Propuesta	71.67%	89.28%	1468 s	1.77 GB	1.75 GB
ResNet50	83.33%	93.92%	1579 s	7.51 GB	1.92 GB
VGG16	77.51%	91.51%	2755 s	5.26 GB	3.92 GB
MobileNetV1	70.41%	88.34%	1151 s	7.88 GB	1.92 GB

Fuente: Elaboración propia

Como se puede observar, en las cuatro redes se obtuvo porcentajes de Exactitud por encima del 88%, siendo el más alto el obtenido de la red ResNet50, luego VGG16, en tercer lugar la red neuronal convolucional propuesta y al final esta MobileNetV1. Ahora, teniendo en consideración el uso de recursos computacionales, la red propuesta es la más eficiente de las cuatro, además que el tiempo de respuesta es el segundo mejor de todas.

Este resultado indica que la red convolucional propuesta, puede usarse en ordenadores con especificaciones austeras, con una Exactitud del 89.28 %, es decir, de 100 hojas evaluadas un aproximado de 11 no serían clasificadas de forma correcta.

2.3. Aporte práctico.

Como guía visual del método utilizado en la resolución del problema de investigación, se elaboró un diagrama como indica la figura :

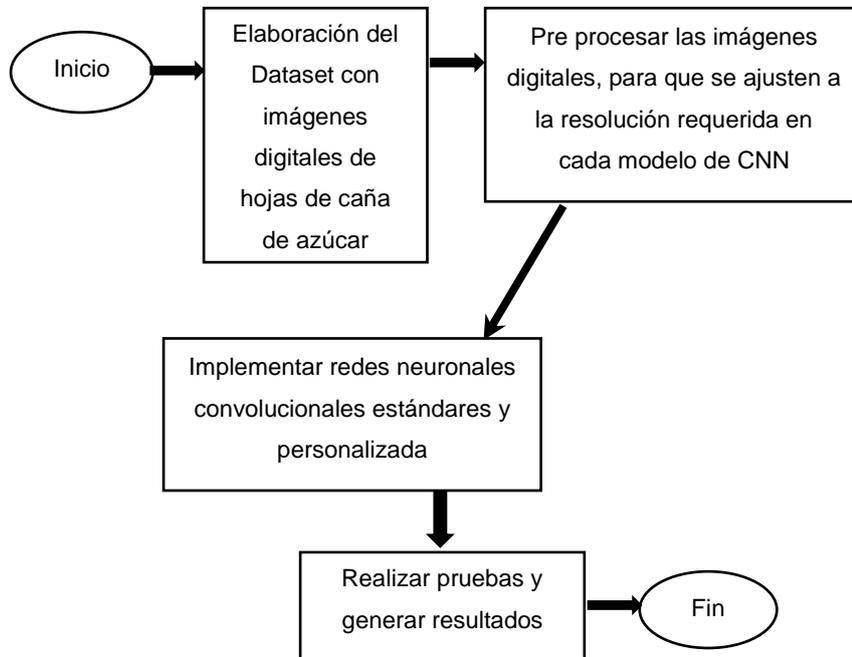


Figura 12. Diagrama de los objetivos propuestos para la resolución del problema abordado.

Fuente: Elaboración propia

1. Elaborar el dataset con imágenes digitales de las hojas de caña de azúcar.

Para la obtención de las imágenes que se utilizaron en esta investigación, primero se concurre a campos de cultivo de agricultores del distrito de Ferreñafe, a los cuales se les pidió autorización para recolectar algunas muestras de las hojas de sus cultivos, entre especímenes sanos y enfermos. Para diferenciarlas, se utilizó un libro de referencia (Ovalle, Guía para la identificación de enfermedades de la caña de azúcar, 2018).

Posteriormente, dichas hojas fueron recortadas en secciones de 6 cm de largo, teniendo en cuenta que el grosor de las mismas sea mayor de 2 cm, todo esto con la

finalidad que puedan ser colocadas dentro del espacio remarcado en el escáner, el cual es de 6x6 cm, habiendo usado para ello una cartulina color negro, con sendos recortes de 6x6 cm, logrando un total de doce imágenes por cada escaneado realizado, lo que permitió poder diferenciar entre el área de interés del resto de la imagen escaneada, dado el caso que el área total corresponde a una hoja A4, con un área aproximada de 29.7 x 21cm. Se decidió realizarlo utilizando esa guía visual debido a que el problema que se abordó no se trataba de segmentación, sino de clasificación, mostrado a continuación:



Figura 13. Disposición de las hojas de caña de azúcar en el escáner

Fuente: Elaboración propia

Tabla 13.

Características de impresora utilizada

Característica	Detalle
Fabricante	Epson
Modelo	L350

Fuente: Elaboración propia

El escáner utilizado está equipado en una impresora multifuncional con las siguientes características:

En el caso de la configuración del escáner para la obtención de imágenes, fue la siguiente:

Tabla 14.

Configuración del escáner.

Opción	Detalle
Tamaño de documento	210x297 mm
Modelo de color	RGB
Puntos por pulgada	300
Formato de imagen	PNG
Compensación de exposición	0.0

Fuente: Elaboración propia

Por lo expuesto, el proceso de adquisición de imágenes digitales fue en ambiente controlado, obteniendo finalmente 680 imágenes, las cuales fueron organizadas en carpetas, separándolas en primera instancia en entrenamiento y validación, luego en ambas divisiones, según si se trataba de muestras sanas, infectadas con *Acidovorax Avenae*, *Bipolaris Sacchari* o *Xanthomonas Albilineass* siendo colocadas en sendas carpetas de acuerdo al tipo de enfermedad. Obteniendo en total 170 imágenes por cada clase. Se separó 110 imágenes para el entrenamiento y 50 para la validación, para cada una de las cuatro clases de esta investigación.



Figura 14. Imagen digital obtenida de una hoja de caña de azúcar, utilizando el método de captura de imágenes planteado. Fuente: Elaboración Propia

2. Pre procesar las imágenes digitales de las hojas de caña de azúcar.

Una vez se obtuvo las imágenes, fue necesario ajustar las dimensiones de las mismas, puesto que las redes seleccionadas tienen un tamaño de input fijo, siendo esta de 224x224x3, el 3 representa a las dimensiones de los colores que constituyen una imagen RGB, siendo estos rojo, verde y azul.

Para automatizar el cambio de resolución y estandarizar el nombre de cada imagen se implementó un script en el lenguaje de programación python, siendo el siguiente:

```
from PIL import Image
import glob
image_list = []
resized_images = []
for filename in glob.glob('E:/ML IMAGEN/carpeta contenedor/*.jpg'):
```

Para ello se utilizó la librería de python “Pillow” la cual permite manipular de diferentes formas archivos en variados formatos, además del módulo “GLOB”, con el cual es posible utilizar rutas de acceso a carpetas y archivos. Las imágenes obtenidas fueron cambiadas a la resolución 224x224, el cual es el tamaño de la capa de entrada de las redes neuronales utilizadas.

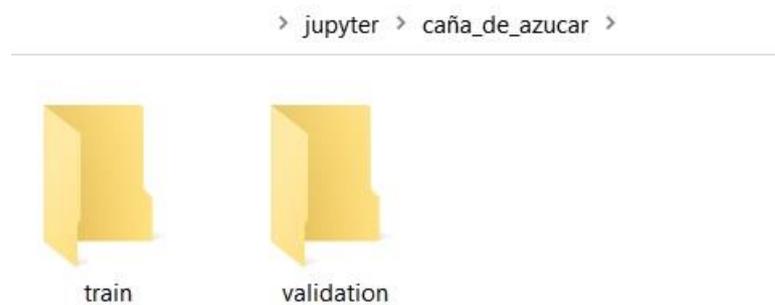


Figura 15. Carpetas de entrenamiento y validación.

Fuente: Elaboración propia



Figura 16. Carpetas de hojas sanas y enfermas.

Fuente: Elaboración propia

Posteriormente, para incrementar de forma artificial número de imágenes disponibles, En este se aplicó un rango de rotación de 20 grados, se varió la altura en un 10%, se aplicó un zoom del 20% y se giró las imágenes en su eje horizontal, se recurrió a técnicas de data augment, mediante el código siguiente:

```
train_gen=ImageDataGenerator(
    rotation_range=20
    height_shift_range=0.1,
    zoom_range=0.2,
    horizontal_flip=True,
    preprocessing_function=preprocess_input
)

val_gen= ImageDataGenerator(
    preprocessing_function = preprocess_input
)

test_gen = val_gen.flow_from_directory(valid_path,
    target_size=IMAGE_SIZE)
```

Figura 17. Fragmento del código que permite usar técnicas de data augmentation. Fuente: Elaboración propia

- 3. Seleccionar e implementar redes neuronales convolucionales estándares y personalizadas para el procesamiento de las imágenes del dataset.**

Con la finalidad de seleccionar las redes neuronales convolucionales se realizó la búsqueda de documentación de investigaciones realizadas hasta la fecha de la realización de esta investigación, en el campo de visión por computador, para ello se utilizó las siguientes cadenas de búsqueda, las cuales fueron usadas en los repositorios científicos EBSCO, ScienceDirect, IOPScience y Scopus.

Tabla 15.

Cadenas de búsqueda utilizadas en la presente investigación.

Cadena de búsqueda	Número total de resultados
“leaf disease computer vision”	2791
“classification disease computer vision”	20092
“classification disease deep learning”	43236
“plant disease convolutional neural network”	2058

Fuente: Elaboración propia

En la revisión de las investigaciones resultado al utilizar esas cadenas de búsqueda, se observó el uso de diversas arquitecturas de redes neuronales convolucionales, con el presente resultado:

Tabla 16.

Investigaciones relacionadas a redes neuronales convolucionales

N°	Técnica	Investigación	Autores	Resultado	
01	CNN arquitectura propia	de Clasificación de nivel de crecimiento de cultivos	Indri Nanna Herman, Mohd	Neforawati, Suryana Othman	82%
02	Multilayer Perceptron - ANN	Detección de la enfermedad de sigatoka en plantaciones de bananos	F.O. V.O.	Sweetwilliams, Matthews, E. D.T. Adetiba, Babalola, V. Akande.	98%.
03	CNN arquitectura propia	de Detección de enfermedades en hojas de yuca	G. Sambasivam y Geoffrey Duncan Opiyo		92%
04	VGGnet	Clasificación de imágenes médicas con detección de bordes	Suvajit Dutta Shalva Rai and Vijayarajan V.		78.3%
05	CNN arquitectura propia	de Reconocimiento de códigos de seguridad	Jingtian Gu		99.9%
06	CNN arquitectura propia	de Reconocimiento de números	Yunfei Lai		98.85%
07	Support Machine	Vector Reconocimiento entre tumores malignos y benignos de osteosarcoma	Harish Arunachalam, Rashika Mishra, Ovidiu Daescu	Babu	89.9%

08	CNN arquitectura propia	de Diagnóstico enfermedad Alzheimer	de la	Eman N. Marzban, Ayman M. Eldeib, , Inas A. Yassine y Yasser M. Kadah	93.18%
09	CNN arquitectura propia	de Diagnóstico enfermedades con uso de imágenes ultrasonidos	de	Xiaowen Liang, Jinsui Yu, Jianyi Liao y Zhiyi Chen	88.1%
10	CNN VGGNET	Diagnóstico enfermedades plantaciones de arroz	de	RuhuaLu,YalanLi, PanYang y WenfenZhang	83.57 %
11	CNN GoogLeNet	Identificación de plagas en cultivos		Y Nagaraj, A Hema Sai Teja1 y · A V Narasimhadhan	97%
12	CNN ResNet	Reconocimiento enfermedades tomates	de	Tao Zhang , Xiankun Zhu , Yiqing Liu , Kun Zhang	88.3%
13	CNN MobileNetV1	Reconocimiento pestes en tomates	de	Liu, Y., Limin, Y., Zhenghong, Y., Wanlin, G., & Yanzhao, R.	97.19%
14	CNN Propia	Deteccion enfermedades plantones de arroz	de	Purbasari, I. Y., Rahmat, B., & Putra, C. S.	91%
15	CNN Resnet	Deteccion enfermedades en hojas de liima	de	Han, J., Zhi, P. X., & Yan, G.	83.7%

Fuente: *Elaboración propia*

Con los resultados previamente obtenidos, se tomó como muestra 3 arquitecturas de redes neuronales convolucionales, habiendo tenido en cuenta que estas tuvieran una precisión por encima del 85%, siendo estas MobileNetV1, VGG16, RESNET50, además se implementó una arquitectura propia basada en las anteriores mostradas, En cuanto a la red convolucional propia, se utilizó el API funcional de python.

Fig
ura
18.
Blo
que
de
iden
tida
d
con
API
func
ional.

```
#Definiendo el bloque de indentidad
def Bloque_Identidad (input_, kernel_size, filters):
    f1, f2, f3 = filters
    x= Conv2D(f1, (1,1),)(input_)
    x = BatchNormalization()(x)
    x = Activation('relu')(x)
    x = Conv2D(f2, kernel_size, padding='same',)(x)
    x = BatchNormalization()(x)
    x = Activation('relu')(x)
    x = Conv2D(f3, (1,1),)(x)
    x = BatchNormalization()(x)
    x = add([x, input_])
    x = Activation('relu')(x)
    return x
```

Fuente: *Elaboración propia*

```
#Definiendo el bloque de convolución de la arquitectura propuesta
```

```
def bloq_conv(input_,  
              kernel_size,  
              filters,  
              strides=(2,2)):  
    f1, f2, f3 = filters  
  
    x = Conv2D(f1, (1,1), strides=strides,)(input_)  
    x = BatchNormalization()(x)  
    x = Activation('relu')(x)  
    x = Conv2D(f2, kernel_size, padding='same,')(x)  
    x = BatchNormalization()(x)  
    x = Activation('relu')(x)  
    x = Conv2D(f3, (1,1),)(x)  
    x = BatchNormalization()(x)  
    shortcut = Conv2D(f3, (1,1), strides=strides,)(input_)  
    shortcut = BatchNormalization()(shortcut)  
    x = add([x, shortcut])  
    x = Activation('relu')(x)  
    return x
```

Figura 19. Bloque de convolución con API funcional.

Fuente: *Elaboración propia*

```

#Red propuesta
i = Input(shape=TAM_IM + [3])
x = ZeroPadding2D(padding=(3,3))(i)
x = Conv2D(64, (7,7), strides=(2,2), padding='valid')(x)
x = BatchNormalization()(x)
x = Activation('relu')(x)
x = ZeroPadding2D(padding=(1,1))(x)
x = MaxPooling2D((3,3), strides=(2,2))(x)
x = bloq_conv (x, 3, [64,64,256], strides= (1,1))
x = Bloque_Identidad (x, 3, [64,64,256])
x = Bloque_Identidad (x, 3, [64,64,256])
x = bloq_conv (x, 3, [128,128,512])
x = Bloque_Identidad (x, 3, [128,128,512])
x = Bloque_Identidad (x, 3, [128,128,512])
x = Flatten()(x) prediction = Dense(len(folders), activation='softmax')(x)

#Instancia del modelo
model = Model(inputs=i, outputs=prediction)

```

Figura 20. Arquitectura de la red propuesta.

Fuente: *Elaboración propia*

```
#Importación de librerías y funciones utilizadas
from tensorflow.keras.layers import Input, Dense, Lambda, Flatten
from tensorflow.keras.models import Model
from tensorflow.keras.applications.resnet50 import ResNet50,
preprocess_input
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import ImageDataGenerator

from sklearn.metrics import confusion_matrix, f1_score, roc_curve,
precision_score, recall_score, accuracy_score, roc_auc_score,
classification_report
from sklearn import metrics
import numpy as np
import matplotlib.pyplot as plt

from glob import glob
import itertools
```

Figura 21. Código de importación de librerías y funciones.

Fuente: *Elaboración propia*

```

Tam_Im = [224,224]
epocas=140
Lotes=4
from google.colab import files
uploaded = files.upload()
!unzip -nq Investigacion.zip

ent_path='Investigacion/Entrenamiento'
valida_path='Investigacion/Validacion'

arch_imagen =glob(train_path + '/*/*.jpg')
valid_imagen_arch=glob(valid_path + '/*/*.jpg')
carpeta=glob(ent_path + '/*')
#en el caso donde dice "ResNet50 se cambia por la CNN a usar
resn = ResNet50(input_shape=Tam_Im + [3], weights='imagenet',
include_top=False)
for layer in resn.layers:
    layer.trainable=False
x = Flatten()(res.output)
prediction = Dense(len(folders), activation='softmax')(x)
modelo = Model(inputs=res.input, outputs=prediction)
modelo.compile(loss='sparse_categorical_crossentropy',
optimizer='adam', metrics=['accuracy'])

```

Figura 22. Fragmento de código usado.

Fuente: Elaboración propia

```

train_generator=ImageDataGenerator(rotation_range=15,
                                   width_shift_range=0.2,
                                   height_shift_range=0.2,
                                   shear_range=0.15,
                                   zoom_range=0.15,
                                   horizontal_flip=False,
                                   vertical_flip=True,
preprocessing_function=preprocess_input
                                   )
val_gene= ImageDataGenerator(
    preprocessing_function = preprocess_input
)
test_gen = val_gen.flow_from_directory(valida_path,
target_size=Tam_Im)
print(test_gene.class_indices)
labels=[None]*len(test_gene.class_indices)
for k,v in test_gene.class_indices.items():
    labels[v]=k
train_generator=train_generator.flow_from_directory(
    train_path, target_size=Tam_Im,
    shuffle=True, batch_size=lotes,
    class_mode='sparse',)
valid_generator=val_gen.flow_from_directory(
    valida_path, target_size=Tam_Im,
    shuffle=False, batch_size=lotes,
    class_mode='sparse',)

```

Figura 23. Fragmento de código usado.

Fuente: Elaboración propia

```

c = modelo.fit(
    train_generator,
    validation_data=valid_generator,
    epochs=epocas,
    steps_per_epoch=len(arch_imagen)//lotes,
    validation_steps=len(valid_imagen_arch)//lotes,
)
plt.plot(c.history['loss'], label='train loss')
plt.plot(c.history['val_loss'], label='validation loss')
plt.legend()
plt.plot(c.history['accuracy'], label='train accuracy')
plt.plot(c.history['val_accuracy'], label='val accuracy')
plt.legend()
def obt_confusion_matrix(data_path, N):
    #Ver los datos en el mismo orden para ambas predicciones de los objetivos
    print("genera matriz de confusion", N)
    predictions= [ ]
    targets = [ ]
    i = 0
    n_imagenes= 0
    for x,y in val_gene.flow_from_directory(
        data_path,
        target_size=Tam_Im,
        shuffle=False,
        batch_size=lotes * 2):
        i += 1
        n_images += len(y)
        if i % 50 == 0:
            print(f'{n_images} imagenes procesadas.')
        p = modelo.predict(x)
        p=np.argmax(p,axis=1)
        y=np.argmax(y, axis=1)
        predictions = np.concatenate((predictions, p))
        targets = np.concatenate((targets, y))

```

Figura 24. Fragmento de código usado.

Fuente: Elaboración propia

IV. CONCLUSIONES Y RECOMENDACIONES

3.1. Conclusiones.

- a) En la presente investigación se realizó la toma de imágenes en ambiente controlado, al utilizar un escáner y luz artificial, por lo que no se garantiza que se pueda obtener resultados similares o superiores, en caso de capturar las fotografías en ambiente no controlado, con diferentes grados de iluminación.
- b) Para el pre procesamiento de las imágenes, se tuvo que cambiar de resolución las imágenes obtenidas, puesto que las redes neuronales convolucionales utilizadas solo permitían inputs de 224x224 píxeles. También se tuvo que utilizar data augmentation, puesto que en la revisión de la bibliografía, los datasets usados eran muy extensos, como el dataset "fruits360" o el dataset "plantvillage".
- c) En la selección de las redes neuronales convolucionales que se utilizó se tuvo en cuenta que las investigaciones realizadas se encuentren en los repositorios científicos ScienceDirect, Scopus, Ebsco, y que tengan buenos resultados, con un porcentaje mayor al 90% de precisión.
- d) La red neuronal convolucional que arrojó resultados más bajos fue MobileNet1, con una precisión del 70.41 % y Exactitud de 88.34 %, además de ser el menos eficiente en cuanto al uso de GPU y memoria RAM, siendo estos de 7.88 GB y 1.92 GB respectivamente, pero teniendo el menor tiempo de respuesta total, el cual fue de 1151 segundos .

La red neuronal convolucional ResNet50 fue la que dio mejores resultados, los cuales fueron una precisión del 83.33% y Exactitud de 93.92 %. En cuanto al uso de recursos de cómputo estos fueron de 7.51 GB en el caso de GPU y 1.92 GB de memoria RAM, tomándole un tiempo de respuesta total de 1579 segundos. La red neuronal convolucional propuesta es la tercera en cuanto a resultados obtenidos, con una precisión del 71.67 % y Exactitud de 89.28 %. En cuanto al uso de recursos computacionales, estos

fueron de 1.75 GB de GPU y 1.75 de memoria RAM, en un tiempo de respuesta total de 1468 segundos, siendo el más eficiente de todos en este aspecto.

3.2 Recomendaciones.

Se recomienda el uso de las máquinas virtuales de google collab en caso se desee replicar esta investigación y el investigador no cuente con un computador con GPU, significando que la ejecución del algoritmo podría demorar mucho más de lo previsto. Además estas son de uso gratuito y son compatibles con el lenguaje de programación Python, mismo que fue usado en esta investigación.

Otra recomendación es, que una vez obtenidas la hojas con las cuales se evaluarán las arquitecturas de redes neuronales, se capturen las imágenes digitales de forma oportuna, esto es debido a que suelen secarse muy rápido y en esa situación es una ardua tarea por mantenerlas aplanadas sobre el cristal del scanner.

Referencias

- [1] MINAGRI, «Ministerio de Agricultura,» Ministerio de Agricultura - Peru, 12 junio 2017.
[En línea]. Available:
<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwjX-q3ihcrqAhXFLLkGHYkbCQgQFjADegQIBRAB&url=https%3A%2F%2Fwww.minagri.gob.pe%2Fportal%2Fanalisis-economico%2Fanalisis-2017%3Fdownload%3D11375%3Aboletin-produccion-de-cana-de-azucar&u>. [Último acceso: 17 mayo 2020].
- [2] G. Pantaleon, «MANEJO ECOLOGICO DE PLAGAS Y ENFERMEDADES DE LA CAÑA DE AZUCAR,» ATAM Mexico, Mexico, 2018.
- [3] L. V. Dicks, What works in conservation? Using expert assessment of summarised evidence to identify practices that enhance natural pest control in agriculture, Cambridge: Open Books, 2015, pp. 15,16.
- [4] Y. Muñoz, «EL CARBÓN DE LA CAÑA DE AZÚCAR (*Ustilago scitaminea* Sydow),» *Revista de protección vegetal*, vol. 19, nº 1, pp. 1-6, 2004.
- [5] T. M. Mitchell, Machine Learning, Nueva York: McGraw Hill, 1997.
- [6] A. M. David Poole, Artificial intelligence foundations of computations agents, Cambridge: Cambridge University Press, 2017.
- [7] A. Bosch, Deep learning: principios y fundamentos, Cataluña: UOC, 2020.
- [8] G. Hinton, «Learning multiple layers of representation,» *TRENDS in Cognitive Sciences*, vol. 11, nº 10, pp. 1,2, 2007.
- [9] M. Schmidt, «Painless Stochastic Gradient: Interpolation, Line-Search, and Convergence

Rates,» Universidad de Montreal, Montreal, 2019.

[1 S. Osindero, «A fast learning algorithm for deep belief nets,» Universidad de Toronto,
0] Toronto, 2006.

[1 I. Sutskever, «ImageNet Classification with Deep Convolutional,» Universidad de
1] Toronto, Toronto, 2013.

[1 E. Bienstock, «Neural Networks and the Bias/Variance Dilemma,» *Neural*
2] *Computation* , vol. 4, nº 1, pp. 5,6, 1992.

[1 T. Jain, «OpenGenus,» 11 Octubre 2017. [En línea]. Available:
3] <https://iq.opengenus.org/basics-of-machine-learning-image-classification-techniques/>.
[Último acceso: 20 diciembre 2021].

[1 E. Achtert, «Advances in Knowledge Discovery and Data Mining,» Sudafrica, 2006.
4]

[1 Corinna Cortes, Support Vector Networks, Estados Unidos: AT&T Labs, 1995.
5]

[1 X. Yan, Linear regression analysis: Theory and computing, Estados Unidos: World
6] Scientific, 2009.

[1 D. McFadden, Conditional Logit Analysis of Qualitative Choice Behavior, Nueva York:
7] Academic Press, 1976.

[1 J. Rennie, «Tackling the poor assumptions of naive bayes text classifiers,» MIT,
8] Cambridge, 2003.

[1 J. H. Friedman, «Regularized Discriminant Analysis,» *Journal of the American Statistical*
9] *Association*, vol. 84, nº 405, pp. 1-3, 1989.

[2 K. S. Beyer, «When Is "Nearest Neighbor" Meaningful?,» de *In Proceedings of the 7th*
0] *International Conference on Database Theory*, Heidelberg, 1999.

[2 R. C. Gonzalez, Digital Image Processing, Nueva York: Pearson, 2017.
1]

- [2 A. Kimmel, «Regularized Laplacian Zero Crossings as Optimal Edge Integrators,»
2] *International Journal of Computer Vision*, vol. 53, nº 3, pp. 225-243, 2003.
- [2 A. M. Domink Scherer, «valuation of Pooling Operations in convolutional Architectures
3] for objects recognition,» de *20th International Conference on Artificial Neural Networks (ICANN)*, Tesalonika, 2010.
- [2 D. Wang, «Unsupervised Learning: Foundations of Neural Computation,» *AI Magazine*,
4] vol. 22, nº 2, p. 101, 2001.
- [2 J. E. Solem, *Programming Computer Vision With Python*, Suecia: O'Reilly, 2012.
5]
- [2 D. A. K. H. Jim Hugunin, *Numerical python*, California: McGraw Hill, 1999.
6]
- [2 W. Ovalle, «Guia para la identificación de enfermedades de la caña de azucar,»
7] Cengicaña, Guatemala, 2018.
- [2 S. A. Lopes, «Xanthomonas Albilineans Detection, Identification,» Louisiana State
8] University, Louisiana, 1996.
- [2 J. I. Victoria, «Enfermedades de la caña de azúcar en Colombia,» Cengicaña, Cali, 1995.
9]
- [3 J. A. Borges, «caña de azúcar (*Saccharum spp.*) con fines forrajeros en Yaracuy,
0] Venezuela,» *Revista de la Facultad de Agronomía*, vol. 25, nº 3, p. 5, 2008.
- [3 G. I. W. Claude Sammut, «Encyclopedia of Machine Learning,» Springer US, Boston,
1] 2010.
- [3 G. Nolte, «Sugar anual Information Network,» GAIN, Peru, 2020.
2]
- [3 K. Nigam, *Learning to Classify Text from Labeled and Unlabeled Documents*, Pittsburg:
3] Carnegie melon university, 2000.
- [3 R. M. M. Jose Tomas Palma, *Inteligencia Artificial*, McGraw Hill, 2008.

4]

[3 G. Hinton, *Unsupervised Learning: Foundations of Neural Computation*, Massachusetts:

5] The MIT Press, 1999.

[3 M. M. Dessouky, «Multi-class Image Classification Using Deep Learning Algorithm,» de

6] *Journal of Physics: Conference Series*, Egipto, 2020.

ANEXOS.

Anexo 1. Resolución de aprobación del trabajo de investigación

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO
RESOLUCIÓN N°1332-2020/FIAU-USS

Pimentel, 17 de julio de 2020

VISTO:

El Acta de reunión N°1606-2020, de fecha 16 de junio de 2020 del Comité de investigación de la Escuela profesional de INGENIERIA DE SISTEMAS, para la ejecución : "EVALUACIÓN DE LA EFICIENCIA DE TÉCNICAS DE APRENDIZAJE AUTOMÁTICO PARA LA IDENTIFICACIÓN DE ENFERMEDADES EN LOS CULTIVOS DE CAÑA DE AZÚCAR", presentado por el(los) tesista(s) REUPO ATTO PAUL GIANMARCO, del Programa de estudios INGENIERIA DE SISTEMAS, y;

CONSIDERANDO:

Que, de conformidad con la Ley Universitaria N° 30220 en su artículo 48° 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 21° señala: "Los temas de trabajo de investigación, trabajo académico y tesis son aprobados por el Comité de Investigación y 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 24° 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 25° señala: "El tema debe responder a alguna de las líneas de investigación institucionales de la USS S.A.C."

Que, en el Acta de reunión N°1606-2020 de fecha 16 de junio de 2020, del Comité de investigación de la Escuela profesional de INGENIERIA DE SISTEMAS, se indica entre los acuerdos la aprobación del Proyecto de tesis denominado "EVALUACIÓN DE LA EFICIENCIA DE TÉCNICAS DE APRENDIZAJE AUTOMÁTICO PARA LA IDENTIFICACIÓN DE ENFERMEDADES EN LOS CULTIVOS DE CAÑA DE AZÚCAR" de la línea de investigación de INFRAESTRUCTURA, TECNOLOGÍA Y MEDIO AMBIENTE, a cargo de REUPO ATTO PAUL GIANMARCO en condición de estudiante, del Programa de estudios INGENIERIA DE SISTEMAS.

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, el Proyecto de denominado "EVALUACIÓN DE LA EFICIENCIA DE TÉCNICAS DE APRENDIZAJE AUTOMÁTICO PARA LA IDENTIFICACIÓN DE ENFERMEDADES EN LOS CULTIVOS DE CAÑA DE AZÚCAR", perteneciente a la línea de investigación de INFRAESTRUCTURA, TECNOLOGÍA Y MEDIO AMBIENTE, a cargo de REUPO ATTO PAUL GIANMARCO, del Programa de estudios INGENIERIA DE SISTEMAS.

ARTÍCULO 2°: ESTABLECER, que la inscripción del Título de Proyecto de tesis se realice a partir de emitida la presente resolución y tendrá una vigencia de dos (02) años.

ARTÍCULO 3°: DEJAR SIN EFECTO, toda Resolución emitida por la Facultad que se oponga a la presente Resolución.

REGÍSTRESE, COMUNÍQUESE Y ARCHÍVESE



Dr. Mario Fernando Ramos Mescol
Decano - Facultad de Ingeniería,
Arquitectura y Urbanismo
UNIVERSIDAD SEÑOR DE SIPÁN SAC.



MBA. María Noelia Sialer Rivera
Secretaría Académica / Facultad de Ingeniería,
Arquitectura y Urbanismo
UNIVERSIDAD SEÑOR DE SIPÁN SAC.

Anexo 2. Instrumentos de recolección de datos.

Matrices de confusión obtenidas al ejecutar la red neuronal propuesta.

Tabla 17

Matriz de confusión contenida de resultados de entrenamiento en red propuesta

Síntoma visible	C. Avenae	B. Sacchari	Hoja Sana	X. Albillineans
A. Avenae	91	2	12	5
B. Sacchari	10	78	2	20
Hoja Sana	3	1	101	5
X. Albillineans	2	0	1	107

Fuente: Elaboración propia

Tabla 18

Matriz de confusión contenida de resultados de validación en red propuesta

Síntoma visible	II. Avenae	B. Sacchari	Hoja Sana	X. Albillineans
A. Avenae	44	3	8	5
B. Sacchari	12	25	2	21
Hoja Sana	2	1	46	11
X. Albillineans	0	0	3	57

Fuente: Elaboración propia

Matrices de confusión obtenidas al ejecutar la red neuronal convolucional

MobileNetV1.

Tabla 19

Matriz de confusión contenida de resultados de entrenamiento en red MobileNet1

Síntoma visible	A. Avenae	B. Sacchari	Hoja Sana	X. Albillineans
A. Avenae	103	0	0	7
B. Sacchari	0	94	1	15
Hoja Sana	3	0	98	9
X. Albillineans	0	2	0	108

Fuente: Elaboración propia

Tabla 20

Matriz de confusión contenida de resultados de validación en red MobileNet1

Síntoma visible	B. Avenae	B. Sacchari	Hoja Sana	X. Albillineans
A. Avenae	45	4	2	9
B. Sacchari	8	28	0	24
Hoja Sana	8	0	42	10
X. Albillineans	2	4	0	54

Fuente: Elaboración propia

Matrices de confusión obtenidas al ejecutar la red neuronal convolucional VGG16.

Tabla 21

Matriz de confusión contenida de resultados de entrenamiento en red VGG16.

Síntoma visible	C. Avenae	B. Sacchari	Hoja Sana	X. Albillineans
A. Avenae	90	9	2	0
B. Sacchari	1	109	0	0
Hoja Sana	1	2	107	0
X. Albillineans	5	29	7	69

Fuente: Elaboración propia

Tabla 22

Matriz de confusión contenida de resultados de validación en red VGG16.

Síntoma visible	A. Avenae	B. Sacchari	Hoja Sana	X. Albillineans
A. Avenae	53	6	0	1
B. Sacchari	6	47	3	4
Hoja Sana	2	5	51	2
X. Albillineans	7	13	5	35

Fuente: Elaboración propia

Matrices de confusión obtenidas al ejecutar la red neuronal convolucional ResNet50.

Tabla 23

Matriz de confusión contenida de resultados de entrenamiento en red ResNet50

Síntoma visible	C. Avenae	B. Sacchari	Hoja Sana	X. Albillineans
A. Avenae	101	3	5	1
B. Sacchari	0	106	1	3
Hoja Sana	0	0	110	0
X. Albillineans	0	2	7	101

Fuente: Elaboración propia

Tabla 24

Matriz de confusión contenida de resultados de validación en red ResNet50

Síntoma visible	A. Avenae	B. Sacchari	Hoja Sana	X. Albillineans
A. Avenae	50	4	2	4
B. Sacchari	4	42	2	12
Hoja Sana	1	0	56	3
X. Albillineans	0	5	3	52

Fuente: Elaboración propia