



**FACULTAD DE INGENIERÍA, ARQUITECTURA Y
URBANISMO**

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

TESIS

**Método de Clasificación Automática de Deficiencias
Nutricionales en Hojas de Cafeto Mediante Procesamiento
de Imágenes Digitales y Aprendizaje Profundo**

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

Autor

Bach. Luna Becerra Jherson Isac
ORCID: <https://orcid.org/0000-0003-2623-713X>

Asesor

Dr. Forero Vargas Manuel Guillermo
ORCID: <https://orcid.org/0000-0001-9972-8621>

Línea de Investigación

Infraestructura, Tecnología y Medio Ambiente

Pimentel – Perú

2024

**MÉTODO DE CLASIFICACIÓN AUTOMÁTICA DE DEFICIENCIAS
NUTRICIONALES EN HOJAS DE CAFETO MEDIANTE PROCESAMIENTO DE
IMÁGENES DIGITALES Y APRENDIZAJE PROFUNDO**

Aprobación del jurado

DR. TUESTA MONTEZA VICTOR ALEXCI

Presidente del Jurado de Tesis

DR. ATALAYA URRUTIA CARLOS WILLIAM

Secretario del Jurado de Tesis

MG. CACHAY MACO JUNIOR EUGENIO

Vocal del Jurado de Tesis




DECLARACIÓN JURADA DE ORIGINALIDAD

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

MÉTODO DE CLASIFICACIÓN AUTOMÁTICA DE DEFICIENCIAS NUTRICIONALES EN HOJAS DE CAFETO MEDIANTE PROCESAMIENTO DE IMÁGENES DIGITALES Y APRENDIZAJE PROFUNDO

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

En virtud de lo antes mencionado, firman:

Luna Becerra Jherson	DNI: 73505562	
----------------------	---------------	---

Pimentel, 06 de marzo de 2024.

Dedicatoria

Dedicado a mi querida madre, Honorina Becerra López.

Aunque no te pueda ver, sé que siempre estás conmigo, en cada momento y serás eternamente la fuente de inspiración que me impulsa a seguir adelante. Desde el cielo, sé que me cuidas y proteges con tu amor incondicional. Gracias por ser mi guía, mi mayor ejemplo y mi más grande motivación para seguir adelante.

A mi padre, y hermanos, quienes han sido el sustento para poder seguir estudiando y por el apoyo incondicional recibido de cada uno.

Agradecimientos

Primeramente, agradecer a Dios por la vida, la salud, por brindarme la fortaleza de seguir adelante, afrontando obstáculos en la vida diaria e ir cumpliendo mis objetivos trazados.

También quiero agradecer a mi asesor de tesis, al Dr. Manuel Guillermo Forero Vargas, por su invaluable orientación, tiempo dedicado a las revisiones y correcciones, así como por su constante apoyo y guía a lo largo de este proceso. Asimismo, quiero extender mi gratitud a los docentes de nuestra casa de estudios, cuyos conocimientos impartidos han sido esenciales para el desarrollo de esta investigación.

Asimismo, quiero retribuir el apoyo incondicional de mis queridos padres y hermanos, quienes han sido mi fuente de inspiración y motivación en todo momento. Su apoyo constante han sido un pilar fundamental para el logro de mis metas académicas y profesionales. Agradezco también a mis compañeros de estudio, a quienes en esta travesía se volvieron amigos.

Índice

Dedicatoria	IV
Agradecimientos.....	V
Índice de tablas, figuras y fórmulas (de ser necesario).....	VII
Resumen	XI
Abstract	XII
I. INTRODUCCIÓN.....	13
1.1. Realidad problemática.....	13
1.2. Formulación del problema	29
1.3. Hipótesis	29
1.4. Objetivos	29
1.5. Teorías relacionadas al tema	30
II. MATERIALES Y MÉTODO.....	64
2.1. Tipo y Diseño de Investigación.....	64
2.2. Variables, Operacionalización	64
2.3. Población de estudio, muestra, muestreo y criterios de selección.....	67
2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad..	70
2.5. Procedimiento de análisis de datos	78
2.6. Criterios éticos.....	83
III. RESULTADOS Y DISCUSIÓN	85
3.1. Resultados	85
3.2. Discusión.....	92
IV. CONCLUSIONES Y RECOMENDACIONES.....	119
4.1. Conclusiones.....	119
4.2. Recomendaciones.....	120
REFERENCIAS.....	122
ANEXOS	129

Índice de tablas

Tabla 1: Principales productores de café del mundo, tomado de cinco cosechas.....	14
Tabla 2: <i>Valores de exportación del café en los últimos años expresados en dólares</i>	15
Tabla 3: <i>Matriz de confusión</i>	34
Tabla 4: Operacionalización de la variable.....	65
Tabla 5: Población considerada para la etapa del procesamiento de imágenes	67
Tabla 6: Población considerada para la etapa de clasificación mediante aprendizaje automático y aprendizaje profundo.....	68
Tabla 7: Muestra seleccionada para la etapa del procesamiento de imágenes	69
Tabla 8: Muestra seleccionada para la etapa de clasificación.....	69
Tabla 9: Resultado del consumo de recursos informáticos del sistema y tiempo de respuesta utilizado para el procesamiento de las imágenes y entrenamiento del modelo propuesto ...	85
Tabla 10: Resultado del rendimiento global del modelo propuesto	86
Tabla 11: Resultado del rendimiento por clases del moldeo propuesto	88
Tabla 12: Comparación de resultados del modelo CNN propio con otras arquitecturas preexistentes	91
Tabla 13: Características de las deficiencias nutricionales para identificarlos visualmente.	98

Índice de Figuras

Fig. 1: Valores FOB de exportación del café en los últimos años expresados en dólares [6].	16
Fig. 2 : Plantación de cafeto.....	30
Fig. 3: Inteligencia Artificial y su jerarquía de conceptos [26].	32
Fig. 4: Esquema básico de una neurona artificial [26].	37
Fig. 5: Una red neuronal convolucional CNN. Tomado de Diego Calvo [35].	42
Fig. 6: La figura muestra la capa convolucional 2D. Tomado de Bosch, Casas y Lozano [26].	43
Fig. 7: La figura muestra la operación de convolución de una imagen de 4x4 y un kernel de 2x2.....	44
Fig. 8: Representación de capa Max Pooling de una matriz de 4x4 para obtener una resultante de 2x2. Tomado de [31]......	44
Fig. 9: Ejemplo de padding de 2x2. Tomado de Soria, Rodríguez y García [31].	45
Fig. 10: Arquitectura GG16. Tomado de Soria, Rodríguez y García [31].	46
Fig. 11: Ilustración de la arquitectura de red del modelo VGG-19: totalmente conectado. Tomado de [36].	46
Fig. 12: Representación gráfica del modelo Inception V3. Tomado de Benjamín Qochuk [37].	47
Fig. 13: Estructura de la arquitectura Mobile net. Tomado [38].	48
Fig. 14: <i>Representación gráfica general de procesamiento de imágenes. Tomado de Bosch, Casas y Lozano [26].</i>	49
Fig. 15: representación de aplicar el método Otsu. Tomado de MathWorks [46].	54
Fig. 16: Representación de imagen RGB, binarizada y aplicada la región de interés. Tomado de ICHI.PRO [47]......	56
Fig. 17: Representación gráfica del de color HSV. Tomado de Forero Vargas [39].	58

Fig. 18: Imágenes tomadas considerando los protocolos iniciales	73
Fig. 19: imágenes tomadas considerando los parámetros finales establecidos	74
Fig. 20: imágenes aplicando morfología 2x2, 3x3 y 5x5.....	75
Fig. 21: Imagen redimensionada 128 x 240 pixeles	75
Fig. 22: Resultado del rendimiento por clases del moldeo propuesto	88
Fig. 23: Método propuesto. Fuente: Elaboración propia.....	97
Fig. 24: Vista superior prototipo propuesto para la adquisición de las imágenes. Fuente elaboración propia.	104
Fig. 25: Vista frontal del prototipo propuesto para la adquisición de las imágenes. Fuente elaboración propia.	104
Fig. 26: Prototipo construido para la adquisición de las imágenes	105
Fig. 27: Vista celular controlado por un computador de manera remota utilizando la aplicación AnyDesk	106
Fig. 28: Organización de los directorios para el conjunto de datos	107
Fig. 29: Etiquetado de las imágenes, considerando el número y clase a la que pertenecen	107
Fig. 30: Imágenes del subconjunto test, no etiquetadas.....	108
Fig. 31: Segmentación de las imágenes por el método de umbralización	109
Fig. 32: Representación del proceso de segmentación de las imágenes por el método de umbralización	110
Fig. 33: Código para obtener la región de interés (ROI).....	111
Fig. 34: Imagen recortada de acuerdo con el área de interés (ROI) y redimensionada a un tamaño de 128x240 pixeles.....	111
Fig. 35: Distribución de imágenes previas a la implementación de la red propuesta.....	112
Fig. 36: Diseño de la red propuesta para el modelo propio. Fuente: Elaboración propia ..	112
Fig. 37: Arquitectura CNN -Extracción de características.....	113
Fig. 38: Implementación del clasificador propuesto.....	113

Fig. 39: Compilación del modelo	114
Fig. 40: Técnica de aumento de datos aplicado a los datos de entrenamiento.	114
Fig. 41: Ejecución del entrenamiento del modelo.....	115
Fig. 42: Implementación de la red pre entrenada VGG16	116
Fig. 43: Implementación de la red pre entrenada InceptionV3	117
Fig. 44: Implementación de la red pre entrenada MobileNet	117
Fig. 45: Prueba realizada con el modelo propio, para clasificar una imagen de café sana y con deficiencia de magnesio.....	118
Fig. 46: Prueba realizada con el modelo propio, para clasificar una imagen con deficiencia de hierro y deficiencia de nitrógeno.	118

Resumen

El objetivo del estudio fue proponer un método de clasificación automática de deficiencias nutricionales en hojas de café mediante procesamiento de imágenes digitales y aprendizaje profundo, con el fin de superar las limitaciones de tiempo y costo asociadas al análisis químico foliar. Para lograrlo, se llevó a cabo una caracterización visual de las deficiencias nutricionales del café con la asesoría de un ingeniero agrónomo experto. Posteriormente, se construyó un prototipo para la adquisición de imágenes, estas se capturaron siguiendo un protocolo específico y se elaboró un conjunto de datos con un total de 4032 imágenes, de las cuales se consideró 666 para el subconjunto pruebas y 3366 para entrenamiento y validación. Las imágenes fueron procesadas mediante técnicas de binarización y segmentación, utilizando el método Otsu y regiones de interés (ROI) para recortar la imagen de acuerdo a su tamaño. A continuación, se implementó un modelo de red neuronal convolucional de aprendizaje profundo, el cual se comparó con los modelos VGG16, InceptionV3 y MobileNet. Los resultados mostraron que el modelo propio superó a los demás modelos, alcanzando una exactitud del 99.11%, MobileNet 98.22%, InceptionV3 95.57% y VGG16 un 95.39%. Estos hallazgos sugieren que la combinación de técnicas de procesamiento de imágenes digitales con aprendizaje profundo es altamente efectiva para la clasificación de deficiencias nutricionales en hojas de café. Como conclusión, se destaca la viabilidad de este enfoque para automatizar el proceso de diagnóstico, lo que podría beneficiar significativamente a los agricultores al permitirles contar con una herramienta eficiente para reconocer las deficiencias nutricionales en hojas de café.

Palabras Clave: Clasificación automática, Procesamiento de imágenes, Aprendizaje profundo, Deficiencias nutricionales, Café.

Abstract

The objective of the study was to propose a method for automatic classification of nutritional deficiencies in coffee leaves using digital image processing and deep learning, in order to overcome the time and cost constraints associated with chemical leaf analysis. To achieve this, a visual characterisation of nutritional deficiencies in coffee plants was carried out with the advice of an expert agronomist. Subsequently, a prototype for image acquisition was built, images were captured following a specific protocol and a dataset with a total of 4032 images was produced, of which 666 were considered for the test subset and 3366 for training and validation. The images were processed by binarization and segmentation techniques, using the Otsu method and regions of interest (ROI) to crop the image according to its size. A deep learning convolutional neural network model was then implemented and compared with the VGG16, InceptionV3 and MobileNet models. The results showed that the proprietary model outperformed the other models, achieving an accuracy of 99.11%, MobileNet 98.22%, InceptionV3 95.57% and VGG16 95.39%. These findings suggest that the combination of digital image processing techniques with deep learning is highly effective for the classification of nutritional deficiencies in coffee leaves. In conclusion, the feasibility of this approach to automate the diagnostic process is highlighted, which could significantly benefit farmers by providing them with an efficient tool to recognize nutritional deficiencies in coffee leaves.

Keywords: Automatic classification, Image processing, Deep learning, Nutritional deficiencies, Coffee plant.

I. INTRODUCCIÓN

1.1. Realidad problemática.

La agricultura desempeña un rol trascendental dentro de la economía y la sociedad, ya que no solo proporciona materias primas y alimentos, sino también oportunidades de empleo para gran parte de la población. Llegando a representar el 4% del producto bruto interno (PBI) en países desarrollados y más del 25% en los países subdesarrollados, lo que indica que este sector es de dos a cuatro veces más efectivo en el aumento de los ingresos económicos en comparación con otros sectores [1]. Sin embargo, el desarrollo económico impulsado por la agricultura cada día se ve más afectado por diversas causas, tales como el cambio climático, eventos meteorológicos, plagas y conflictos. Son eventos que tienen un resultado perjudicial en la población, ya que ocasionan el aumento de los precios en los alimentos o amenazan el bienestar de las personas. Como resultado de ello, la desnutrición, la obesidad y el sobrepeso están en incremento cada día y están convirtiéndose en las causas principales de muerte a nivel mundial [2].

A pesar de ello, los productos agrícolas y alimentarios de Sudamérica han tenido reconocimiento como uno de los más sobresalientes del planeta en los últimos años, dotando de alimentos a una población en vasto crecimiento, impulsando la economía, posibilitando la urbanización, produciendo exportaciones de materias primas y ayudando a reducir el hambre y la pobreza a nivel del mundo. Entre los cuales, el ábaco, el cáñamo, las nueces de palma, las aceitunas, el agave, el aguacate, el ajo, los chiles, los albaricoques, las alcachofas, el algodón, las almendras, el alpiste, el anís, el arroz, los arándanos, las avellanas, la caña de azúcar, los plátanos, patatas, berenjenas, café, calabazas, batatas, castañas, caucho, etc., son ejemplos representativos que se pueden destacar [3].

Como es conocido, el café es uno de los productos de exportación más destacados

del Perú. Según el Ministerio de Agricultura y Riego, en la actualidad se cultivan aproximadamente 425.400 hectárea en quince regiones, distribuidas en noventa y cinco provincias y cuatrocientos cincuenta distritos, de las cuales Amazonas, Cajamarca, Huánuco, San Martín, Junín, Pasco y Cusco tienen el 91% del total de productores y superficie cultivable [3]. A pesar de la considerable producción de café en el país, de acuerdo con la Organización Internacional del Trabajo [4], Perú se encuentra en el noveno lugar como productor de café a nivel de todo el mundo, siendo superado por superado en primer lugar por Brasil, segundo lugar Vietnam, seguido por Colombia, Indonesia, Honduras, Etiopía, India y Uganda, tal como se detalla a continuación.

Tabla 1: Principales productores de café del mundo, tomado de cinco cosechas.

Año	2013/ 2014	2014/ 2015	2015/ 2016	2016/ 2017	2017/ 2018
1. Brasil	54698	52299	25426	56714	51000
2. Vietnam	27610	26500	28737	25540	29500
3. Colombia	12163	13339	14009	14634	14000
4. Indonesia	12818	10862	12535	11491	10902
5. Honduras	4583	5268	5786	7457	8349
6. Etiopía	6427	6575	6714	7297	7650
7. India	5075	5450	5800	5200	5840
8. Uganda	3633	3744	3650	4962	5100
9. Perú	4106	2883	3304	4223	4280
10. México	3916	3591	2903	3781	4000

Nota: Tomado de Organización Internacional del Trabajo (OIT) [4].

No obstante, las consecuencias económicas provocadas por el COVID-19 y el conflicto ruso-ucraniano, el café peruano sigue gozando de una alta demanda en el mercado externo, aportando un fuerte capital social y económico para más de 223,000 familias dedicadas a este cultivo, involucrando cerca de dos millones de personas y representando un tercio de los empleos en nuestro país [5]. Según el reporte estadístico emitido por la Unidad de Inteligencia Comercial - SSE, las exportaciones de café en el año 2021 fueron de 761.9 millones de dólares en comparación a los 644.0 millones del año anterior, mostrando un crecimiento del 18,3%. Para el mes de diciembre se registraron envíos por 144.6 millones de dólares, lo que representa un incremento del 91% en comparación con los 75.7 millones del mismo período del año anterior [6].

Tabla 2: Valores de exportación del café en los últimos años expresados en dólares

Mes	2019	2020	2021	Val % 2020/2019	Val % 2021/2020
Enero	42,161,376	29,028,137	31,258,371	-31.1%	7.7%
Febrero	18,560,920	14,605,904	13,079,118	-21.3%	-10.5%
Marzo	7,765,722	5,477,621	8,815,590	-29.5%	60.9%
Abril	8,840,565	7,775,701	8,017,706	5.6%	3.1%
Mayo	12,944,644	17,180,063	11,113,705	32.7%	-35.3%
Junio	27,351,837	24,100,330	28,454,897	-11.9%	18.1%
Julio	64,062,970	85,862,047	55,125,850	-8.1%	-6.3%
Agosto	82,505,394	89,934,214	67,863,381	9.0%	-24.6%
Septiembre	106,238,363	117,273,755	126,739,516	10.4%	8.1%
Octubre	93,343,695	110,812,837	133,143,263	18.7%	20.2%
Noviembre	86,783,752	93,336,700	133,703,281	7.6%	43.2%
Diciembre	84,905,865	75,672,491	144,603,487	-10.9%	91.1%
Total	634,865,103	644,059,834	761,858,165	1.4%	-
Var. % anual	-	1.4%	18.3%		

Nota: Tomado de la Unidad de Inteligencia Comercial -SSE

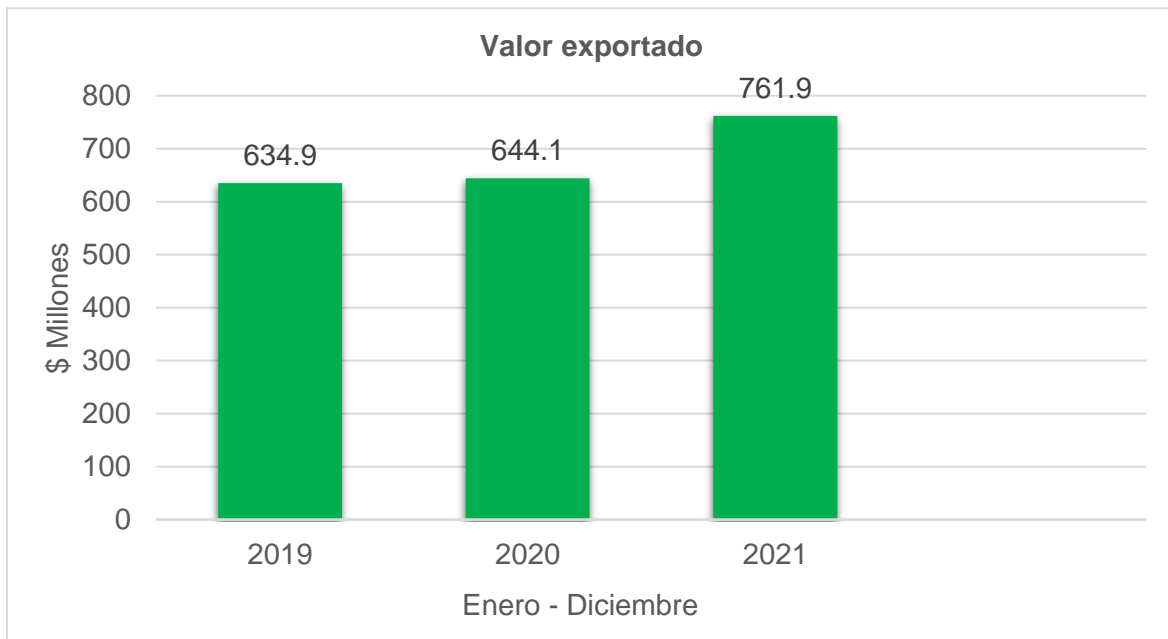


Fig. 1: Valores FOB de exportación del café en los últimos años expresados en dólares [6].

Tal como se ha evidenciado, el café posee una gran importancia social y económica. Uno de los principales desafíos para los caficultores es la desnutrición de las plantas, lo cual las hace más susceptibles a diversas enfermedades y afecta la calidad y productividad de los cultivos. Esta situación conduce a la escasez del producto. Identificar la desnutrición de manera precisa representa un desafío significativo, ya que en muchas ocasiones los agricultores carecen de los conocimientos necesarios y reciben poca asistencia técnica. En su mayoría, el diagnóstico se realiza mediante simples muestras visuales, las cuales pueden ser inexactas dependiendo de la experiencia de cada experto agrónomo. Para obtener un diagnóstico más preciso, es necesario enviar una gran cantidad de hojas de muestra de las plantas de café a un laboratorio para realizar un análisis químico foliar. Aunque este método es más preciso, requiere mucho tiempo y tiene un alto costo, lo que representa una dificultad para los agricultores de pequeña y mediana escala de producción [7].

Con el avance de las innovaciones tecnológicas, han surgido nuevas prácticas agrícolas conocidas como agricultura de precisión, las cuales están ganando cada vez más importancia en la actualidad. Estas prácticas están destacando de manera constante gracias a la amplia aceptación tanto de los agricultores como de las organizaciones que respaldan la investigación agraria. La agricultura de precisión se fundamenta en la recopilación de información sobre los cultivos con el fin de identificar las mejores medidas para abordar los problemas en el desarrollo de las plantas. Esto se logra mediante el análisis de datos recopilados a través de imágenes, sensores, GPS y otros mecanismos [8].

Con el firme propósito de ayudar a mitigar los graves problemas que enfrentan los agricultores, numerosos investigadores han unido sus esfuerzos para proponer diversas técnicas y métodos de procesamiento de imágenes, aprendizaje automático y aprendizaje profundo como posibles soluciones. Entre estas técnicas, se destaca la capacidad para identificar de manera anticipada la presencia de plagas y enfermedades en las plantas. Asimismo, mediante el análisis de imágenes satelitales o capturadas por drones, es posible monitorear el crecimiento de los cultivos en grandes extensiones. Utilizando técnicas de teledetección, se puede evaluar la humedad en el suelo para optimizar el riego. Además, gracias al análisis de imágenes, es posible clasificar automáticamente frutas y hortalizas según su tamaño, forma, color y calidad. Algunas muestras representativas de estas investigaciones son:

En su investigación, titulada "Rice nitrogen nutrition monitoring classification method based on the convolution neural network model: Direct detection of rice nitrogen nutritional status" [9], realizada en la Universidad de Nanchang en China, se abordó el problema de la falta de nitrógeno, un factor que afecta el rendimiento del arroz. Por lo general, los expertos realizan visitas a las tierras de cultivo para observar el desarrollo de las plantas y, en función de los resultados, aplican la cantidad adecuada de fertilizante nitrogenado. Sin embargo, este

método requiere mucho trabajo y tiempo. Por tal razón, se propuso desarrollar un sistema automatizado para la monitorización del arroz. En primer lugar, las imágenes se capturaron con una cámara Canon EOS80D (Canon Inc., Tokio, Japón), que tenía una resolución de 5184 x 3456 píxeles, pero debido al gran tamaño de los datos, se redujeron a 1024 x 1024. Luego, se construyó el conjunto de datos de la siguiente manera: 540 imágenes de nitrógeno sobrecargado, 270 imágenes de nitrógeno óptimo y 270 imágenes de nitrógeno estresado. En la etapa de procesamiento, las imágenes fueron redimensionadas a 224 x 224 píxeles mediante interpolación bilineal, posteriormente se convirtieron al tipo tensor y se normalizaron cada canal en un rango de 0 a 1. Para la clasificación se utilizaron las arquitecturas de Google Net, DLANet, ResNet y MobileNet. Se empleó la función de entropía cruzada, el optimizador SGD, un tamaño de lote de 32 y 100 épocas. Los resultados mostraron que la arquitectura Google Net logró una precisión del 95.99%, seguida por ResNet con un 86.27%, MobileNet con un 83.64%, y DLANet con un 75.54%. Se concluyó que una solución efectiva y confiable para el monitoreo automatizado del estado de nutrición del nitrógeno del arroz es el uso de CNN, como el modelo Google Net. Esto podría tener beneficios significativos en la eficiencia y precisión de la gestión nutricional de los cultivos de arroz.

La investigación titulada "Deep Ensemble Mobile Application for Recommendation of Fertilizer Based on Nutrient Deficiency in Rice Plants Using Transfer Learning Models" [10], realizada en el Departamento de Ciencias e Ingeniería Informática del V R Siddhartha Engineering College en India, abordó el problema de la disminución en la productividad del arroz debido al conocimiento limitado de los agricultores sobre la selección de semillas, abonos e infraestructuras de riego. Esto ha llevado a una mala nutrición de las plantas y a cosechas deficientes. Por lo tanto, se propuso implementar cinco arquitecturas de aprendizaje por transferencia utilizando los modelos InceptionV3, Xception, VGG16, Resnet50 y MobileNet para detectar deficiencias nutricionales. Se utilizó el conjunto de datos de Kaggle, dividiendo 1157 imágenes a color RGB entre las tres categorías de deficiencias de fósforo,

nitrógeno y potasio. Debido a la falta de imágenes, se aumentaron los datos para obtener 3005 imágenes, de las cuales 2686 se utilizaron para entrenamiento, 335 para validación y 335 para pruebas. Luego, los datos se redimensionaron a 150 por 150 píxeles y se normalizaron en un rango de 0 a 1. Se agregaron capas densas, una capa softmax y una capa pooling a las arquitecturas de clasificación. Estas se entrenaron utilizando el modelo preentrenado Imagenet, con 30 épocas y un tamaño de lote de 32. Se utilizó la entropía cruzada como función de pérdida y el optimizador Adam. Los resultados mostraron una precisión del 93% con la arquitectura MobileNetV2, 94% con VGG16, 96% con ResNet50, 97% con InceptionV3 y 97% con Xception. Además, los modelos que combinaron arquitecturas, como InceptionV3 + Xception, VGG16 – ResNet50 y MobileNet + InceptionV3, alcanzaron una precisión del 98%, 96% y 95%, respectivamente. Posibilitar la identificación anticipada y precisa de enfermedades y carencias nutricionales en los cultivos a través de modelos de aprendizaje profundo puede tener un impacto importante en la agricultura y la seguridad alimentaria, al permitir que los agricultores tomen medidas preventivas y correctivas a tiempo para mejorar tanto la producción agrícola como la seguridad alimentaria.

La investigación titulada "An automated segmentation and classification model for banana leaf disease detection" [11], llevada a cabo en la Universidad Autónoma de Zacatecas, México, abordó el desafío de detectar y clasificar enfermedades en las plantas, un procedimiento complicado que requiere una amplia recolección de datos, conocimientos especializados y técnicas avanzadas de procesamiento de imágenes digitales. Para abordar este problema, se propuso un método para la segmentación y clasificación automáticas de enfermedades en hojas de plátano. Se tomaron imágenes de hojas de plátano afectadas por cinco enfermedades principales: fusarium, sigatoka negra, sigatoka amarilla, marchitez por Xanthomonas y enfermedad del cogollo del plátano, generando un total de 9,000 imágenes. Se aplicaron técnicas de preprocesamiento para mejorar la calidad de las imágenes y recuperar la información más importante. Esto incluyó el uso de modelos de espacio de color

HSI y HSV en lugar de RGB. Luego, se realizó la segmentación para identificar con precisión las áreas afectadas de las hojas utilizando la técnica TGVFCMS (Total Generalized Variation y Fuzzy C-Means Segmentation). La imagen segmentada se convirtió en una matriz de 32 x 32 en escala de grises, que se utilizó como entrada a la red neuronal convolucional (CNN). Los resultados mostraron que al combinar la segmentación precisa de las áreas afectadas en hojas de plátano con la clasificación mediante redes neuronales convolucionales (CNN), se logró una precisión del 93.45%. El sistema propuesto tiene el potencial de ser una herramienta útil para que los agricultores detecten y manejen rápidamente las enfermedades de las plantas de banano, lo cual puede contribuir significativamente a mejorar la salud de los cultivos y la productividad agrícola.

En el estudio "Coffee Disease Visualization and Classification" [12], llevado a cabo en el Departamento de Ingeniería Informática del Instituto Nacional de Tecnología de Kumho, Gumi, Corea del Sur, se abordó el desafío de mejorar la confiabilidad de los modelos de clasificación de enfermedades en plantas, que a menudo se consideran "cajas negras". Se propuso una metodología para visualizar las enfermedades del café utilizando diferentes enfoques de visualización con el fin de comprender mejor cómo el modelo de aprendizaje automático realiza las clasificaciones. Se utilizaron 1560 fotografías de hojas de café sanas y enfermas como conjunto de datos. Se emplearon dos enfoques de clasificación: uno guiado y otro ingenuo, y tres métodos de visualización: Grad-CAM, Grad-CAM++ y Score-CAM. Se utilizó una CNN ResNet con entropía cruzada como función de pérdida y RMSprop como optimizador, con una tasa de aprendizaje inicial de 0.0001 para el enfoque ingenuo. El tamaño del lote fue de 32. Se comparó la distribución de las predicciones con la realidad para evaluar la confiabilidad del modelo. Para el enfoque guiado, se realizó un cambio de fondo para facilitar el análisis, se utilizó la red pre entrenada U2net para generar un enmarcado a nivel de píxeles entre la imagen original y la segmentada, y luego se aplicaron los métodos de visualización. Los resultados mostraron que el enfoque guiado alcanzó una precisión del 98

%, en comparación con el 77 % del enfoque ingenuo. Esta diferencia significativa demuestra la utilidad de la visualización en los métodos de detección de enfermedades para comprender mejor la eficacia del modelo. El estudio contribuye significativamente al campo de la detección de enfermedades en la agricultura al proporcionar una herramienta para evaluar y comprender mejor el funcionamiento de los modelos de aprendizaje automático. Esto permite que estos métodos de detección de enfermedades dejen de ser percibidos como "cajas negras" y se reconozcan como avances importantes para la ciencia agrícola.

En el estudio titulado "Plant Diseases Classification using Machine Learning", llevado a cabo en la Universidad Malaysia Perlis en Malasia [13], se abordó el problema de las enfermedades de las plantas, que impactan la calidad y productividad de los cultivos, y pueden llevar a la escasez de alimentos. Se planteó la implementación de un modelo de Máquina de Aprendizaje Extremo (ELM), que es un algoritmo utilizado en el campo del aprendizaje automático, con el fin de clasificar las enfermedades de las plantas según la calidad de sus hojas. Para este propósito, se utilizaron hojas de tomate obtenidas del conjunto de datos Plant Village de Kaggle, considerando 10 clases y 1000 imágenes por cada clase. Inicialmente, el tamaño de las imágenes se ajustó a 256 x 256 píxeles y se realizó una conversión de color a HSV. Luego, se aplicó un filtro gaussiano para distinguir los bordes y se llevaron a cabo operaciones morfológicas de cierre para eliminar el ruido. Además, se utilizó la textura Haralick para extraer características de los colores. Se dividió los datos en un 70% para entrenamiento y un 30% para validación. Se utilizó el Extreme Learning Machine (ELM) durante la fase de clasificación, el cual incluía diversas funciones de activación como lineal, relu, sigmoid y tanh. Se obtuvieron resultados que indicaban una exactitud del modelo en un rango de 69% a 84.94%. Identificar enfermedades presentes en las hojas de tomate fue el principal objetivo del estudio. Se recomienda considerar la opción de extender esta investigación a otros cultivos como arroz, maíz y trigo, lo que ayudaría a comprender y detectar enfermedades en varios tipos de plantas.

Se realizó la investigación titulada "Automatic Detection of Citrus Fruit and Leaves Diseases Using Deep Neural Network Model, en la Zayed University" [14], en la Zayed University, Dubái, en los Emiratos Árabes Unidos. En los últimos años, se han realizado múltiples estudios acerca del diagnóstico de enfermedades en plantas, como por ejemplo en los cultivos de cítricos, empleando distintas técnicas de aprendizaje automático y detección de patrones. Uno de los desafíos identificados es la dificultad para aumentar las tasas de precisión en la clasificación de enfermedades en cultivos, así como la degradación del rendimiento debido a la selección inadecuada de parámetros y capas en los modelos de redes neuronales. Por lo tanto, se planteó la creación de un método basado en aprendizaje profundo para identificar enfermedades en las hojas de cítricos de manera automatizada. El conjunto de datos utilizado fue tomado del repositorio Plant Village, con un total de 2293 imágenes distribuidas en cuatro grupos de enfermedades: mancha negra, cancro, costra, enverdecimiento, melanosis, y un grupo de frutos y hojas saludables. Se utilizó el modelo CNN para extraer características discriminativas mediante la integración de múltiples capas. En primer lugar, se dividió la imagen en píxeles y se representó como una matriz tridimensional (7 x 8 x) con capas rojas, verdes y azules. Para obtener las características de entrada, se agregó una capa convolucional después de la interpretación de la imagen. Posteriormente, se realizó una operación convolucional entre la matriz del filtro y la matriz de la imagen. Se aplicó una segunda capa convolucional para extraer las características de nivel alto, y se utilizó un segundo maxpooling para disminuir la dimensión del tamaño y facilitar la detección. Luego, se utilizó una capa plana para convertir una matriz de características agrupadas en una columna o vector de características. Para la clasificación, después de la entrada de la capa plana, se utilizó una nueva capa con activación softmax para calcular las probabilidades de las distintas enfermedades en las hojas o los frutos de cítricos. Los resultados demostraron que el modelo CNN sugerido tenía un alto rendimiento y una precisión del 94.55%. El modelo propuesto para la identificación de enfermedades de las hojas demostró que es capaz de distinguir entre hojas y frutos sanos y enfermos, lo que indica que

es un modelo muy eficaz para detectar enfermedades en hojas y frutos cítricos.

Realizaron la investigación titulada "Classification and Detection of Nutritional Deficiencies in Coffee Plants Using Image Processing and Convolutional Neural Network (CNN)" [15], en la AMA University, Filipinas. Resulta complicado para los agricultores identificar visualmente las deficiencias nutricionales en las hojas de café, lo que puede llevar a intervenciones inadecuadas o ineficaces en las plantas, ocasionando una disminución del rendimiento de los cultivos y pérdidas económicas para los productores. Entonces, se propusieron desarrollar un modelo que emplee redes neuronales convolucionales y procesamiento de imágenes para identificar y detectar deficiencias nutricionales en las hojas de café automáticamente. Para el estudio, se utilizaron mil imágenes de hojas de café con deficiencias de zinc, magnesio, potasio, fósforo, hierro, nitrógeno y boro, las cuales fueron capturadas utilizando un Raspberry Pi 4 y cámaras Logitech, mediante un prototipo diseñado para la adquisición de las imágenes. Posteriormente, se procesaron estas imágenes convirtiéndolas de RGB a escala de grises y luego a valores binarios para la umbralización y segmentación. Después de eso, las imágenes se transformaron a formato vectorial y se utilizó una red neuronal convolucional (CNN) para clasificar y visualizar los resultados basados en la imagen de entrada. Los resultados mostraron que la aplicación de estas técnicas resulta una alternativa efectiva, logrando una precisión en las deficiencias de Boro (B) del 92,5%; Calcio (Ca) 90,2%; Hierro (Fe) 90,2%; Potasio (K) 90,00%; Nitrógeno (N) 93,7%; Fósforo (P) 93,0%; Magnesio (Mg) (S) 90,1%; y Zinc (Zn) 92,20% respectivamente, con una exactitud global del 91,49%. Por lo consiguiente, se ha demostrado que el procesamiento de imágenes y las redes neuronales convolucionales pueden clasificar e identificar deficiencias nutricionales en las plantas de cafeto.

En su trabajo "Multi-class Image Classification Using Deep Learning Algorithm" [16], realizado en la Universidad de Menouf, Menoufia, Egipto, se abordó el desafío de la

clasificación de imágenes, un problema complejo que requiere algoritmos de aprendizaje profundo con un alto coste de hardware y tiempo de entrenamiento. El objetivo del estudio fue evaluar la capacidad de las redes neuronales convolucionales (CNN) para clasificar imágenes de múltiples clases. Las CNN fueron entrenadas primero con el conjunto de datos ILSVRC y luego probadas con un conjunto de datos más pequeño, el PASCAL VOC de 2007. El conjunto de datos que se utilizó incluía cuatro clases: pájaro, gato, TV y monitor. Se generaron tres archivos CSV mediante programación en Python, cada uno con 702 imágenes, que contenían los índices de las imágenes elegidas aleatoriamente en tres subconjuntos iguales (G1, G2 y G3). Estos archivos incluían el nombre de la imagen, la clase a la que pertenecían y el número de archivo. Posteriormente, estos archivos se separaron en tres carpetas con el mismo nombre de archivo al que pertenecían. Para el entrenamiento y la validación, se utilizó la técnica de validación cruzada. En primer lugar, se emplearon los subconjuntos G1 y G2 para el entrenamiento y el subconjunto G3 para la validación. Luego, se utilizaron los subconjuntos G1 y G3 para el entrenamiento y el subconjunto G2 para la validación. Finalmente, se emplearon los subconjuntos G2 y G3 para el entrenamiento y el subconjunto G1 para la validación. Los resultados obtenidos mostraron que el modelo CNN con aprendizaje profundo alcanzó una precisión del 90.81 %, en comparación con el 65.27 % de la clasificación mediante vectores de soporte. Estos hallazgos demuestran que las CNN de aprendizaje profundo son una forma eficaz y sencilla de clasificar imágenes multiclase.

En el trabajo de investigación "Hemp Disease Detection and Classification Using Machine Learning and Deep Learning" [17], realizado en la Universidad Estatal de San José, CA, USA, se abordó el impacto de las enfermedades en el crecimiento y producción del cáñamo, una planta con valor industrial y medicinal. Se dispuso la implementación de un modelo de máquina de vectores de soporte (SVM) y tres modelos de aprendizaje profundo como una forma para mitigar esta situación. Se recopilaron manualmente las imágenes utilizadas de diversas fuentes, considerando 450 imágenes por cada clase. Después, se

redimensionaron estas imágenes a 512 x 512 píxeles y se cambió su resolución al 70%. Se usaron técnicas de aumento de datos, incluyendo desplazamiento, rotación y volteo, para crear un banco de datos que contenía veinte mil imágenes de cáñamo con diversas características. El conjunto de datos se dividió en tres categorías: Entrenamiento, validación y pruebas. Se utilizaron dos enfoques para la clasificación de enfermedades del cáñamo. El primer enfoque consistió en entrenar un modelo SVM utilizando diferentes características extraídas de las imágenes, mientras que el segundo enfoque se basó en tres modelos de aprendizaje profundo: VGG16, AlexNet e Inceptionv3. Aunque los modelos de aprendizaje profundo mostraron mejores resultados que el modelo SVM, enfrentaron problemas de sobreajuste debido a limitaciones de recursos informáticos. Por lo tanto, se optó por desarrollar tres modelos de aprendizaje profundo utilizando transferencia de aprendizaje y modelos pre-entrenados, los cuales demostraron un excelente rendimiento y precisión. Los resultados obtenidos fueron los siguientes: un 75% de exactitud con el modelo SVM, 90.76% de precisión con la AlexNet modificada con clasificador Random Forest, 95.2% con Inception modificada con clasificador SVM, y 98.5% con VGG16 modificada utilizando aprendizaje por transferencia. Estos hallazgos indican que el desarrollo de modelos ensamblados es muy efectivo en situaciones en las que hay pocos datos y recursos informáticos.

Realizaron la investigación, " New method for subject identification based on palm print" [18], en la Universidad Señor de Sipán de Perú. Actualmente, existen diversos sistemas automatizados para la identificación de personas, como los sistemas biométricos convencionales que detectan huellas dactilares, voz, facial, contorno de la mano, entre otros. Sin embargo, algunos de estos métodos biométricos han sido clonados para usos ilegales, lo que ha motivado estudios para encontrar alternativas más seguras. Entre estas alternativas, se ha evidenciado que los sistemas de detección de huellas palmares son la mejor manera de lograr la identificación y protección de una persona. Sin embargo, en sectores donde se requieren medidas adicionales de seguridad, como áreas de alta seguridad y laboratorios de

investigación, se necesitan imágenes de huellas dactilares adquiridas por contacto para garantizar la presencia física de la persona. Esto ha generado nuevos requisitos para la identificación de sujetos. Por esta razón, los investigadores decidieron crear un nuevo conjunto de datos de huellas palmares adquiridas por contacto y un método de reconocimiento de patrones para la identificación. Se tomaron 100 imágenes de 1275 x 1754 píxeles de 50 personas, con 100 imágenes por mano, utilizando el escáner incluido en la impresora HP L415. Inicialmente, se redujo el tamaño de las imágenes y luego se convirtieron a escala de grises. A continuación, se segmentaron utilizando el método Otsu y se aplicaron operaciones morfológicas de apertura, como erosión y dilatación, para eliminar manchas blancas del fondo. La imagen resultante se utilizó como máscara para conservar solo los valores de los píxeles dentro de la silueta de la mano. Posteriormente, todas las imágenes se redimensionaron a 1233 x 1392 píxeles para tener el mismo tamaño. Se empleó la validación cruzada, utilizando 80 muestras para el entrenamiento y 20 muestras para la validación en cada período. Para la clasificación, se utilizó la arquitectura VGG16, reduciendo las imágenes a 24 x 224 píxeles. Los resultados mostraron que el modelo VGG16 propuesto alcanzó una exactitud del 100% y una puntuación F1 de 1. Estos resultados demuestran la efectividad del método sugerido para identificar sujetos utilizando la impresión palmar. Con el continuo desarrollo de nuevos sistemas de seguridad, la huella de la palma se vuelve cada vez más una opción viable para este propósito.

En su investigación titulada "A Comparative Study of Deep CNN in Forecasting and Classifying the Macronutrient Deficiencies on Development of Tomato Plant" [19], realizada en la Universidad Dong-A, Corea, se abordó la principal causa de enfermedades en las plantas que afectan el rendimiento de los cultivos: la falta de nutrientes. Para abordar este problema, se propuso implementar modelos de redes neuronales artificiales para clasificar deficiencias nutricionales en plantas de tomate, centrándose en deficiencias de calcio, potasio y nitrógeno. Se utilizó la cámara de un smartphone para capturar imágenes, obteniendo un

total de 571, de las cuales el 80% se asignó para entrenamiento y el 20% para validación. Se emplearon dos modelos de clúster de aprendizaje automático para la clasificación: aprendizaje supervisado (Inception-ResNetv2) y aprendizaje no supervisado (Autoencoder), ambos basados en CNN y utilizados para entrenar y evaluar predicciones. Para la arquitectura del modelo Inception-ResNet V2, se consideró un tamaño de imágenes de entrada de 299 x 299 píxeles y después de la primera capa de filtro se aplicó una reducción del tamaño de la capa a 35 x 35 píxeles. Se utilizaron capas de normalización por lotes, Relu como función de activación y una tasa de aprendizaje de 0.0001. Para la arquitectura del modelo Autoencoder, se consideraron imágenes de entrada de 149 x 149 píxeles y se aplicó una reducción a 35 x 35 píxeles. Se utilizaron núcleos de convolución de 1 x 1 y 3 x 3 con zancadas de 1 o 2, y se empleó la función Relu como función de activación en todas las capas de convolución. Los resultados mostraron una tasa de validación del 87.27% para el modelo Inception-ResNetv2, superior al 79.09% obtenido por el modelo Autoencoder, a pesar de tener el mismo número de parámetros. En el campo del cultivo, la predicción precisa de deficiencias nutricionales puede contribuir a mitigar los factores que conducen a las patologías de las plantas, lo que resalta la importancia de una evaluación precisa y un pronóstico temprano de deficiencias de nutrientes en el sector agrícola.

Realizaron la investigación "Automatic detection of nutritional deficiencies in coffee tree leaves through shape and texture descriptors" [20], en San Miguel de Naranjas y La Palma Central, provincia de Jaén, Cajamarca, Perú. Aunque hay muchas investigaciones sobre la identificación de deficiencias nutricionales en cultivos populares como maíz, arroz y hortalizas mediante el análisis de imágenes digitales, se ha observado una falta de estudios centrados en el café. Así que se decidió abordar esta brecha en la literatura científica. Primero se obtuvo el conjunto de imágenes del entorno y se procesó utilizando técnicas de umbralización y el método de Otsu. Después, se extrajeron las características de forma y textura de las hojas utilizando la matriz de co-ocurrencia de niveles de gris (GLCM). Estos

datos se emplearon para entrenar clasificadores supervisados como Naive Bayes, k-vecinos cercanos y redes neuronales para reconocer deficiencias nutricionales en nuevas imágenes. Los resultados mostraron que utilizando el descriptor BSM, se obtuvo un 46.09% de exactitud en promedio con el clasificador k-vecinos cercanos (KNN), 65.05% de exactitud con el clasificador Naive Bayes y 59.11% utilizando el clasificador de redes neuronales. Utilizando la GLCM, se logró un 46.84% de exactitud con el clasificador k-vecinos cercanos (KNN), 46.09% de exactitud con el clasificador Naive Bayes y 49.81% utilizando el clasificador de redes neuronales. Debido a la variedad de síntomas visibles que pueden presentarse, detectar deficiencias nutricionales en las hojas de café mediante el procesamiento de imágenes es una tarea difícil. Como resultado, es fundamental fomentar una variedad de investigaciones y la creación de productos computacionales que ayuden a los especialistas en este campo.

El café es uno de los cultivos de gran importancia del Perú y el mundo, ya que hace una contribución significativa a la economía global y representa el medio de vida de millones de agricultores. Sin embargo, la calidad y productividad de los cultivos de café cada vez se ven comprometidas especialmente por las deficiencias de nutrientes en las plantas, que pueden ser difíciles de identificar y manejar de manera efectiva. En este contexto, el uso de técnicas avanzadas de procesamiento de imágenes y aprendizaje profundo ofrece una oportunidad única para resolver este problema de forma precisa y automatizada. Aunque hoy en día existen diversas tecnologías aplicables a la agricultura, hay diversos trechos que no han sido cubiertos a cabalidad o si ya existen tienen elevados costos, de este modo representa un factor importante sobre todo para los pequeños agricultores. Por lo que creemos que el presente estudio contribuirá al avance tecnológico, al proporcionar un método de clasificación automática de deficiencias nutricionales mediante procesamiento de imágenes y aprendizaje profundo. Además, permitirá contar con información de rendimiento en eficiencias de diversos algoritmos, las cuales servirán como fuentes de información para

futuros trabajos. En cuanto al ámbito social será dar a conocer los métodos y técnicas existentes de visión artificial aplicables a la agricultura, así mismo se brindará a los agricultores la posibilidad de poder contar con instrumento que les ayude a clasificar en etapa temprana las deficiencias nutricionales en cultivos de café y de esa manera puedan tomar acciones preventivas y correctivas a tiempo y así aseguren su producción.

1.2. Formulación del problema

¿Cómo clasificar de manera automática deficiencias nutricionales del café mediante el análisis de las hojas?

1.3. Hipótesis

Mediante procesamiento de imágenes digitales y aprendizaje profundo se podrán clasificar de manera automática las deficiencias nutricionales del café.

1.4. Objetivos

Objetivo general

Desarrollar un método para la clasificación automática de deficiencias nutricionales en hojas de café mediante procesamiento de imágenes digitales y aprendizaje profundo.

Objetivos específicos

- Caracterizar las deficiencias nutricionales de la hoja del café a nivel visual.
- Elaborar el conjunto de datos de acuerdo con las características de las deficiencias nutricionales de las hojas del café.
- Realizar el procesamiento digital a las imágenes de las hojas del café.
- Implementar un modelo propio de aprendizaje profundo para clasificar las deficiencias nutricionales de las hojas del café y comparar los resultados con otros modelos.

1.5. Teorías relacionadas al tema

1.5.1. El cafeto

Familia de las rubiáceas, de nombre científico *Coffea*, es un arbusto perenne que generalmente se cultiva en zonas tropicales de climas húmedos y cálidos. Tiene una altura aproximada de entre 2 y 5 metros y en algunos casos particulares llegando a 12 metros, dependiendo mucho de la zona y la variedad. En cuanto a su historia, fue mencionada por primera vez en el año 1592 en África y en 1737 fue designada como un nuevo género denominado *coffea* por el naturalista Carlos Linneo. La altitud óptima para el cultivo del cafeto abarca entre los 700 msnm y 1700 msnm. En cuanto a su temperatura, se ubica entre los 18°C y 22°C. Por encima o debajo de estas condiciones desfavorecen y dificultan el correcto desarrollo de la planta y calidad del producto. En Perú, la especie que más se cultiva es la arábica en la variedad de café caturra, típica y bourbon [21].



Fig. 2 :Plantación de cafeto

1.5.2. Deficiencias nutricionales

Para llevar a cabo sus funciones esenciales, las plantas dependen de la absorción de

catorce nutrientes minerales del suelo a través de sus raíces. Entre estos nutrientes, hay seis que son denominados macronutrientes, entre ellos tenemos al nitrógeno (N), magnesio (Mg), potasio (K), fósforo (P), azufre (S), calcio (Ca). Los ocho nutrientes restantes, son conocidos como micronutrientes, en este grupo se encuentran el boro (B), hierro (Fe), cobre (Cu), molibdeno (Mo), manganeso (Mn), cloro (Cl), zinc (Zn) y níquel (Ni). Las deficiencias nutricionales en las plantas, es la falta o carencia de uno o varios nutrientes esenciales, las cuales reducen no solo el desarrollo, sino también el rendimiento del cultivo. Para diferenciar los síntomas de otros problemas se debe tener en cuenta que las deficiencias de nutrientes siempre siguen un patrón simétrico en la hoja, en cuanto a la coloración, forma y tamaño [22].

1.5.3. Análisis foliar

A través del análisis foliar, se puede verificar la presencia de signos visibles de deficiencias nutricionales, detectar la falta de nutrientes, ayudar a evaluar la capacidad relativa del suelo y medir el impacto del suministro de nutrientes en su concentración en la planta. Además, permite investigar cómo el rendimiento del cultivo y el estado nutricional de la planta están relacionados [23].

1.5.4. Inteligencia Artificial (IA)

Para López, R. y Meseguer, P. [24], mencionan que la inteligencia artificial es el campo de la ciencia e ingeniería que permite diseñar y programar máquinas para realizar tareas que requieren inteligencia humana. El enfoque se centra en desarrollar sistemas y tecnologías que asistan con las tareas diarias de las personas, tales como la percepción, el aprendizaje y la toma de decisiones ante eventos particulares. Se usan modelos matemáticos y algoritmos en este tipo de sistemas para analizar datos, identificar patrones y solucionar problemas. En los últimos años, su aplicación se ha expandido a una variedad extensa de usos, como asistentes virtuales, sistemas de reconocimiento, vehículos autónomos y

diagnósticos médicos avanzados. La forma en que interactuamos con nuestro entorno ha experimentado cambios y ha generado varias oportunidades en distintos aspectos de la vida humana.

Así mismo, para Palma y Marín [25], la IA se divide como ciencia y como ingeniería. Como ciencia es un trabajo de análisis que está compuesto por sucesos de neurología y cognición; en otras palabras, subrutinas de determinación, el lenguaje, la percepción, el efecto, el recuerdo y el hecho. Como ingeniería presenta el conocimiento con una finalidad accesible, clara y un tiempo determinado, que tiene como base la observación, el modelado, la formalización y la transformación, mediante procesos que se le denomina aprendizaje.

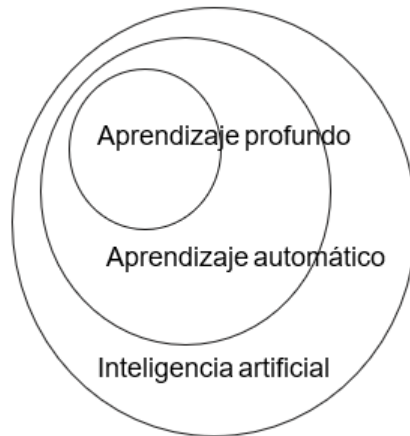


Fig. 3: Inteligencia Artificial y su jerarquía de conceptos [26].

1.5.4.1. Aprendizaje automático

Según [26] es una rama de la ciencia de la inteligencia artificial que utiliza métodos y algoritmos para ayudar a las computadoras a aprender de manera automatizada a partir de experiencias previas. Las máquinas pueden aprender patrones utilizando una Variedad de métodos y modelos, pero no pueden Aprender conceptos complejos y abstractos a partir de Los mismos datos. De acuerdo con los expertos de Microsoft [27], el aprendizaje automático tiene en los siguientes puntos:

- Recopilar y preparar los datos
- Entrenamiento del modelo.
- Implementación del modelo.
- Predicciones de manera automatizada.

1.5.4.2. Tipos de aprendizaje automático

A. Aprendizaje supervisado

El aprendizaje supervisado es una técnica algorítmica predictiva que se fundamenta en características o comportamientos previos para aprender de datos históricos etiquetados. Se inicia este enfoque con un conjunto de observaciones que incluyen las entradas y salidas esperadas, para así aprender la relación entre ambas. Principalmente se utiliza el aprendizaje supervisado en dos tipos de problemas: el análisis de regresión y la clasificación. La clasificación busca asignar una etiqueta o categoría a cada instancia de datos, mientras que la regresión intenta predecir un valor numérico continuo [25].

B. Aprendizaje no supervisado

La capacidad de este tipo de aprendizaje es realizar tareas mucho más complejas y predecir patrones sin la necesidad de datos previamente estructurados. El aprendizaje no supervisado busca características o patrones significativos en los datos de entrada, ya que carece tanto de salida como de un método para comparar, por lo tanto, el modelo solo utiliza la entrada [25].

1.5.4.3. Matriz de confusión

La matriz de confusión binaria, conocida en inglés como Confusion matrix, sirve para representar visualmente los aciertos y errores del modelo de clasificación y también para mostrar cómo funciona un algoritmo de aprendizaje supervisado [26].

Tabla 3: Matriz de confusión

		Clase predicha	
		P	N
Clase verdadera	P	TP	FN
	N	FP	TN

Nota: Tomado de Bosh, Casas, y Lozano [26].

En la tabla anterior se muestran dos clases: La clase P incluye los números positivos y la clase N los negativos. Las calificaciones correspondientes se encuentran dentro de estas clases: Los verdaderos positivos se representan como TP, los falsos positivos se indican con FP, los falsos negativos son FN y los verdaderos negativos están representados por TN. Estos son considerados componentes de una matriz de confusión.

1.5.4.3.1. Métricas de Rendimiento

Las métricas de rendimiento datos y números que evalúan la eficacia y calidad de los modelos entrenados son conocidos como métricas de rendimiento. Con estas métricas podemos evaluar el rendimiento de un modelo en tareas concretas, como la clasificación o regresión. Al evaluar el rendimiento de un modelo, podemos tomar decisiones informadas sobre su idoneidad para su implementación en la producción. Existen diversas métricas que nos ayudan a comprender el desempeño de un modelo, las más comunes son la exactitud, la precisión, recall y puntuación F1 Score, entre otros [28].

A. Exactitud (accuracy)

Según Giacomantone [29], es una métrica que mide el grado de éxito del modelo. Para realizar su cálculo se debe dividir la suma de los verdaderos positivos y negativos por el tamaño total de la muestra. Básicamente, la exactitud mide cuántas predicciones precisas hace el modelo en comparación con el total de predicciones. Su fórmula se denota de la siguiente manera.

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

B. Precisión

Bejar [30], menciona que es la medida que muestra la proporción de predicciones positivas correctas que realiza un modelo en comparación con todas las predicciones positivas. En otras palabras, evalúa la frecuencia con la que el modelo acierta al predecir muestras positivas correctamente. Su fórmula es la siguiente:

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

C. Exhaustividad (recall)

Para [26], La métrica de recall evalúa la capacidad de un clasificador para identificar todas las muestras positivas correctamente en el conjunto de datos. Básicamente, mide la capacidad del modelo para detectar los verdaderos positivos, sin importar si fueron identificados o no. Su fórmula se representa de la siguiente manera.

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

D. Puntuación f1 (F1 score)

La puntuación F1 es una métrica eficaz para abordar los desafíos de clasificación que involucran conjuntos de datos desequilibrados, ya que combina precisión y recall. La puntuación que se obtiene cuantifica el equilibrio entre estas dos métricas, lo que la

convierte en una herramienta útil para resolver problemas de este tipo [26]. El cálculo de esta métrica implica el uso de la siguiente fórmula

$$F1 = 2x \frac{Pre + Rec}{Pre + Rec}$$

1.5.4.4. Aprendizaje Profundo

También conocido como redes neuronales artificiales, según Soria, Rodríguez y García [31], es un subconjunto del aprendizaje automático, a diferencia, esta área abarca modelos capaces de aprender por sí solos. El modelo crea de manera automática una jerarquía conceptual relacionando unos con otros hasta lograr resultados más complejos y abstractos al relacionar múltiples capas, donde se encuentra la profundidad de los datos. El aprendizaje profundo se aplica en muchos campos, incluyendo el reconocimiento de voz, el procesamiento del lenguaje natural, la visión por computadora, la generación de imágenes y el aprendizaje por refuerzo. Algunas de sus características son:

- Según el contexto de problema que se pretende abordar y los datos disponibles, se puede aprender de manera supervisada, semi supervisada o no supervisada.
- Se necesita una mayor cantidad de datos y recursos computacionales para entrenar de forma eficaz.
- Puede generalizar y adaptarse a nuevas situaciones y desafíos.

i. Redes neuronales artificiales

El propósito de estos modelos informáticos es replicar las acciones del cerebro humano. En esencia, el objetivo es que las máquinas adquieran conocimientos de forma similar al cerebro humano utilizando neuronas artificiales dispuestas en varias capas para enviar y recibir información [26].

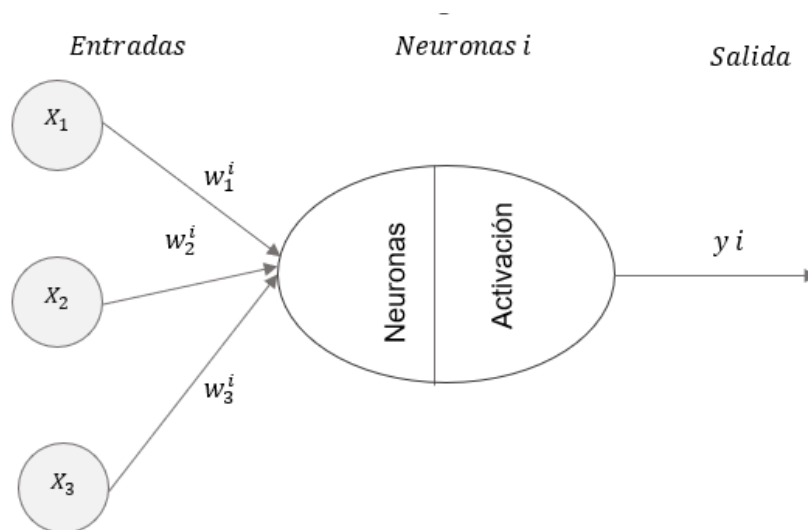


Fig. 4: Esquema básico de una neurona artificial [26].

Como se puede observar en la figura anterior, cada neurona tiene un número determinado de entradas, que se denotan con $x = \{x_1, \dots, x_2, x_n\}$. Cada una de las entradas tienen pesos que denotan como $w^i = \{w_1^i, w_2^i, \dots, w_n^i\}$. Cada neurona combina estos diversos valores de entrada, para ello utiliza la llamada función de entrada o combinación y su valor resultante se procesa por una función de activación, el cual modula los valores de entrada para poder generar un valor de salida, que en la figura se denota como y_i .

a. Funciones de entrada o combinación

De acuerdo con [32], las funciones de entrada son las encargadas de acoplar conexiones entrantes con su nivel de importancia o peso y agregan valores que se obtienen de las conexiones entrantes para adquirir un valor único de salida. Estas funciones actúan como un filtro que determina si la neurona debe dispararse o no y en qué medida. Elegir la función de activación correcta puede tener un impacto significativo en el rendimiento de una red. Los tipos de función de entrada o combinación son:

1. **Lineales:** es una función simple y se define como $f(x)=x$. permite generar combinaciones lineales a partir de las entradas, pero no es capaz de modelar relaciones complejas.
2. **Sigmoidales:** función que actúa como separador más o menos suave de las salidas, esto a partir de un parámetro que usa para determinar la forma de la curva $f(x) = \frac{1}{1+e^{-\frac{x}{p}}}$.
3. **Relu:** Es una función no lineal que toma una entrada y la transforma mediante una suma ponderada para generar la salida, se utiliza para la generación de predicciones. Está compuesta por tres tipos de capas. Las de entrada utilizan los datos de entrada en bruto para realizar el cálculo. Las ocultas, que recogen las entradas de otras capas y luego generan las salidas para otras capas. Y finalmente las de salida, las que predicen el resultado final.
4. **Softmax:** es una función que toma un conjunto de valores, como las salidas de una neurona en una red neuronal, y los convierte en un conjunto de probabilidades, lo que significa que esta función se puede usar para determinar que una determinada neurona pertenezca a una determinada clase. Se define de la siguiente manera:

$$z_j^L = \sum_k w_k^L y_k^{L-1} + \varphi_k^L$$

Donde:

w_k^L es el peso que se le asigna a una neurona k de la capa $L - 1$ y neurona j de la capa L .

y_k^{L-1} es el valor que produce la neurona k de la capa $L - 1$.

φ_k^L indica el valor de sesgo de la neurona j de la capa L .

ii. Arquitecturas de redes neuronales

También conocidas como topologías de redes neuronales, se definen como la manera en que están organizadas en distintas capas, parámetros de configuración tales como las funciones de activación. Cada una de las arquitecturas depende de algunos tipos de problemas que se quiere enfrentar y presenta diversos niveles de eficacia de los resultados y niveles de coste computacional. Dentro de estas arquitecturas tenemos a las redes neuronales simples, que son redes compuesta por una sola neurona que está conectadas al conjunto de entradas disponibles y tienen una sola capa de salida, estas redes permiten efectuar funciones relativamente sencillas, tales como técnicas estadísticas de regresión no lineal. Las redes neuronales de una sola capa se componen de una capa de entrada, una capa oculta para el procesamiento y una capa de salida con una o más neuronas. A diferencia de estas redes, las redes neuronales de una sola capa no pueden realizar clasificaciones o predicciones complejas o de dimensiones superiores. La característica que distingue a las redes neuronales recurrentes (RNN) de sus predecesoras es que su salida se utiliza como entrada para otra capa. En las RNN, sus neuronas se activan durante un tiempo limitado, lo que permite que otras neuronas se activen un poco más tarde y también durante un tiempo limitado. En otras palabras, las RNN reutilizan todas las salidas de una capa (i) como entradas en otra capa [26].

iii. Optimización de los procesos de aprendizaje.

Existen diversas técnicas que permiten la optimización, dentro de esto se puede diferenciar tres grupos principales, tales como los problemas de rendimiento, denominado también performance, que se utiliza para mejorar la capacidad de predicción de una red. Velocidad de aprendizaje que se utiliza para la reducción del tiempo cuando se entra una red neuronal y Problemas de o overfitting o sobre

entrenamiento que se da por el sobre ajuste de la red, es decir la red se vuelve demasiado flexible con los datos de entrenamiento, pero que, al evaluarlos con otros tipos de datos, no es capaz hacer generalizaciones de la manera correcta. Frente a esto existen diversas técnicas para automatizar este proceso, dentro de los que más resaltan son la búsqueda en cuadrícula (grid search), que consiste en la distribución uniforme de los valores entre los distintos parámetros aplicando las más posibles combinaciones y la otra técnica denominada búsqueda aleatoria (random search), que consiste en elegir combinaciones aleatorias dentro de sus rangos prefijados [33].

iv. **Técnicas de rendimiento de las redes neuronales**

Palma y Marín [25], mencionan que con el fin de mejorar el rendimiento de una red se puede establecer utilizar diversas técnicas que ayuden a un notable rendimiento, de las cuales, las que más se utilizan son:

- a) **Las épocas:** Se emplean para señalar cuántas veces el conjunto completo de datos de entrenamiento debe pasar por la misma red, ya que hacerlo una vez no es adecuado.
- b) **Tamaño de lote:** conocido como batch size, se refiere al número de interacciones necesarias para completar una época, poniendo por ejemplo un número de datos de 200 y ponemos lotes de 50, se necesitaría cuatro interacciones para completar una época.

v. **Algoritmos de entrenamiento de las redes**

Según Bosch, Casas y Lozano [26], mencionan que aumento de rendimiento y velocidad para la etapa de entrenamiento de una red depende del tipo de optimizadores que se haya seleccionado. Los más populares son:

1. **Adargad:** algoritmo basado en gradientes con velocidad de aprendizaje adaptativa de acuerdo con los parámetros realizando bajas tasas de aprendizaje para características frecuentes y altas tasas de aprendizaje para características infrecuentes.
2. **Ada delta:** En lugar de acumular gradientes anteriores, se limita a una ventana de tamaño fijo para reducir la velocidad de aprendizaje.
3. **Adam:** del inglés Adaptive moment estimation, en español, estimación del momento adaptativo, este optimizador es una combinación de otros optimizadores y principalmente se basa en el uso promedio exponencial de decreciente de gradientes pasadas.

vi. Técnicas de sobre entrenamiento

1. **Regularización:** esta técnica consiste en agregar un término extra a la función de coste, es decir es la modifica los pesos en la red para que esta se haga de manera gradual, esta se establece como una forma de equilibrio entre encontrar los pesos pequeños y minimizar los costes [33].
2. **Dropout:** Es una técnica de regularización para evitar el sobreajuste, donde determinadas neuronas se desactivan aleatoriamente durante el proceso de entrenamiento. Al incorporar diversidad y redundancia en las conexiones neuronales, el modelo se vuelve más resistente y capaz de generalizarse [33].

vii. Redes neuronales convolucionales (CNN)

Para Bobadilla [34], el uso típico del aprendizaje profundo involucra el empleo de

redes neuronales convolucionales especializadas y diseñadas para procesar datos estructurados en forma de cuadrículas, como imágenes. Estas redes se destacan por su capacidad para resolver múltiples problemas, como la clasificación de objetos según sus características únicas. Cuando se entrenan con ejemplos etiquetados, estas redes pueden desarrollar la capacidad de reconocer rostros, escenas, objetos y otros elementos visuales. Esta tecnología se ha aplicado de manera notable en la clasificación de imágenes, el reconocimiento facial, la detección de objetos y la segmentación semántica.

1. **Extracción de características:** En el aprendizaje profundo implica transformar los datos de entrada en un conjunto más pequeño y significativo de características.
2. **Clasificación:** Esta tarea consiste en asignar una etiqueta a una imagen en función de su contenido, permitiendo así identificar la clase a la que pertenece la imagen.

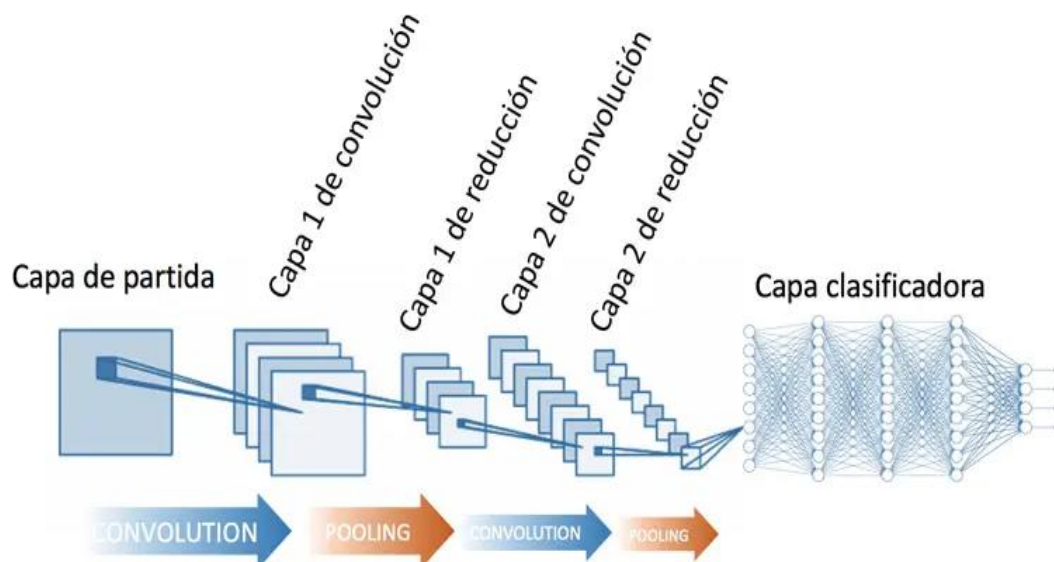


Fig. 5: Una red neuronal convolucional CNN. Tomado de Diego Calvo [35].

viii. Estructura y componentes de una CNN

a) Capa convolucional

Es un conjunto de operaciones matemáticas que se encarga de recibir de manera sucesiva la información contenido en los diversos pixeles y determina que va a pasar a las señales obtenidas después de pasar por un determinado proceso [26]. Cuando hablamos de una capa convolucional se debe considerar los conceptos de kernel o filtros, padding y Maxpooling.

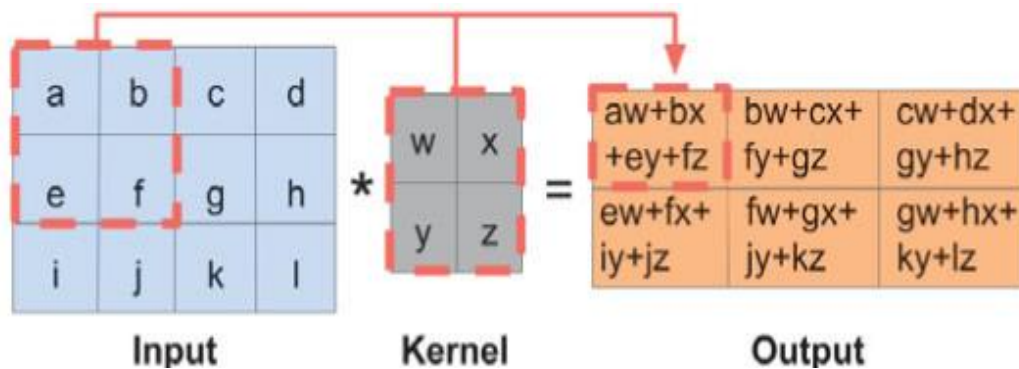


Fig. 6: La figura muestra la capa convolucional 2D. Tomado de Bosch, Casas y Lozano [26].

b) Kernel o filtro

Utiliza la convolución para extraer las características relevantes deslizándose sobre una imagen de entrada con una matriz pequeña. Cada núcleo puede reconocer patrones específicos, como bordes o texturas. A lo largo del proceso de entrenamiento, la red neuronal obtiene valores óptimos para sus núcleos, mejorando así su capacidad para extraer las características más significativas de las imágenes

de entrada [26].

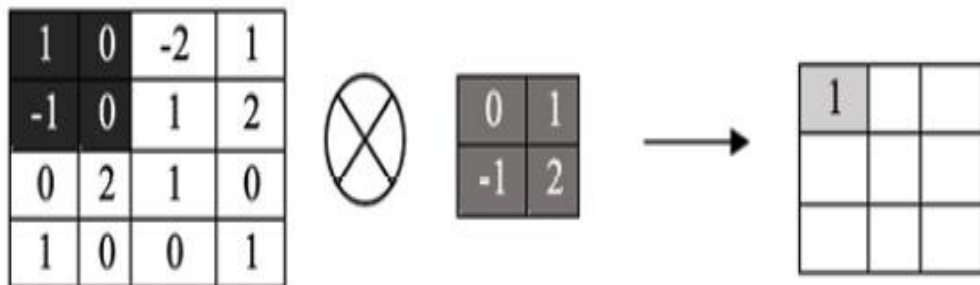


Fig. 7: La figura muestra la operación de convolución de una imagen de 4x4 y un kernel de 2x2

c) Capa Maxpooling

Es una función que selecciona el máximo valor de los conjuntos de entrada, de tal modo que sus salidas corresponden al máximo de cada región. Esto reduce la cantidad de parámetros y el costo computacional, manteniendo las características más importantes de la imagen.



Fig. 8: Representación de capa Max Pooling de una matriz de 4x4 para obtener una resultante de 2x2. Tomado de [31].

d) Padding

Es una función con la cual se añade ceros a los bordes es de entrada, de tal manera que las dimensiones de salida sean iguales a las dimensiones de entrada aun después que se aplican convoluciones.

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	12	95	4	1	3	0	0
0	0	15	27	8	2	1	0	0
0	0	25	39	41	11	2	0	0
0	0	8	10	18	57	71	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Fig. 9: Ejemplo de padding de 2x2. Tomado de Soria, Rodríguez y García [31].

ix. Arquitecturas CNN

VGG16

La arquitectura VGG16 es una Convolutional Neural Network (CNN) que se conoce por su profundidad porque tiene 16 capas convolucionales y está completamente conectada. Fue creado por el Visual Geometry Group (VGG) de la Universidad de Oxford. Utiliza filtros pequeños (3x3), capas de agrupación máxima y activación Relu en todas las capas convolucionales para que se destaque. Por su sencillez y alto rendimiento se utiliza como referencia para tareas de clasificación de imágenes [31].

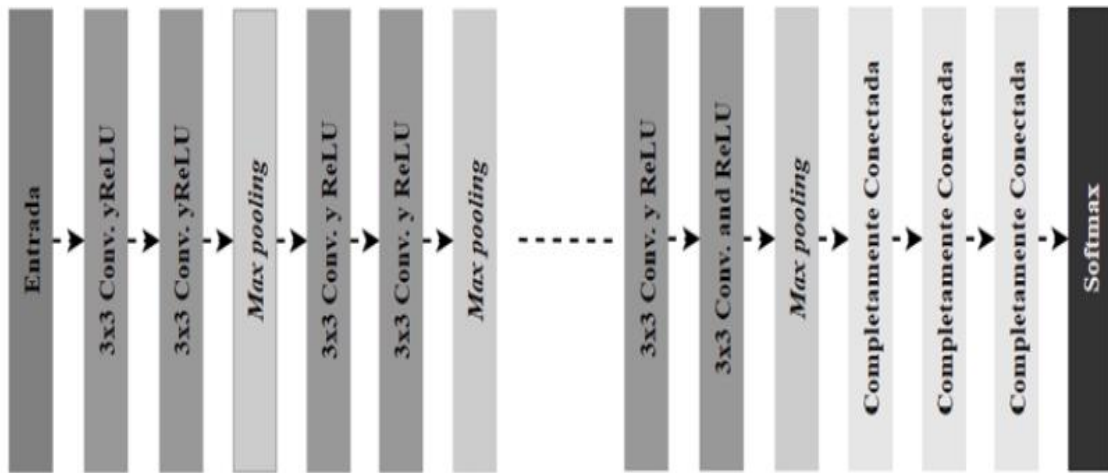


Fig. 10: Arquitectura GG16. Tomado de Soria, Rodríguez y García [31].

VGG19

Es conocido por su profundidad, al igual que VGG16, pero tiene 19 capas en total. Utiliza activaciones Relu en todas las capas convolucionales y filtros pequeños (3x3). Esto lo hace muy efectivo en tareas de visión por computadora como clasificar imágenes.

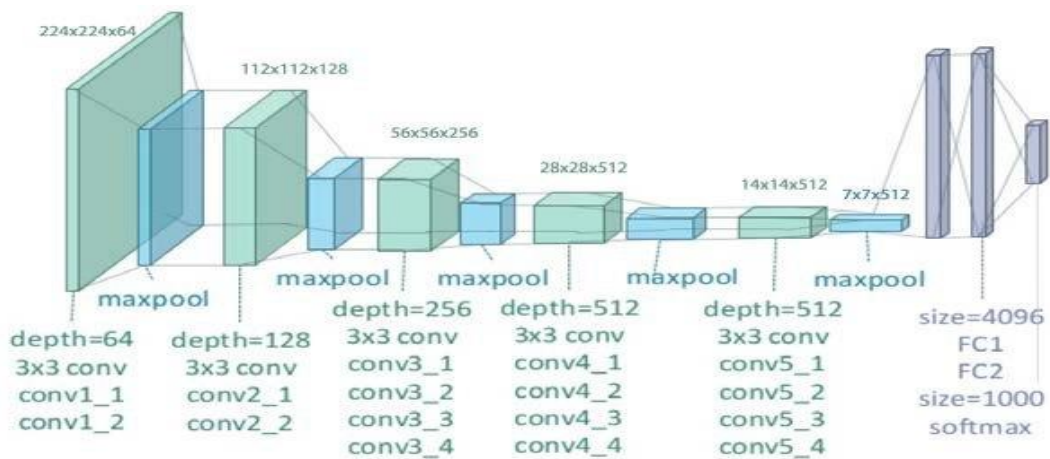


Fig. 11: Ilustración de la arquitectura de red del modelo VGG-19: totalmente conectado.

Tomado de [36].

Inceptionv3

La arquitectura Inception v3 de Google es una Convolutional Neural Network (CNN) que consta de 42 capas, utilizado para la clasificación de imágenes. Su uso de módulos de convolución, también conocidos como módulos de convolución, destaca por su eficiencia en el uso de recursos computacionales. Mediante estos módulos, se pueden capturar características de diversas escalas en una única capa, lo que resulta en la reducción de parámetros sin sacrificar el rendimiento en tareas de clasificación de imágenes.

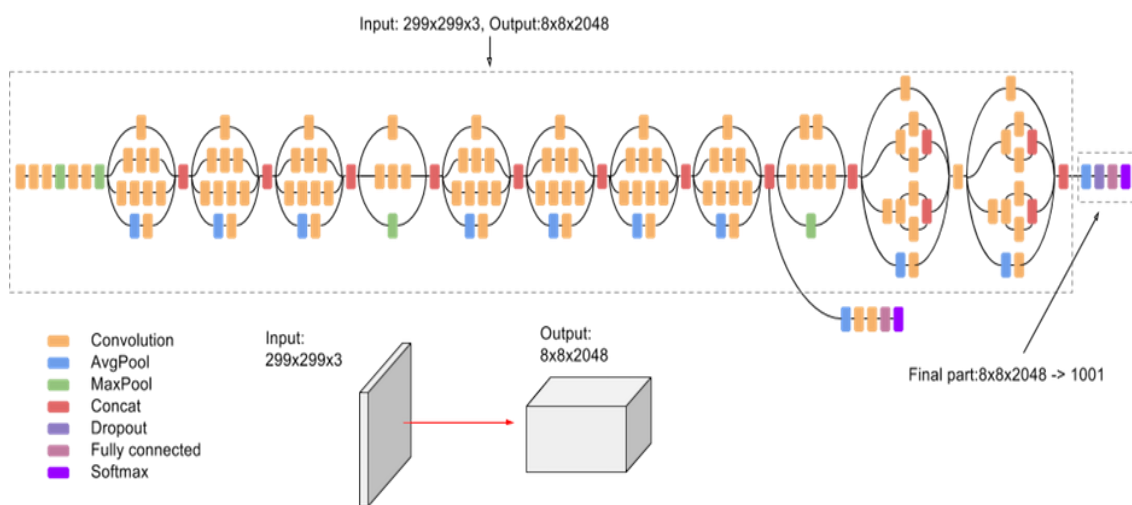


Fig. 12: Representación gráfica del modelo Inception V3. Tomado de Benjamín Qochuk [37].

MobileNet

La arquitectura MobileNet fue desarrollada por Andrew G. Howard y otros investigadores de Google con el propósito de integrar redes neuronales convolucionales en dispositivos móviles que cuentan con recursos informáticos limitados. Esta tecnología, conocida como "descomposición convolucional profunda", se destaca por su eficacia al reducir tanto la cantidad de parámetros como las operaciones computacionales, manteniendo al mismo tiempo un alto nivel de competitividad en las

tareas de clasificación de imágenes. Este modelo consta de 27 capas de convolución, que incluyen una capa Softmax, una capa totalmente conectada, una capa de grupo promedio y trece capas de convolución en profundidad. Las versiones más pequeñas de MobileNet poseen 1,32 millones de parámetros, mientras que el modelo estándar cuenta con 4,2 millones. Estas cifras son inferiores a las de otros modelos de aprendizaje automático comunes, lo que lo convierte en una opción altamente efectiva incluso cuando se dispone de un conjunto de datos muy pequeño [37].

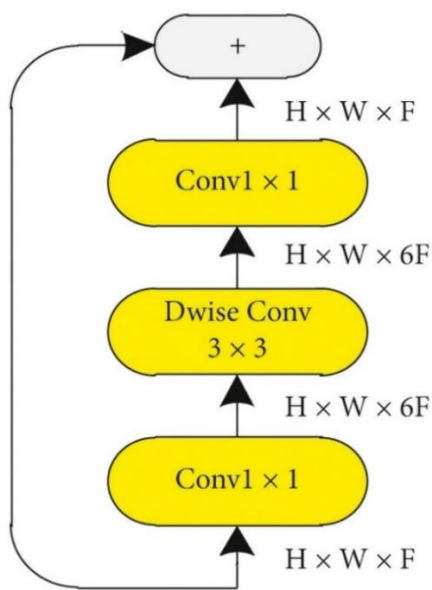


Fig. 13: Estructura de la arquitectura Mobile net. Tomado [38].

1.5.4.5. Procesamiento de imágenes

Una imagen bidimensional se puede definir como la representación en un plano o escena de un objeto situado dentro de un espacio tridimensional. El procesamiento de imágenes digitales nace con la necesidad que después que una imagen ha sido adquirida, puede ser deseable modificarla con el fin de mejorar la representación a la que hace realidad o con el motivo de resaltar algunas características particulares [39]. Así mismo, Gonzales,

Woods y Másteres [40], mencionan que son técnicas y procedimientos que son aplicadas a imágenes digitales con el fin de mejorar la información que contienen u obtener información imperceptible a simple vista. Este proceso también es conocido como visión por computadora. El siguiente diagrama muestra el procedimiento para procesar imágenes digitales.

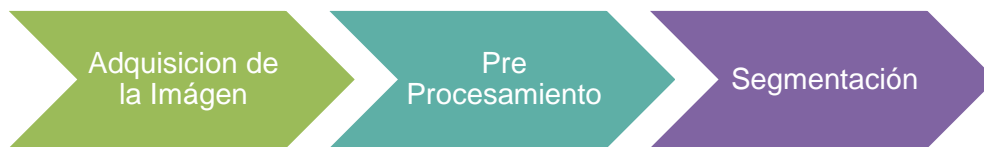


Fig. 14: Representación gráfica general de procesamiento de imágenes. Tomado de Bosch, Casas y Lozano [26].

1. Adquisición de la imagen.

La adquisición de imágenes es una de las etapas primordiales del procesamiento de imágenes, es la etapa donde se captura la información visual del mundo real. Esta información se convierte en una señal digital que puede ser procesada y analizada por una computadora, las imágenes esta puede ser capturadas por medio de cámaras digitales o mediante la digitalización de escáner, representa un medio de entrada para el modelo [41].

2. Pre-Procesamiento

Después que se tiene una imagen digitalizada es posible realizar mejoras y ajustes en la apariencia visual y calidad de unas varias imágenes con finalidad de corregir posibles errores que estas puedan presentar de tal modo que nos facilite un mejor análisis, dentro de una aplicación en específico. Es decir, es la etapa inicial del procesamiento de imágenes donde se modifica la imagen original para mejorar su calidad o facilitar su análisis posterior

[42]. Algunas técnicas más comunes incluyen.

a. Eliminación de ruido

El ruido es una distorsión no deseada que puede ocurrir en las imágenes debido a varios factores, se manifiesta generalmente por las variaciones en el nivel de gris que presentan los píxeles, respecto a sus vecinos. Es producido por señales aleatorias impredecibles que se originan de manera natural dentro o fuera del sistema de adquisición de la imagen [39]. Algunas técnicas para la eliminación de ruido son:

- **Filtro Gaussiano**

También conocido como filtro de suavizado, es un filtro lineal que se utiliza para eliminar el ruido de una imagen y suavizar las transiciones entre píxeles adyacentes. Asigna peso a los píxeles cercanos según su función gaussiana, que disminuye a medida que aumenta la distancia. Los dos parámetros que afectan el efecto del filtro gaussiano son el tamaño de la función gaussiana y su desviación estándar; el primero producirá un efecto de suavizado más fuerte, mientras que el segundo producirá un filtro más suave. Es muy utilizado en el procesamiento de imágenes debido a que, al ser un filtro lineal, se puede aplicar de forma rápida y eficiente a grandes imágenes. Al ser un filtro separable, se puede dividir en dos filtros de una dimensión, lo que reduce el tiempo de procesamiento. Así mismo, tiene la propiedad de preservar las formas de los objetos en la imagen, mientras que elimina el ruido no deseado [42].

- **Filtro Ponderado**

El filtro de promedio ponderado es un método de suavizado de imágenes que utiliza una combinación lineal de los valores de sus vecinos para asignar el valor de cada píxel, asignando diferentes pesos a cada vecino, basados en la distancia espacial entre el píxel central y sus vecinos. Este filtro es más efectivo y flexible que

el filtro de promedio simple, ya que conserva los bordes y detalles importantes de la imagen, y permite ajustar los pesos a las necesidades específicas de la aplicación [43].

b. Mejora de contraste

Las condiciones de iluminación desiguales en una imagen digital pueden crear sombras y distorsiones, lo que plantea desafíos para el proceso de segmentación. El contraste, se refiere a la variación en la intensidad de los píxeles, juega un papel crucial en la percepción y el análisis de las imágenes. Las imágenes de bajo contraste presentan dificultades para visualizar e interpretar los datos subyacentes. Para abordar estos problemas, existen varias técnicas para mejorar el contraste de la imagen, que incluyen:

- Ecuación de histograma

El histograma muestra la distribución de los valores de intensidad de los píxeles en forma gráfica. La ecualización del histograma estira este rango de valores para que abarque todo el espectro de grises disponibles, de 0 a 255. El objetivo es lograr un histograma más uniforme, donde cada nivel de intensidad tenga una representación similar [44].

- Adaptación del contraste local

La adaptación del contraste local es una técnica que mejora el contraste de una imagen local, adaptándose a las características específicas de cada región de la imagen, en contraste más precisa y preservando detalles importantes. Se basa en la noción de que el contraste es más crucial en áreas con alto detalle, como los bordes de

los objetos, por lo que divide la imagen en pequeñas áreas y calcula el contraste local de cada una de ellas, lo que mejora la percepción visual y resalta los detalles relevantes [43].

3. Segmentación

El proceso principal del procesamiento de imágenes es la segmentación de imágenes, que divide una imagen en diferentes regiones o segmentos. El objetivo principal es identificar y aislar los objetos o características de interés para facilitar el análisis e interpretación posteriores. Es una etapa crucial para una variedad de aplicaciones, como el reconocimiento de objetos, la detección de bordes y la clasificación de imágenes [42]. Existen diferentes técnicas de segmentación, tales como:

a. Segmentación por umbralización

La segmentación por umbralización es un método de procesamiento de imágenes que implica dividir una imagen en dos regiones: el objeto y el fondo. Está basada en un umbral de intensidad, que es un valor que separa los píxeles que pertenecen al objeto de los píxeles que pertenecen al fondo. Su funcionamiento se basa en primer lugar en seleccionar el umbral adecuado, que puede ser elegido manualmente o calculado automáticamente mediante métodos como Otsu, entropía máxima o k-means. En segundo lugar, la clasificación de píxeles, donde todos los píxeles de la imagen con un valor de intensidad mayor o igual que umbral se clasificarán como objeto y aquellos con intensidad menor que el umbral se clasificarán como fondo. Por último, como resultado de la segmentación por esta técnica, es una imagen binaria, en la que los píxeles del objeto se representan con un valor blanco (1) y los píxeles del fondo se representan con un valor negro (0)

[45].

Método Otsu

De acuerdo con Nobuyuki Otsu [47], es un algoritmo que usa el procesamiento de imágenes para segmentar automáticamente una imagen en dos o más clases, generalmente objetos y fondo. El principio fundamental de este enfoque implica determinar un umbral óptimo para dividir los píxeles de una imagen en dos grupos distintos. Este umbral se determina minimizando la varianza intraclase y maximizando la varianza Inter clase en un histograma de nivel de grises. El método se puede resumir en los siguientes pasos: Se calcula el histograma de nivel de grises, y a continuación se normaliza dividiendo cada valor del píxel por el número total de píxeles. Después, se calcula la probabilidad acumulada y los promedios para cada nivel de gris y finalmente se obtiene la varianza intraclase para todos los umbrales posibles. El umbral óptimo seleccionado divide la imagen en dos categorías: píxeles menores o iguales al umbral y píxeles mayores que el umbral. Para calcular la varianza intraclase utiliza la siguiente fórmula:

$$\sigma_w^2(t) = q_1(t) \cdot \sigma_1^2(t) + q_2(t) \cdot \sigma_2^2(t)$$

Donde:

$q_1(t)$ y $q_2(t)$ son las probabilidades totales de los dos grupos de píxeles que se formaron después de aplicar el umbral.

$\sigma_1^2(t)$ y $\sigma_2^2(t)$ son las varianzas de los dos grupos de píxeles.

La fórmula para calculo la varianza Inter clase es la siguiente:

$$\sigma_b^2(t) = q_1(t) \cdot q_2(t) \cdot [\mu_1(t) - \mu_2(t)]^2$$

Donde:

$\mu_1(t)$ y $\mu_2(t)$ son las medias acumuladas de los dos grupos de píxeles.

Se puede maximizar la relación entre la varianza intraclase y la varianza Inter clase seleccionando el umbral ideal, que se expresa como:

$$\eta(t) = \frac{\sigma_b^2(t)}{\sigma_w^2(t)}$$

Finalmente, el umbral óptimo se selecciona como el valor de t que maximiza la relación $\eta(t)$.

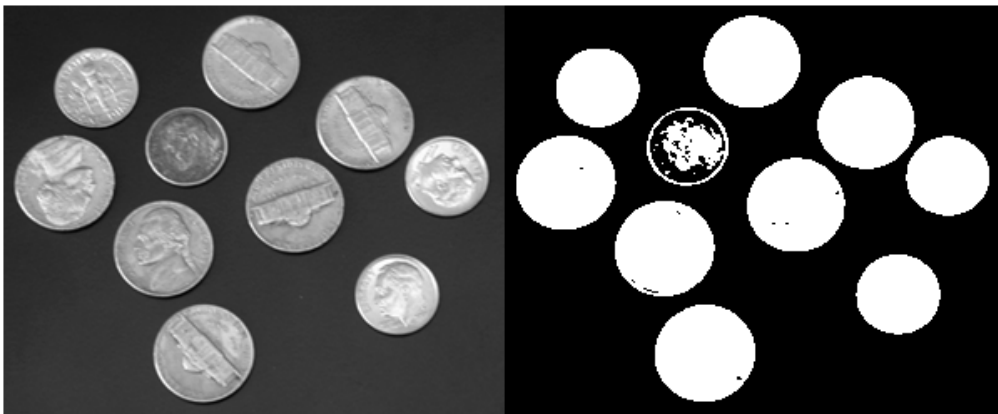


Fig. 15: representación de aplicar el método Otsu. Tomado de MathWorks [46].

b. Segmentación por regiones

Esta técnica se basa en la mezcla de píxeles cercanos con características similares, como intensidad, color y textura. Para aplicar esta técnica, primero se define las características que se van a utilizar para agrupar los píxeles. Seguido se hace la agrupación de los píxeles que comparten características similares, para ello se pueden utilizar algoritmos como k-means o watershed, entre otros. Por último, se funcionan las

regiones adyacentes que pertenecen al mismo objeto [43].

c. Segmentación por borde

Forero [39], menciona que la segmentación de imágenes es un proceso que divide una imagen en áreas o clases disyuntivas para identificar estructuras u objetos dentro de ellas. Segmentando los bordes o contornos de las áreas de interés, se pueden encontrar discontinuidades locales en la intensidad de los niveles de gris de la imagen. Para utilizar esta estrategia, el primer paso es identificar los bordes del objeto; se pueden utilizar varios métodos, como el Filtro de Sobel, el detector de Canny, el Filtro de Prewit, etc. En segundo lugar, se examinan los bordes para identificar las áreas de interés de la imagen. Finalmente, se rellenan las áreas encontradas con un valor específico para que se puedan identificar.

4. Regiones de Interés (ROI)

El concepto región de interés, se utiliza para identificar y seleccionar áreas específicas de una imagen que son pertinentes para un análisis o procesamiento posterior. Su funcionamiento implica varios pasos: El primer paso es encontrar el objeto relevante dentro de la imagen. Esto se puede lograr mediante el uso de técnicas de detección de características, reconocimiento de patrones u otras técnicas que sean apropiadas para el tipo de objeto que se busca. Después de identificado el objeto de interés, se puede delimitar su región en la imagen manualmente dibujando un contorno a su alrededor o automáticamente utilizando algoritmos de segmentación de imágenes. Una vez que se ha identificado la zona de interés, se pueden extraer las características pertinentes del objeto que se encuentra dentro de esa zona. Esto puede incluir características como el color, la

textura y la forma, dependiendo del tipo de análisis que se realice. Finalmente, se usa la región de interés y las características extraídas para realizar el análisis o procesamiento deseado de acuerdo de la aplicación específica, esto puede incluir tareas como clasificación, reconocimiento y seguimiento [47].

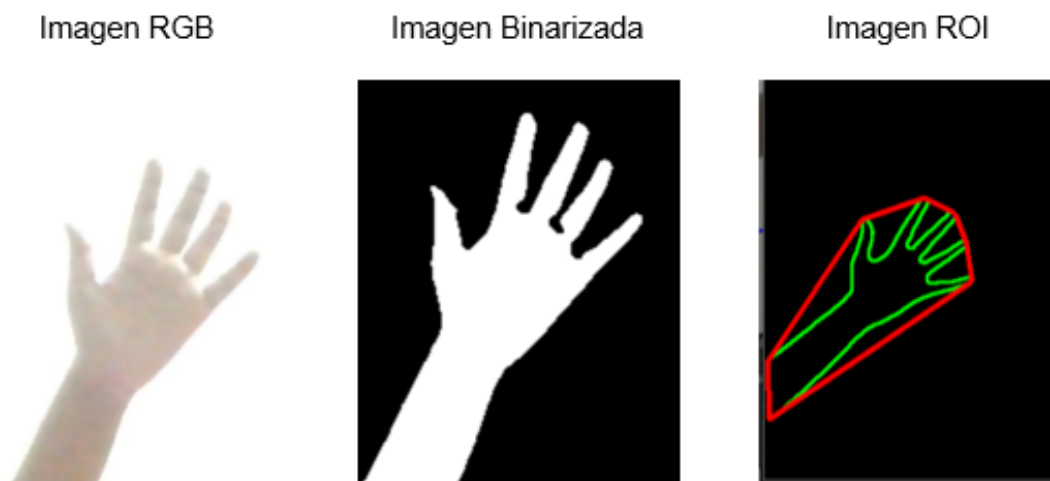


Fig. 16: Representación de imagen RGB, binarizada y aplicada la región de interés. Tomado de ICHI.PRO [47].

5. Espacios de color

De acuerdo con Forero [39], es una forma de representación abstracta, que permite crear, especificar o visualizar cualquier color. Dentro de sus principales características está que nos permite explicar el comportamiento del color en un contexto específico, facilitar la especificación del color en una forma normalizada y aceptada genéricamente, poseer una gama normalizada de colores y especificar un sistema de coordenada y un espacio, donde el color sea representado por un único punto, dentro de estos tenemos:

a. RGB

De sus iniciales en inglés: Red, Green y Blue, De sus iniciales en inglés: "rojo,

verde y azul", es un espacio de color que combina los colores primarios rojo, verde y azul para producir una amplia gama de colores sin usar demasiado otros componentes. Así que cualquier variación resulta de la suma aritméticas de los componentes, $X = R + G + B$. A pesar de ser uno de los modelos más populares, carece de características como tono, saturación e intensidad. De tal manera que no resulta adecuado para ciertos procesos como por ejemplo si quiere modificar su luminosidad de imagen.

b. RGB normalizado

Es una variante cromática que se expresa en el espacio de color RGB, como se muestra a continuación:

$$r = \frac{R}{R + G + B}$$

$$r = \frac{G}{R + G + B}$$

$$r = \frac{B}{R + G + B}$$

c. HSI

Espacio de color basado en el modo en que las personas perciben los colores, cada color se caracteriza en términos de tono, llamado H, saturación S y brillo o intensidad I. para pasar del espacio de color RGB a HSI, se utiliza las siguientes transformaciones matemáticas.

$$I = \frac{R + G + B}{3}$$

$$S = 1 - \frac{\min(R, G, B)}{I}$$

$$H = \begin{cases} \theta & \text{si } G \geq B \\ 2\pi - \theta & \text{si } G < B \end{cases}$$

Donde:

$$\theta = \arccos \frac{(R - G) + (R - B)}{2\sqrt{(R - G)^2 + (R - G)(G - B)}}$$

d. HSV

Definido por sus siglas en inglés, Hue Saturation value, es un modelo intuitivo de los artistas, emplea un sistema de coordenadas cilíndrico y subconjunto del espacio donde está definido en un cono hexagonal. Así como se muestra a continuación.

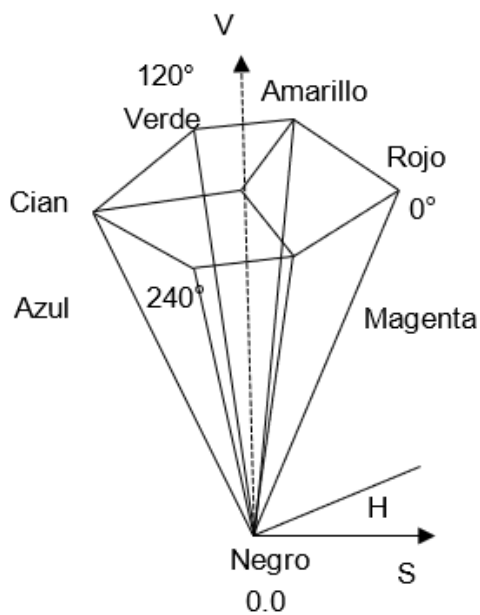


Fig. 17: Representación gráfica del de color HSV. Tomado de Forero Vargas [39].

6. Lenguajes de programación

Se trata de una herramienta para la redacción de software o programas que

definen y gestionan el comportamiento de dispositivos físicos y lógicos en computadoras. Los símbolos, las reglas gramaticales y la semántica son los elementos principales que le dan significado a sus componentes y expresiones, definiendo así su estructura. Las instrucciones escritas son convertidas en lenguaje de máquina (0 y 1) por el compilador, mientras que el intérprete lee las instrucciones para obtener el código de máquina adecuado. Un programa ejecutable se crea cuando un compilador convierte los símbolos de programación de un lenguaje de programación a lenguaje máquina [48].

- **Python**

En un lenguaje de programación más popular en la actualidad para automatizar funciones, analizar datos y crear sitios web. Además de su versatilidad y facilidad, se desarrolló bajo una licencia open source aprobada por OSI, y puede usar y compartir libremente, incluso en contextos comerciales [49].

7. bibliotecas o librerías

Ridge [50], menciona que una biblioteca es un conjunto de funciones y recursos predefinidos que nos permiten aumentar las capacidades de nuestro código sin tener que reescribir todo. Para hacernos una idea, suponiendo que estamos construyendo una casa y necesitamos poner un estante. Simplemente compras un estante ya construido y lo instalas en lugar de fabricar todos los componentes desde cero. De manera similar, las librerías nos brindan componentes y funcionalidades listas para usar en nuestro programa. Muchos lenguajes de programación, como Python, Java, JavaScript y C++, pueden usar librerías. Cada lenguaje tiene su propia biblioteca que se adapta a sus características y propósitos particulares.

a. Tensor Flow

Tensor Flow es una biblioteca de aprendizaje profundo que es propiedad de Google y está disponible de manera gratuita. Realiza una variedad de tareas de aprendizaje automático mediante programación de flujo de datos. Fue diseñado para funcionar en múltiples CPU o GPU e incluso en sistemas operativos móviles, y está codificado en una variedad de lenguajes, como Python, C++ o Java [51].

b. Keras

Keras es una biblioteca de redes neuronales de código abierto que puede usar Tensor Flow o Theano. Es modular, rápido y fácil de usar. François Chollet, ingeniero de Google, fue responsable de su creación. Es una biblioteca útil para desarrollar algoritmos de aprendizaje profundo [52].

c. OpenCV

Open Source Computer Vision, es una librería de programación de código abierto desarrollada en Nizhni Nóvgorod por la división rusa de Intel, se enfoca principalmente en la visualización de computadoras en tiempo real. Posee licencia open source BSD, lo cual permite su uso sin restricciones. Además, es compatible con múltiples plataformas. Está diseñado para funcionar con procesadores Intel, de tal manera que se utilizará automáticamente para acelerar el rendimiento de la aplicación si la librería detecta que las librerías de Intel IPP (Primitivas Integradas de Potencia) se encuentran en el sistema. Desde la versión 2.4.8, admite instrucciones vectoriales NEON para sistemas ARM, optimizaciones para Intel TBB (Threading Building Blocks) y optimizaciones SIMD.

d. Os

Biblioteca que proporciona acceso a las funciones del sistema operativo, lo que permite a los usuarios modificar archivos y directorios, obtener información sobre el entorno y ejecutar comandos del sistema. Ofrece una variedad de funciones, que incluyen manipulación de archivos y directorios, ejecución de comandos del sistema y operaciones de sincronización [53] .

e. Pathlib

Este módulo proporciona clases que representan rutas de acceso semánticas al sistema de archivos que se adaptan a varios sistemas operativos. Mientras que los caminos puros heredan de los caminos puros, los caminos concretos también ofrecen operaciones de E/S. Los trazados puros son útiles en situaciones especiales, como la manipulación de Las rutas de Windows en una máquina Unix [54].

f. Matplotlib

Matplotlib es una biblioteca Python versátil que permite la creación de visualizaciones estáticas, animadas e interactivas con estilos y diseños personalizables. También admite la exportación a varios formatos de archivo, la incrustación en JupyterLab y la integración con paquetes de terceros [55].

g. Time

Es una biblioteca estándar de Python que ofrece funciones relacionadas con el tiempo. Medición del tiempo de ejecución de un programa, trabajo con fechas y horas y otras operaciones relacionadas con el tiempo son posibles con esta biblioteca. [56].

h. Psutil

Psutil es una biblioteca versátil que funciona sin problemas en varias

plataformas y brinda acceso a datos cruciales relacionados con los procesos activos y el consumo de recursos del sistema, incluidos CPU, memoria, discos, redes y sensores. Su utilidad principal radica en la gestión de procesos en ejecución, monitoreo del sistema, creación de perfiles y la imposición de limitaciones a los recursos del proceso. Es compatible con los sistemas operativos Linux, Windows, macOS y Solaris [57].

8. Servicios en la nube

Son sistemas de infraestructura y software disponibles a través de Internet y están administrados por especialistas. En los últimos años, los servidores virtuales, los escritorios remotos, el almacenamiento de archivos, el procesamiento de datos y la gestión de recursos informáticos se han convertido en principales servicios para las operaciones comerciales [58].

a. Google Colaboratory

Es un servicio adicional de Google Research, Se trata de un editor que puede usarse desde un navegador para escribir código fuente en Python. Es un servicio que se basa en Jupyter Notebook y se aloja gratuitamente con una cuenta de Gmail. Además, proporciona herramientas informáticas que permiten la modificación y la prueba del código [59].

b. Google Drive

Se trata de un método para guardar datos en la nube. Los usuarios pueden almacenar y acceder a sus datos desde cualquier dispositivo conectado a Internet.

Entre sus características principales resaltan las siguientes:

- Almacenamiento en la nube
- Capacidad gratuita de hasta 15 GB.
- Compatibilidad: es multiplataforma, funciona en sistemas operativos como Android, Linux, Mac y Windows.
- Tipos de archivos: Admite archivos como fotos, videos, documentos de presentación, hojas de cálculo y otros.
- Privacidad y colaboración: Puede permitir que otros usuarios vean o modifiquen sus archivos privados en tiempo real. Esto resulta muy beneficioso para el trabajo en equipo.

II. MATERIALES Y MÉTODO

2.1. Tipo y Diseño de Investigación

La presente investigación es tipo aplicada debido a que aborda un problema real en el ámbito agrícola, proponiendo un método automatizado para la clasificación de deficiencias nutricionales en plantaciones de café mediante procesamiento de imágenes y aprendizaje profundo. Además, tiene un enfoque cuantitativo porque los resultados que se obtendrán serán numéricos medibles, que servirán como indicadores para evaluar el desempeño del modelo.

En cuanto al diseño de la investigación es cuasi experimental debido a que se trabajó con un grupo de deficiencias nutricionales conocidas y al método propuesto se realizó la evaluación del desempeño mediante análisis estadístico para comprobar nuestra hipótesis planteada. Aunque este enfoque no permite un control completo sobre todas las variables, nos brinda la oportunidad de obtener información valiosa sobre la capacidad de nuestro método en condiciones prácticas y reales.

2.2. Variables, Operacionalización

- **Variable Independiente**

Procesamiento de imágenes digitales y aprendizaje profundo

- **Variable Dependiente**

Clasificación automática de deficiencias nutricionales

Tabla 4: Operacionalización de la variable

Variable de estudio	Dimensión	Indicadores	Fórmula	Descripción	Técnicas e instrumentos de recolección de datos
Procesamiento de imágenes y aprendizaje profundo	Consumo de recursos	Consumo de CPU	$C_{cpu} = \sum_j^n \frac{cc_i}{n}$	<p>C_{cpu}: Consumo de la CPU</p> <p>\sum_j^n: Sumatoria de todas las cargas</p> <p>cc_i: representa la carga de la CPU en el instante de tiempo j.</p> <p>n: Es el número total de mediciones de la carga de la CPU realizadas durante el período de tiempo</p>	Instrumentos mecánicos o electrónicos - Registro electrónico
		Consumo de memoria RAM	$C_{ram} = \sum_j^n \frac{cm_i}{n}$	<p>C_{ram}: Consumo de memoria RAM</p> <p>\sum_j^n: Sumatoria de todas las cargas</p> <p>cm_i: representa la cantidad de memoria RAM utilizada en el instante de tiempo j.</p> <p>n: Es el número es el número total de mediciones realizadas durante el período de tiempo</p>	
		Tiempo de respuesta	$Tr = \sum_j^n \frac{tf_j - ti_j}{n}$	<p>Tr: Tiempo de respuesta</p> <p>\sum_j^n: Sumatoria del tiempo</p> <p>tf_j: Tiempo final</p> <p>ti_j: Tiempo inicial</p> <p>n: número total de instantes de tiempo evaluados</p>	

Clasificación automática de deficiencias nutricionales	Rendimiento del modelo	Exactitud	$E = \frac{TP + TN}{TP + TN + FP + FN}$	<p>E = Exactitud</p> <p>TP = Verdaderos positivos</p> <p>TN = Verdaderos negativos</p> <p>FP = Falsos positivos</p> <p>FN = Falsos negativos</p>	Instrumentos mecánicos o electrónicos - Registro electrónico
		Precisión	$P = \frac{TP}{TP + FP}$	<p>P = Precisión</p> <p>TP = Verdaderos positivos</p> <p>FP = Falsos positivos</p>	
		Recall	$R = \frac{TP}{FN + TP}$	<p>R = Recall o exhaustividad</p> <p>TP = Verdaderos positivos</p> <p>FN = Falsos negativos</p>	
		F1 score	$F1 = 2 \times \frac{P \times R}{P + R}$	<p>$F1$ = F1 score</p> <p>P = Precisión</p> <p>R = Recall</p>	

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

2.3.1. Población

Para llevar a cabo esta investigación, la población se compone cinco técnicas o métodos de procesamiento de imágenes y siete técnicas y modelos de aprendizaje automático y aprendizaje profundo. Estas selecciones se realizaron después de una exhaustiva revisión de la literatura en artículos y revistas científicas, utilizando bases de datos confiables como IEEE, Scopus, Science Direct, entre otras. Se identificaron diversas investigaciones relevantes para nuestro propósito de estudio, de las cuales se extrajeron las siguientes:

Tabla 5: Población considerada para la etapa del procesamiento de imágenes

N°	Técnica / Método	Exactitud	Ámbito	Ref.
1	Regiones de Interés (ROI)	98%	Visualización y clasificación de las enfermedades del café	[12]
2	Umbralización	91.49%	Clasificación y detección de deficiencias nutricionales en plantas de café	[15]
3	Filtro gaussiano	69 – 84.94%	Clasificación de enfermedades vegetales mediante aprendizaje automático	[13]
4	Operaciones morfológicas	100%	Identificación de sujetos basado en la huella palmar	[18]
5	Otsu	100%	Identificación de sujetos basado en la huella palmar	[18]

Nota: Elaboración propia

Tabla 6: Población considerada para la etapa de clasificación mediante aprendizaje automático y aprendizaje profundo.

N°	Técnica / Modelo	Exactitud	Ámbito	Ref.
1	ResNet50	77%	clasificación de las enfermedades del café.	[12]
2	VGG16	100%	identificación de sujetos basado en la huella palmar	[18]
3	DLANet	75.54%	Identificación de deficiencia de nitrógeno en plantaciones de arroz	[13]
4	MobileNet	93%	Aplicación móvil para la recomendación de fertilizantes en cultivos de arroz	[10]
5	InceptionV3	97%	Aplicación móvil para la recomendación de fertilizantes en cultivos de arroz	[10]
6	CNN	94.55%	Detección de enfermedades en cítricos	[14]
7	SVM	77%	Detección y clasificación de enfermedades del cáñamo	[17]

2.3.2. Muestra

El tipo de muestra seleccionada para la investigación es poblacional no estadístico y se ha elegido por conveniencia, tomando en cuenta al desempeño del modelo o técnica propuesta. Por lo que se eligieron los siguientes:

Tabla 7: Muestra seleccionada para la etapa del procesamiento de imágenes

N°	Técnica o método
1	Umbralización
2	Operaciones morfológicas (apertura y cierre)
3	Regiones de interés (ROI)
4	Método Otsu

Para la etapa de clasificación se consideran los siguientes

Tabla 8: Muestra seleccionada para la etapa de clasificación

N°	Técnica o modelo
1	Red Neuronal Convolutacional (CNN)
2	VGG16
3	Inceptionv3
4	MobileNet

2.3.3. Muestreo

Esta investigación emplea un tipo de muestreo no probabilístico ya que implicó una selección deliberada de métodos y técnicas de procesamiento de imágenes digitales y aprendizaje profundo. Este proceso de selección no fue aleatorio, sino que se llevó a cabo a través de un examen exhaustivo de la literatura científica existente, en el que se consideraron cuidadosamente los enfoques y técnicas que eran pertinentes para nuestro caso de estudio. Entre estos criterios de selección se incluyen el mayor rendimiento en cuanto a exactitud, mayor porcentaje de aciertos, las investigaciones realizadas y similitud a la presente

investigación.

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

2.4.1. Técnicas

a) Consultoría

Con el fin de obtener información sobre las deficiencias nutricionales del cafeto, se hizo una consultoría a un ingeniero agrónomo experto en cultivos de café. De lo cual él nos proporcionó un documento donde se detallan las diferentes deficiencias nutricionales que afectan al cultivo de cafeto, incluyendo sus síntomas, causas y posibles soluciones.

b) Observación

Con la información proporcionada por el experto agrónomo, se realizó la caracterización visual de las hojas de cafeto en un entorno natural. Este proceso implicó la visualización directa y de manera detallada de las hojas para identificar las deficiencias nutricionales. Luego se procedió a la elaboración de la base de datos, para ello se tuvo que construir un prototipo de adquisición de imágenes con el fin de que las fotografías se hicieran en un ambiente controlado y posteriormente se estableció un protocolo específico para obtener imágenes de mejor calidad.

c) Análisis documental

Se hizo un análisis exhaustivo de la literatura de diversos artículos y revistas científicas, libros, e informes técnicos, que contengan información relevante sobre deficiencias nutricionales en plantas. Después de eso se procedió a revisar y seleccionar aquellos que fueron más pertinentes para nuestra investigación, buscando que aborden

específicamente métodos y técnicas de diagnóstico mediante procesamiento de imágenes y aprendizaje profundo.

d) Experimental

Se realizaron varios experimentos para implementar el modelo propuesto. Las imágenes se procesaron utilizando una variedad de técnicas, mientras que el modelo en sí se entrenó utilizando métodos de aprendizaje profundo. Para evaluar su desempeño, se emplearon métricas específicas. Además, se utilizó un conjunto de datos independiente para realizar múltiples pruebas donde se comprobó la eficacia del modelo para clasificar con precisión las deficiencias nutricionales en las hojas de café.

2.4.2. Instrumentos de recolección de datos

En este estudio, se utilizaron instrumentos electrónicos durante la recolección de datos. Se integró un script en la implementación del modelo con el objetivo de generar la información requerida por el método propuesto. Se eligió el registro electrónico como herramienta principal que permitió verificar los resultados de las variables. Esto incluyó el consumo de CPU, la utilización de la memoria RAM y el tiempo promedio de respuesta para la variable independiente. Asimismo, se evalúan la exactitud, precisión, recuerdo y puntaje F1 como medidas de la variable dependiente, las cuales corresponden al desempeño del método propuesto en la clasificación de deficiencias nutricionales en hojas de café.

2.4.3. Validez y confiabilidad

La validez y confiabilidad de las técnicas utilizadas en la presente investigación están expresadas de la siguiente manera. En primer lugar, la consultoría que se hizo al experto agrónomo nos brindó las pautas necesarias de cómo identificar de manera visual las diversas

deficiencias nutricionales en hojas de cafeto, el cual se respaldó mediante documento, de esta manera garantizando que la caracterización de las deficiencias sea de forma precisa y confiable. En cuanto a la elaboración del conjunto de datos, con el fin de garantizar la consistencia y la calidad del conjunto de datos, se ha implementado un prototipo y se han establecido protocolos estandarizados de adquisición de imágenes. El prototipo se construyó de manera minuciosa para capturar imágenes controlando la luminosidad, brillo y posibles sombras, de modo que sean de buena calidad para su análisis posterior. Por lo tanto, los protocolos establecidos reducen la variabilidad de los datos asegurando que el proceso de adquisición de imágenes sea uniforme y reproducible. De esta manera, se logró obtener un conjunto de datos representativo y confiable para el entrenamiento, validación y pruebas del modelo propuesto. Para el análisis documental, se buscó a fondo artículos, revistas científicas, libros e informes técnicos sobre la clasificación de deficiencias nutricionales en las plantas mediante procesamiento de imágenes y aprendizaje profundo, de los cuales se seleccionaron los más relevantes para nuestra investigación, asegurando así la así la confiabilidad de las fuentes utilizadas. Finalmente, experimentó con varias técnicas y algoritmos de procesamiento de imágenes para encontrar la combinación más efectiva y precisa para aplicar el modelo propuesto. En el entrenamiento del modelo, se probó una variedad de configuraciones de parámetro para determinar la configuración ideal que maximice la eficacia del modelo. Además, en las pruebas de validación se usó un conjunto de datos diferente, con imágenes no etiquetadas para verificar que tan bueno es el modelo propuesto clasificando deficiencias nutricionales en hojas de cafeto con datos que no ha visto previamente, esto demuestra la validez y confiabilidad del modelo en entornos prácticos reales.

Pruebas Piloto

En la aplicación de las pruebas piloto, la información brindada por el experto agrónomo se constató con diversas fuentes, tales como páginas web reconocidas en brindar información

sobre el cultivo del café y bases de datos de otras investigaciones. Durante la adquisición de las imágenes se probaron diversos protocolos:

- Los focos leds se consideraron a una distancia de 5 o 10 cm del prototipo.
- La cámara del celular solo se consideró en modo foto normal sin ajustar ningún otro parámetro.
- La hoja se puso en el interior del prototipo a una distancia de 20 centímetros del celular.
- Para realizar la captura de la imagen se realizó manualmente interactuando con el móvil.

Esto, demostró que, los focos leds estaban muy cerca ya que las imágenes tenían mucha luminosidad, En cuanto a la cámara del celular, las imágenes que se capturaba tenían puntos de enfoque, Así mismo, las imágenes tenían puntos de sombra que hacían la misma hoja. De igual forma, la distancia entre la hoja y el celular era muy grande, ya que la hoja se veía muy pequeña.



Fig. 18: Imágenes tomadas considerando los protocolos iniciales

Para solucionar estos percances se hizo de diversas pruebas con diferentes parámetros, de los cuales, los que dan mejores resultados se consideraron los siguientes:

- Los focos leds se consideraron a una distancia de 25 cm del prototipo.
- La cámara del celular solo se consideró en modo profesional y en cuadrícula para que el enfoque se haga a toda la imagen.
- Para quitar la sombra que tenía imagen, se puso un soporte de 8x1cm de material transparente, de tal manera que parezca que la hoja está flotando en el aire. Esto disminuyó la distancia con el celular, quedando a una distancia considerable de 12 cm.
- Para realizar la captura de la imagen se usó la aplicación Anydesk para controlar al celular desde una laptop y hacer las capturas de las imágenes.



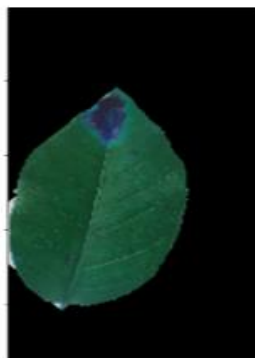
Fig. 19: imágenes tomadas considerando los parámetros finales establecidos

Para el procesamiento de las imágenes, con el fin de no perder la información que las imágenes contenían se redimensionaron a 900x1800 píxeles. Para la segmentación de las imágenes se utilizó el método Otsu, luego se aplicó operaciones morfológicas de apertura y cierre con un kernel elíptico de tamaño de 2x2, pero debido a que las imágenes aún presentaban algunos ruidos se probaron con un kernel de 3x3 y 5x5.

Morfología kernel 2x2



Morfología kernel 3x3



Morfología kernel 5x5

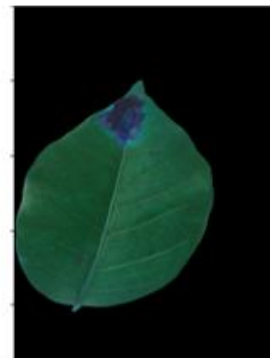


Fig. 20: imágenes aplicando morfología 2x2, 3x3 y 5x5

Haciendo las comparaciones se pudo determinar que usar el kernel de 5x5 se tenía mejores resultados. Finalmente, se aplicó la técnica de regiones de interés (ROI) para seleccionar como área específica a la hoja y hacer un recorte de acuerdo con su tamaño y posteriormente se realizó la redimensión de las imágenes a 128 x 240 píxeles.



Fig. 21: Imagen redimensionada 128 x 240 píxeles

Para entrenar la red neuronal convolucional los datos se dividieron el 80% para entrenamiento y 20% para validación. En la primera prueba, se configuraron tres capas convolucionales con 32, 64 y 128 filtros respectivamente, cada una con un tamaño de 2x2. Cada capa convolucional fue seguida por una capa MaxPooling de 2x2 para reducir la dimensionalidad. En las capas finales, se incluyó una capa Flatten seguida de dos capas densas con 64 filtros cada una, y una capa final con 6 filtros para la clasificación de las imágenes. Se utilizó Adam como optimizador con una tasa de aprendizaje de 0.0001, un tamaño de lote de 128 y con 110 épocas. En esta prueba se obtuvo una exactitud de 96%, precisión de 96%, recall de 96.44% y un f1- score de 95%.

En la segunda prueba se consideró cuatro bloques principales para la extracción de características, seguidas de capas de agrupación máxima para reducir el tamaño y capas Dropout para prevenir el sobreajuste. Para cada bloque se consideró dos capas convolucionales, de 32 filtros de tamaño 3x3 para el primer bloque, para el segundo de 64 filtros de 3x3, 128 filtros de 3x3 para el tercer bloque y 256 filtros de tamaño 3x3 para el cuarto bloque. Cada bloque está intercalado con capas MaxPoling para la reducción de tamaño y capas Dropout para prevenir el sobreajuste. Para todas las capas se consideró la activación relu. En las capas finales para la clasificación de la red se consideró una capa de aplanamiento de las capas convolucionales, seguida por una capa densa de 512 neuronas para la conexión de capas, una capa de salida con una tasa del 50% para evitar el sobreajuste y una capa densa de seis neuronas para la clasificación de imágenes. Se utilizó el optimizador Adam con una tasa de aprendizaje del 0.001, un tamaño de lote de 32 y 100 épocas para entrenar el modelo. En esta prueba se obtuvo una exactitud de 96.90%, precisión 96.94%, recall y f1-score de 96.90%.

Para la tercera prueba, en la extracción de características se consideró solo los tres primeros bloques, con filtros de 32, 64 y 128. Todo lo demás se dejó igual que la segunda prueba, excepto las épocas que establecieron en 50. Se logró una exactitud, precisión y

recall de 95.69 %, y f1 score de 95.68%.

En la cuarta prueba se dejó todas configuraciones de la tercera prueba excepto para las capas finales de clasificación. La capa densa se consideró 256 neuronas en vez de 512 y un Dropout de 25%. Además, se estableció 100 épocas para el entrenamiento. De esta prueba se obtuvo una exactitud de 91.68%, una precisión de 92.43%, recall de 91.68% y un f1-score de 91.53%.

En la quinta prueba se consideró los parámetros establecidos en la cuarta prueba, excepto la capa final densa se estableció a 512 neuronas, DropOut de 50% y una tasa de aprendizaje de 0.0001. Se obtuvo una exactitud de 97.33%, precisión de 97.32%, recall de 97.33% y un f1 – score de 97.32%.

Dado que se quería mejorar los resultados obtenidos, se propuso hacer un aumento con los datos de entrenamiento. Así que para la quinta prueba se dejó todas las configuraciones de la cuarta prueba, pero antes de pasar a compilar el modelo. A los datos de entrenamiento se hizo un aumento de datos, por cada imagen original se generó una imagen aumentada, considerando parámetros de rotación, desplazamiento horizontal y vertical de las imágenes. De esta prueba se obtuvo un 99.11% de exactitud , precisión 99.12%, recall y f1 score de 99.11%.

Así mismo para cada prueba se fue evaluando el consumo de recursos y tiempo de respuesta, tal y como se expresa en la siguiente tabla

Tabla 9: *resultados de las pruebas piloto realizadas*

Indicadores	Prueba 1	Prueba 2	Prueba 3	Prueba 4	Prueba 5	Prueba 6
Consumo de CPU	48.5%	58.2%	54.7%	88.6%	53.5%	63.4%
Consumo de memoria RAM	53%	53.5%	67.9%	57.5%	69.2%	49.5%
Tiempo de respuesta	59 minutos, 42 segundos	54 minutos, 47 segundos	27 minutos, 50 segundos	54 minutos, 39 segundos	55 minutos, 24 segundos	55 minutos, 51 segundos
Exactitud	96%	96.90%	95.69%	91.68%	97.33%	99.11%
Precisión	96%	96.94%	95.69%	92.43%	97.32%	99.12%
Recall	96%	96.90%	95.69%	91.68%	97.33%	99.11%
F1 - score	95%	96.90%	95.68%	91.53%	97.32%	99.11%

2.5. Procedimiento de análisis de datos

Este proceso se hizo teniendo en cuenta una serie de pasos, los que se detallan a continuación.

1. Primero se recopiló información sobre las deficiencias nutricionales, la manera de cómo

afecta a los cultivos y cómo podemos identificarlos visualmente.

2. Con la información obtenida se procedió a la recolección de las hojas de café, mediante observación directa considerando las deficiencias en estudio para la presente investigación.
3. Se elaboró el conjunto de datos, teniendo en cuenta unos protocolos para la adquisición de las imágenes, luego estas fueron etiquetadas de acuerdo a número y clase a la que pertenece.
4. Mediante el análisis documental de investigaciones que anteriormente se han llevado a cabo sobre temas relacionados a la presente investigación, se logró identificar los diversos técnicas y métodos para la clasificación de deficiencias nutricionales en hojas de café.
5. Se hizo la implementación del método propuesto considerando los trabajos de investigación que identificamos previamente.
6. En este paso se evalúa el modelo propuesto, para ello se consideró los indicadores propuestos en la presente investigación, los cuales se detallada a continuación.

Consumo de recursos

a. Consumo del CPU

Se calculó el porcentaje del consumo del CPU al momento de ejecutar determinada tarea. Para ello se utilizó la siguiente formula.

$$C_{cpu} = \sum_j^n \frac{cc_i}{n}$$

Donde:

C_{cpu}: Representa el consumo promedio de la Unidad Central de Procesamiento (CPU).

\sum_j^n = representa la operación de suma de los consumos de CPU, j es un índice que va desde 1 hasta n .

cc_i = Es el consumo de CPU para la instancia i del proceso

n = Representa el número total de instancias del proceso

b. Consumo memoria RAM

Se realizó cálculo del espacio volátil que ocupa un determinado proceso al momento de ejecutarse. Para ello se empleó la siguiente fórmula:

$$Cram = \sum_j^n \frac{cm_i}{n}$$

Donde

$Cram$: Representa el consumo promedio de memoria RAM.

\sum_j^n = representa la operación de suma de los consumos de memoria, j es un índice que va desde 1 hasta n .

cm_i = Es el consumo de memoria RAM para la instancia i del proceso

n = Es el número total de instancias del proceso consideradas en el cálculo del consumo promedio de memoria RAM.

c. Tiempo de respuesta

Se calculó el intervalo de tiempo transcurrido desde que empezó a ejecutarse hasta su término de un determinado proceso. Para ello se utilizó la siguiente fórmula:

$$Tr = \sum_j^n \frac{tf_j - ti_j}{n}$$

Donde:

Tr : Representa tiempo promedio de respuesta

\sum_j^n = representa la operación de suma de los tiempos de ejecución de cada proceso, j es un índice que va desde 1 hasta n .

tf_i = Es el tiempo final ejecución.

ti_i = Es el inicial de la ejecución.

n = Es el número total de ejecuciones.

Rendimiento del modelo

d. Exactitud

Esta métrica dio una medida general del modelo al determinar cuántas de sus predicciones fueron correctas en relación con el número total de predicciones realizadas. Se obtuvo dividiendo el número de predicciones correctas entre el total de predicciones realizadas. Se empleó la siguiente fórmula:

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

Donde:

TP : Verdaderos positivos.

TN : Verdaderos negativos.

FP : Falsos positivos.

FN : Falsos negativos.

e. Precisión

Esta métrica mostró la calidad de las predicciones positivas del modelo en comparación con todas las otras predicciones positivas. En pocas palabras, proporcionó la cantidad de predicciones positivas que fueron correctas en comparación con el número total de predicciones positivas que el modelo encontró.

Este valor mejoró la comprensión de la capacidad del modelo para encontrar casos positivos en el conjunto de datos. Se utilizó la fórmula siguiente:

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

Donde:

TP: Representa los verdaderos positivos.

FP: Representa los falsos positivos.

f. Recall

Esta métrica reveló la capacidad del modelo para identificar correctamente las instancias positivas en todo el conjunto de datos. Esto permitió evaluar cómo el modelo manejaba de manera específica los casos positivos y su capacidad para distinguir entre ellos. Se utilizó fórmula siguiente:

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

Donde:

TP: Representa los verdaderos positivos.

FN: Representa los falsos negativos.

g. F1 – score

Esta métrica proporcionó una evaluación integral del equilibrio y rendimiento general del modelo en la clasificación. No solo indicó la proporción de instancias correctamente clasificadas, sino que también ofreció una medida de la estabilidad y eficacia del modelo en diversos contextos. El cálculo permitió entender cómo el modelo se desempeñaba en la identificación de las clases positivas y negativas, lo cual resultó crucial para evaluar su capacidad para realizar predicciones precisas y equilibradas en diferentes escenarios. Para su cálculo se usó la siguiente fórmula:

$$F1 = 2 X \frac{P X R}{P + R}$$

Donde:

P: Representa a la Precisión.

R: Representa al recall.

2.6. Criterios éticos

Para la presente investigación, se ha considerado criterios éticos que garantizan la fiabilidad, validez y credibilidad de los resultados obtenidos. Los que se han tenido en cuenta son:

a. Confidencialidad

La información involucrada con terceros será debidamente tratada, de tal manera que garantice el total anonimato de sus datos, evitándose así cualquier tipo de infidencia.

b. Derechos de autor

La presente investigación cumple con los principios éticos de autoría de la información empleada, citando a las fuentes consultadas y referenciadas como tales, ya que el plagio es una práctica que no está incluida en los estándares para el presente trabajo.

c. Búsqueda del Bien

La presente investigación, en su esencia, persigue contribuir al beneficio colectivo al compartir los resultados obtenidos como un valioso aporte a la comunidad científica.

Así mismo se ha establecido criterios de rigor científico para garantizar la confiabilidad

de los métodos y técnicas empleados en la investigación actual y la credibilidad de los hallazgos.

a. Recopilación

Los procedimientos realizados en esta investigación serán descritos con gran detalle para que puedan ser utilizados y replicados por terceros como una herramienta para futuras investigaciones.

b. Confiabilidad

Los hallazgos expresados en la siguiente investigación serán confiables ya que sigue las condiciones planteadas para este documento, las cuales están basadas en instrumentos y técnicas estadísticas válidas y herramientas de recolección de datos electrónicos donde se evidencian las métricas de rendimiento de clasificación y los resultados sean los más cercanos posibles a los expuestos.

c. Validez

la información expuesta en el presente documento tendrá conclusiones obtenidas mediante la investigación, tratando de que sus variables sean las más precisas posibles.

III. RESULTADOS Y DISCUSIÓN

3.1. Resultados

El estudio se centró en desarrollar un método de clasificación automática de deficiencias nutricionales en hojas de cafeto utilizando procesamiento de imágenes digitales y aprendizaje profundo.

Para el desarrollo de esta investigación, se utilizó un computador portátil que cumplía con los siguientes requisitos: procesador Intel(R) Core (TM) i5-8250U CPU @ 1.60GHz con 8 núcleos, 12 gigabytes (GB) de memoria RAM y sistema operativo Windows 11 Home Single Language de 64 bits. Además, se implementó el modelo propuesto utilizando Google Colaboratory, una plataforma que permite trabajar con Notebooks en Python y facilita el almacenamiento de conjuntos de datos en Google Drive. Se llevaron a cabo evaluaciones de métricas establecidas en la operacionalización de variables de este estudio, tales como consumo de CPU, tiempo de respuesta, consumo de memoria RAM, exactitud, precisión, retorno y puntaje F1. Estos análisis se realizaron con el fin de medir el rendimiento del modelo propuesto y evaluar su eficacia en la clasificación automática de deficiencias nutricionales en hojas de cafeto.

Tabla 10: *Resultado del consumo de recursos informáticos del sistema y tiempo de respuesta utilizado para el procesamiento de las imágenes y entrenamiento del modelo propuesto*

Indicador	Procesamiento de imágenes	Entrenamiento del modelo
Consumo de CPU	64.1%	63.4%
Consumo memoria RAM	36.6%	49.5%
Tiempo de respuestas	8 minutos con 46 segundos	55 minutos con 51 segundos

Nota: los datos fueron calculados considerando las especificaciones técnicas del computador

descrito al inicio de esta sección.

Para procesar las 3666 imágenes en el conjunto de datos, se utilizó aproximadamente el 64.1% de la capacidad total de la CPU, lo que equivale a 1.0256 GHz. Además, se utilizó el 36.6% de la memoria RAM de 12 GB, lo que significa que se necesitaron aproximadamente 4.392 GB de memoria. En cuanto al tiempo total utilizado para esta tarea, fue de 8 minutos y 46 segundos. Estos datos proporcionan información importante sobre los recursos computacionales requeridos para llevar a cabo el procesamiento de imágenes en el estudio.

Durante el entrenamiento del modelo, se utilizó aproximadamente el 63.4% de la capacidad total de la CPU, lo que equivale a 1.0144 GHz. Además, se utilizó aproximadamente el 49.5% de la memoria RAM de 12 GB, lo que equivale a 5.94 GB. El tiempo total empleado para completar esta fase fue de 51 minutos y 51 segundos. Estos datos proporcionan una visión detallada del consumo de recursos computacionales durante el entrenamiento del modelo, lo que es crucial para evaluar la eficiencia y el rendimiento del algoritmo de aprendizaje automático utilizado.

Tabla 11: *Resultado del rendimiento global del modelo propuesto*

Indicador	Porcentaje
Exactitud	99.11%
Precisión	99.12%
Recall	99.11%
F1-Score	99.11%

Nota: Los datos anteriores fueron calculados usando el mismo conjunto de datos para entrenamiento y validación.

a. Exactitud (Accuracy)

La exactitud indica el porcentaje de predicciones correctas que el modelo hizo para todas las clases. Es decir, cuántas veces el modelo clasificó correctamente una imagen de hoja de café de acuerdo a la clase a la que verdaderamente pertenece. El modelo propuesto presentó una exactitud del 99.11%, lo que significa que acertó el 99.11% de las veces al clasificar imágenes de hojas de café, ya sea con deficiencias nutricionales o como hojas sanas. Esto sugiere que el modelo tiene un alto grado de precisión en su capacidad para distinguir entre hojas sanas y hojas con deficiencias nutricionales.

b. Precisión (Precision)

La precisión del 99.12% indica que casi siempre que el modelo hizo una predicción sobre si una hoja estaba sana o tenía una deficiencia, esa predicción fue correcta. En otras palabras, el 99.12% de las veces, el modelo acertó en su evaluación de la salud de las hojas o su presencia de deficiencia. Esto sugiere que el modelo tiene un alto nivel de confiabilidad en sus predicciones y es efectivo para distinguir entre hojas sanas y aquellas con deficiencias nutricionales. Esta alta precisión es un indicador de la calidad y el rendimiento del modelo en la tarea de clasificación de imágenes de hojas de café.

c. Recall:

Esta métrica indica cuántas hojas con deficiencias nutricionales pudo detectar correctamente el modelo en el conjunto de datos. El modelo propuesto tuvo un 99.11%, lo que indica que el modelo encontró casi todas las deficiencias nutricionales. Esto sugiere que el modelo es altamente efectivo para identificar y clasificar correctamente las deficiencias nutricionales en las hojas de café en el conjunto de datos evaluado.

d. F1-Score:

Esta métrica da una idea general del rendimiento del modelo al combinar la precisión y el recall en un solo valor. Con un 99.11%, el modelo demostró un buen equilibrio entre precisión y recall, lo que indica que puede identificar y clasificar de manera efectiva las deficiencias nutricionales.

Tabla 12: Resultado del rendimiento por clases del moldeo propuesto

Clase	Precisión	Recall	F1- Score
Fosforo (P) [0]	100%	99.09%	99.54%
Hierro (Fe) [1]	97.35%	100%	98.65%
Magnesio (Mg) [2]	100%	98.37%	99.18%
Nitrógeno (N) [3]	99.05%	99.05%	99.05%
Potasio (K) [4]	98.15%	99.07%	98.60%
Sanas [5]	100%	99.15%	99.57%

Nota: Los datos anteriores fueron calculados usando el mismo conjunto de datos para entrenamiento y validación.

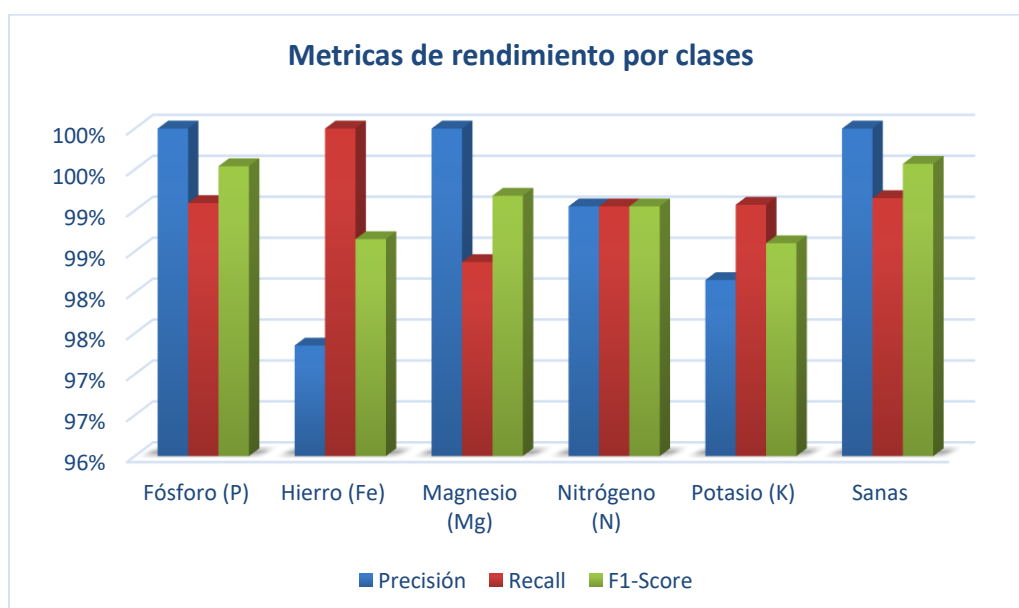


Fig. 22: Resultado del rendimiento por clases del moldeo propuesto

- **Fósforo**

El modelo encontró la deficiencia de fósforo con una precisión del 100%. Esto demuestra que todas las hojas que se pensaba que tenían deficiencias de fósforo en realidad lo tenían. De igual manera, su recall fue del 99.09%, lo que indica que el modelo identificó correctamente el 99.09% de todas las hojas con deficiencia de fósforo. Además, el puntaje F1-score del 99,54% de esta clase indica que el modelo logró equilibrar la precisión y la recuperación, lo que muestra que tiene un alto rendimiento en la clasificación de deficiencias de fósforo.

- **Hierro**

El modelo propuesto tuvo una precisión del 97.35% en la deficiencia de hierro, lo que significa que identificó correctamente alrededor del 97.35% de las hojas con esta deficiencia. En términos de recall, logró un 100%, lo que significa que pudo encontrar todas las hojas afectadas en el conjunto de datos. Además, obtuvimos un puntaje F1 del 98,65%, lo que demuestra un buen equilibrio entre precisión y recall en la clasificación de deficiencia de hierro.

- **Magnesio**

El modelo propuesto para la deficiencia de magnesio alcanzó una precisión total 100%, lo que significa que todas las hojas identificadas como deficientes en magnesio realmente lo eran. Además, logró un recall del 98,37%, lo que indica que identificó correctamente el 98,37% de todas las hojas del conjunto de datos con deficiencia de magnesio. El puntaje F1 de esta clase fue del 99.18%, lo que demuestra que el modelo es muy efectivo para clasificar la deficiencia de magnesio.

- **Nitrógeno**

En cuanto a la deficiencia de nitrógeno, el modelo demostró una precisión y recall del 99.05%. Esto significa que aproximadamente el 99.05% de las hojas identificadas como deficientes en nitrógeno realmente lo eran, y que el modelo reconoció correctamente el 99.05% de todas las hojas con esta deficiencia. Además, el puntaje F1 para esta clase fue del 99.05%, lo que destaca un rendimiento equilibrado y efectivo en la clasificación de esta deficiencia.

- **Potasio**

El modelo logró una precisión del 98,15% para la deficiencia de potasio, lo que significa que alrededor del 98,15% de las hojas identificadas como deficientes en potasio realmente lo estaban. Su recall fue del 99,07 %, lo que indica que el modelo identificó correctamente el 99,07 % de todas las hojas con deficiencia de potasio. Además, el puntaje F1 de esta clase fue del 98.60%, lo que demuestra un desempeño sólido y equilibrado en la detección de esta deficiencia.

- **Sanas**

En cuanto a las hojas sanas, el modelo mostró un alto desempeño, logrando una precisión del 100%, lo que significa que todas las hojas que se consideraban sanas lo estaban en todos los casos. Además, su recall fue del 99,15%, lo que indica que identificó casi todas las hojas sanas. El cuanto al F1- Score fue del 99.57%, lo que demuestra la excelente capacidad del modelo para detectar hojas sanas.

Después de haber obtenido los resultados anteriores, comparamos nuestro propio modelo con otros modelos ampliamente reconocidos en el campo, como la red VGG16, ResNet50, InceptionV3 y MobileNet. Este paso representó un paso importante en el contexto de esta investigación, ya que nos permitió evaluar la validez y el rendimiento de

nuestro modelo frente a las referencias establecidas en la literatura científica. Además, nos ayudó a validar y contextualizar los logros alcanzados, agregando así una nueva perspectiva al conocimiento existente sobre la clasificación de las deficiencias nutricionales en las hojas de café.

Tabla 13: Comparación de resultados del modelo CNN propio con otras arquitecturas preexistentes

Indicadores	VGG16	InceptionV3	MobileNet	Modelo CNN propio
Exactitud (Accuracy)	95.39%	95.57%	98.22%	99.11%
Precisión	95.62%	95.57%	98.28%	99.12%
Recall (Exhaustividad)	95.39%	95.57%	98.22%	99.11%
F1-Score	95.42%	95.56%	98.28%	99.11%

Nota: Los datos anteriores fueron calculados usando el mismo conjunto de datos para entrenamiento y validación.

Cuando probamos la red VGG16, observamos unos resultados bastantes aceptables, obteniendo una exactitud del 95.39%, una precisión del 95.62%, un recall del 95.39% y un F1-Score del 95.42%. En términos generales se puede decir que estos resultados fueron bastantes prometedores, sin embargo, es importante tener en cuenta que tuvo un número considerable de errores.

Asimismo, al evaluar la red InceptionV3, vimos que también tuvo un buen desempeño, con resultados similares a la red VGG16, obteniendo así una exactitud del 95.57%, con una precisión y recall del 95.57%, y un F1-Score del 95.56%. esto significa

que sus resultados son alentadores para nuestro caso de estudio propuesto, sin embargo, no son perfectos, por lo que se podría hacer mejoras.

Por otro lado, al probar la red MobileNet, obtuvimos mejores resultados que las dos redes anteriores, logrando una exactitud y recall del 98.22%, una precisión y un F1-Score de 98.28%. Esto mostró que utilizar esta arquitectura podríamos obtener buenos resultados precisos y confiables.

A pesar de que la arquitectura de MobileNet demostró ser muy exitosa. Con nuestro modelo CNN propio, se obtuvieron resultados aún mejores, con una precisión del 99,11%, una precisión del 99,12%, un recall del 99,11% y un F1-Score del 99,11%. Estos resultados se destacan significativamente en comparación con los otros modelos evaluados, demostrando que nuestro método para clasificar deficiencias nutricionales en hojas de café es efectivo y superior.

3.2. Discusión

En cuanto al consumo de recursos y tiempo de respuesta

Observamos que el procesamiento de las imágenes requiere un tiempo relativamente corto en comparación con el entrenamiento del modelo, lo que indica una eficiencia razonable en esta etapa. Sin embargo, también podemos notar que el consumo de recursos de la CPU y la memoria RAM es un poco elevado, lo que sugiere que podría haber margen para optimizar estos aspectos y mejorar aún más la eficiencia del proceso.

El tiempo de respuesta para el entrenamiento del modelo es considerablemente mayor que el procesamiento de las imágenes. Esto puede deberse a la complejidad del

modelo de aprendizaje profundo, así como también a la cantidad y tamaño utilizado de datos de entrenamiento. Por lo que es importante analizar detalladamente con el fin identificar posibles cuellos de botella y áreas de mejora para esta etapa.

Aunque el entrenamiento requiere muchos recursos y tiempo, también es crucial para lograr un modelo preciso y confiable. Por lo que resulta vital hallar un balance entre un buen rendimiento del modelo y su eficacia en el entrenamiento. En tal sentido consideraríamos explorar diversas técnicas de optimización, tales como probar con arquitecturas pre-entrenadas y ajuste de hiperparámetros, de tal manera que se mejore la eficiencia en el entrenamiento sin comprometer el rendimiento del modelo.

A medida que el conjunto de datos o la complejidad del modelo crezcan, es probable que los requisitos de recursos informáticos y el tiempo de respuesta también aumenten. Por lo que se requeriría hacer pruebas adicionales con conjuntos de datos más grandes para mejorar la eficiencia en situaciones de producción.

En cuanto al rendimiento general del modelo

El modelo fue extremadamente preciso y sensible al detectar deficiencias nutricionales en las hojas de café, como lo demuestran los altos valores de precisión, recall y puntaje F1. Esto demuestra que es una herramienta útil para clasificar de manera confiable estas deficiencias. Además, la alta precisión y la repetición indican que el modelo logró identificar la mayoría de las deficiencias nutricionales en el conjunto de datos al reducir los falsos positivos, es decir, la clasificación incorrecta entre las clases sugeridas. Esto confirma la utilidad práctica del modelo para identificar problemas nutricionales precisos en las plantas de café.

El alto rendimiento del modelo indica que podría ser una herramienta útil para

agricultores y agrónomos para encontrar deficiencias nutricionales en las plantas de café. Esto podría llevar a intervenciones más rápidas y efectivas, mejorando la salud y el rendimiento de los cultivos. A pesar de estos hallazgos, es necesario continuar evaluando el modelo en diferentes condiciones para garantizar que sea robusto y pueda aplicarse a una variedad de escenarios agrícolas. Además, se debería probar con otras deficiencias nutricionales porque solo se han considerado cinco y una clase de hojas sanas para la presente investigación.

En cuanto al rendimiento del modelo por clases

Al examinar las métricas de precisión, notamos que el modelo tuvo un rendimiento destacado en la detección de la deficiencia de fósforo. Sin embargo, en el caso del hierro, se encontraron los resultados más bajos con 97.35%. Esto podría indicar que aún existe la necesidad de realizar algunos pequeños ajustes adicionales en futuras mejoras del modelo.

En cuanto al recall, muestra qué tan bien puede el modelo clasificar todas las muestras reales de una clase, por lo que es notable la capacidad del modelo para la identificar todas las hojas con deficiencia de hierro, logrando así un 100%. Sin embargo, en el caso de las deficiencias de magnesio, aunque se obtuvo un valor alto de 98.77%, el resultado indica que no se logró identificar correctamente todas hojas, pero los errores son mínimos y se podría hacer algunos ajustes.

Finalmente, al analizar el F1- Score, que combina precisión y recall en una sola métrica, notamos un equilibrio generalmente sólido en todas las clases. Destaca la clase de magnesio con un 99.18%, lo que señala un excelente equilibrio para esta deficiencia. Por otro lado, con la deficiencia que se obtuvo los resultados más bajos para esta métrica fue con el potasio con un 98.60%, si bien es cierto que son buenos resultados se puede ajustar algunas cuestiones.

En comparación del modelo propio con otras arquitecturas preexistentes

El modelo CNN propio superó a los modelos preexistentes como la red VGG16, InceptionV3 y MobileNet, demostrando así una eficacia notable en la clasificación de deficiencias nutricionales en hojas de café. Este logro podría tener un impacto significativo en la agricultura al facilitar a los agricultores la identificación de deficiencias de nutrientes en sus cultivos.

En comparación con otras investigaciones

En el estudio realizado por Lewis y Espineli [15], titulado, *Classification And Detection Of Nutritional Deficiencies In Coffee Plants Using Image Processing And Convolutional Neural Network (CNN)*, en la que se propusieron implementar un prototipo basado en procesamiento de imágenes y redes neuronales convolucionales para la detección y clasificación de manera automática las carencias nutricionales en hojas de cafeto obtuvieron una exactitud 91,49%. En contraste a ello, en la presenta investigación se ha alcanzado una exactitud del 99.11% con el modelo propio propuesto, con la arquitectura MobileNet 98.22%, InceptionV3 95.57% y 95.39% con la red VGG16. Estos resultados demuestran que los modelos de clasificación propuestos en esta investigación ofrecen un rendimiento óptimo en comparación con la técnica previamente propuesta.

Asimismo, en la investigación realizada por Ezat et al. [16], titulada, *Multi-class Image Classification Using Deep Learning Algorithm*. Se plantearon determinar el rendimiento del aprendizaje profundo utilizando CNN para la clasificación de imágenes multiclase pre entrenado con el conjunto de datos ILS VRC y probado con otro conjunto de datos más pequeños PASCAL VOC 2007. Obtuvieron una exactitud de 90.81 con la arquitectura CNN y un 65.27% con la clasificación super vectorial. En comparación a esos resultados, en esta investigación se ha obtuvo una exactitud de una exactitud del 99.11% con el modelo propio propuesto, con la arquitectura MobileNet 98.22%, InceptionV3 95.57% y 95.39% con la red

VGG16. demostrando de así que los resultados que se obtuvieron son óptimos con todos los modelos de calificación propuestos.

En otro estudio realizado por Soo y Ngadiran [13], titulado, Plant Diseases Classification using Machine Learning. Se propusieron implementar un modelo de Extreme Learning Machine (ELM), para clasificar enfermedades de las plantas mediante la evaluación de sus hojas. Los resultados mostraron que el modelo tiene una exactitud que oscila entre el 69 y el 84,94%. Mientras que en la presente investigación se ha obtenido una exactitud del 99.11% con el modelo propio propuesto, con la arquitectura MobileNet 98.22%, InceptionV3 95.57% y 95.39% con la red VGG16. Estos hallazgos demuestran la efectividad de los modelos propuestos en esta investigación en comparación con los resultados expuestos anteriormente.

De igual manera A. Khattak et al. [14], en su investigación titulada, Automatic Detection of Citrus Fruit and Leaves Diseases Using Deep Neural Network Model. Plantearon desarrollar un método usando en redes neuronales para la detectar de manera automatizada enfermedades en las hojas de cítricos. Los resultados obtenidos mostraron que el modelo CNN propuesto alcanzó de rendimiento considerable, logrado así un 94.55% de exactitud. Sin embargo, en esta investigación se obtuvo una exactitud de 99.11% con el modelo propio propuesto, con la arquitectura MobileNet 98.22%, InceptionV3 95.57% y 95.39% con la red VGG16. Demostrando de esta manera que se ha obtenido resultados óptimos con los modelos todos los modelos planteados en el presente estudio.

Así también en el estudio realizado por Tran et al. [19], titulada, A Comparative Study of Deep CNN in Forecasting and Classifying the Macronutrient Deficiencies on Development of Tomato Plant. Se plantearon implementar modelos de redes neuronales artificiales para reconocer y clasificar las deficiencias nutricionales en plantas de tomate. Los resultados mostraron una tasa de exactitud del 87,27% para el modelo Inception-ResNet v2, y para el

modelo Autoencoder 79,09%. En cambio, en la presente investigación se obtuvo una exactitud del de 99.11% con el modelo propio propuesto, con la arquitectura MobileNet 98.22%, InceptionV3 95.57% y 95.39% con la red VGG16. Esto demuestra que se ha logrado resultados excelentes con todos los modelos de clasificación propuestos.

Aporte práctico

La inteligencia artificial ha experimentado un crecimiento exponencial en los últimos años y su aplicación en una amplia gama de campos está transformando rápidamente varios aspectos de nuestra vida cotidiana. El objetivo principal de este trabajo de investigación fue proponer un método para la clasificación automática de deficiencias nutricionales en hojas de café. Este método se basa en el procesamiento de imágenes digitales y el aprendizaje profundo. Con este firme propósito se planteó abordar el método de la siguiente manera.

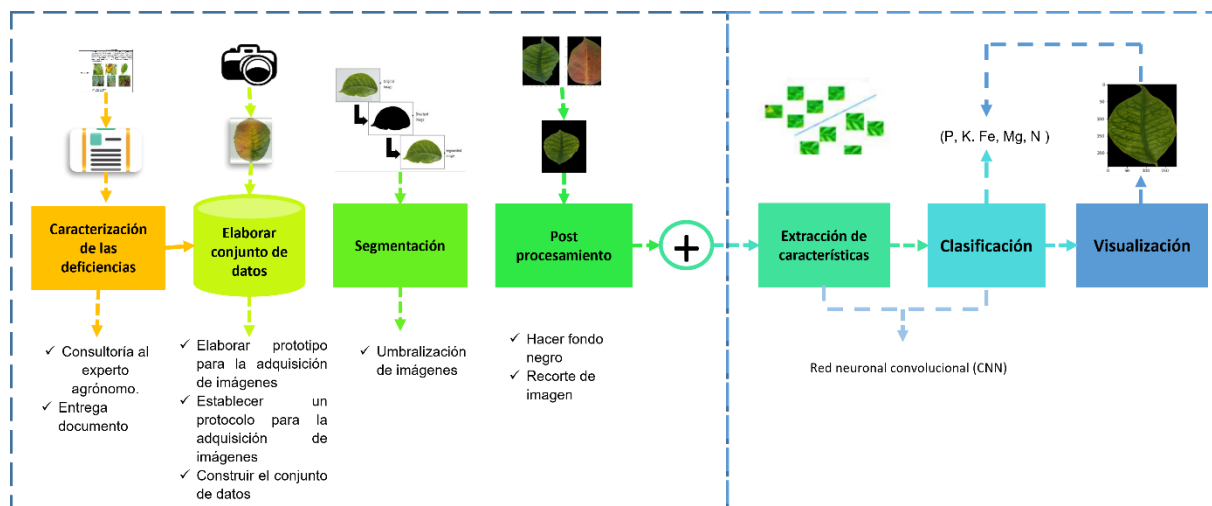





Fig. 23: Método propuesto. Fuente: Elaboración propia.



1. Caracterización de las deficiencias nutricionales de la hoja del café a nivel visual.







Para la presente investigación, primero fue necesario realizar la caracterización de










las deficiencias nutricionales del cafeto a nivel visual, para ello se hizo una consultaría a ingeniero agrónomo experto en cultivos de cafeto, quien nos brindó la información necesaria sobre cómo identificar las distintas clases de deficiencias que existen, el cual, se respaldó mediante documentación, y nos servirá como pase para todo el desarrollo de nuestro proyecto, los resultados obtenidos son tal y como se muestra a continuación.







Tabla 14: Características de las deficiencias nutricionales para identificarlos visualmente







Deficiencias	Síntomas visibles de las deficiencias nutricionales en las hojas de cafeto		
Nitrógeno (N)	Principalmente sus indicios se muestran en hojas jóvenes. Estas presentan un color amarillento verdoso que tiende a claro cubriendo por completo la hoja, en algunas ocasiones presentan una ligera clorosis en la vena que formando así tipo red. Las hojas que directamente están expuestas al sol se tornan de color amarillo rápidamente. Si la deficiencia persiste, la decoloración se extiende a todo el follaje y posteriormente presentaran necrosis, que conducirá la caída prematura defoliación foliar.		
			
Potasio (K)	Las hojas más adultas presentan clorosis o amarillamiento, que se expande de los márgenes hacia la base. Cuando la deficiencia es muy severa, presentan áreas necróticas (muertas) en forma de v invertida. Produce bajo contenido en azúcares, los frutos tienen bajo peso, calidad		

	y tamaño. Sobre todo, hace que la planta se muy susceptible a enfermedades.
	
Fósforo (P)	<p>Las hojas viejas presentan un moteado clorótico entre las venas, en las hojas bajas suele aparecer puntos necróticos, que posteriormente presentan una coloración marrón oscuro. Las hojas jóvenes se mantienen en una tonalidad verde oscura. Cuando la deficiencia es severa las manchas son amarillas rojizas de diversos tamaños, lo que produce la caída total de las hojas, especialmente de las ramas con los frutos en etapas de maduración. Afecta principalmente a la raíz de la planta, la cual dificulta el crecimiento y el correcto desarrollo de los frutos.</p> 
Magnesio (Mg)	<p>Clorosis amarillo verde a naranja o bronce que comienza cerca de la vena central y aumenta entre las venas La hoja tiene un aspecto moteado porque un pequeño espacio verde permanece al lado de las venas laterales y centrales. Dentro del tejido clorótico, pueden aparecer manchas necróticas irregulares y marcadas. Con frecuencia, pueden aparecer pequeños puntos hundidos y anaranjados en las nervaduras o</p>

	<p>en los márgenes de las hojas. Los síntomas mencionados comienzan en hojas antiguas, pero desaparecerán si la falta persiste.</p>		
			
<p>Calcio (Ca)</p>	<p>Los síntomas comienzan en las hojas jóvenes y son más notables por la clorosis irregular, una costura verde amarillenta a lo largo del margen que se extiende hacia la vena central y crea bordes en forma de sierra. En ocasiones, los bordes de las hojas pueden desarrollar características ondulares, lo que significa que las hojas se enrollan hacia abajo y terminan en forma de gancho.</p>		
			
<p>Azufre (S)</p>	<p>Se pueden observar puntos necróticos marrones cerca del margen interveinal de la hoja a medida que avanza la deficiencia. Las hojas jóvenes experimentan una clorosis interveinal de color amarillo tendiendo a verde que luego se uniformiza en toda la hoja. El crecimiento disminuye significativamente.</p>		

			
Hierro (Fe)	<p>Las hojas jóvenes adquieren un tono verde pálido y, con el tiempo, las deficiencias se vuelven cloróticas, lo que da como resultado un tono amarillo claro a casi blanco. Las venas permanecen verdes, y solo en situaciones extremas toda la hoja se vuelve blanca. Las hojas antiguas rara vez se ven afectadas.</p>		
			
Manganeso (Mn)	<p>Aunque las plantas parecen atrofiadas, sus hojas siguen siendo grandes. Las hojas jóvenes están decoloradas de verde olivo a amarillo. A distancia, la planta tiene una palidez uniforme, pero es común un moteado fino. La falta de Mn normalmente mantiene las venas verdes y el área intercostal se vuelve clorótica. Sin embargo, también se ha registrado una alteración en la tonalidad de las venas, lo que contrasta con el tejido intermedio. La superficie de las hojas puede tener puntos amarillos o marrones, y las más afectadas se marchitan de color marrón.</p>		
			

Zinc (Zn)	<p>Las hojas se vuelven delgadas y largas, casi como cordones. Las venas de las hojas parecen una malla verde sobre un fondo pálido de verde. Las deficiencias de Zn se caracterizan por entrenudos más cortos y el crecimiento de múltiples brotes en cada nudo. En condiciones extremas, las hojas se vuelven quebradizas, con forma de lanza y casi blancas.</p>		
			
Boro (B)	<p>La planta muestra signos de deficiencias en las hojas jóvenes y la punta de crecimiento. Las hojas tienen punteos y un tono verde claro a bronce. Las hojas recién cortadas tienen forma asimétrica, irregulares márgenes y puntos necróticos. Los brotes terminales desaparecen gradualmente. El café producirá ramas secundarias que producirán un arbusto deforme como un abanico.</p>		
			
Cobre (Cu)	<p>Las hojas maduras se enrollan hacia abajo, especialmente en la parte inferior del cafeto. A pesar de que las hojas permanecen turgentes o firmes, esto hace que las plantas parezcan marchitas. Los brotes recién emergidos se ven flácidos si la deficiencia continúa. Las hojas jóvenes pueden deformarse y tener una vena central en forma de S. Un moteado clorótico difuso se desarrolla en las hojas maduras. El crecimiento sufre</p>		

	una fuerte atrofia, con entrenudos cortos, lo que da como resultado plantas densas y pequeñas. No afecta el tamaño de las hojas.		
			
Molibdeno (Mo)	La clorosis amarilla con tendencia a verde se desarrolla entre las venas y afecta el borde de las hojas. Las hojas de mediana edad tienen puntas negras. Las hojas se forman en forma de bóveda. También se pueden ver hojas malformadas y alargadas que se enrollan hacia abajo hasta que los márgenes se unen por debajo de la vena central. Los primeros signos se pueden ver en las hojas antiguas.		
			

Nota: la tabla anterior fue proporcionada por el experto agrónomo

2. Elaborar el conjunto de datos de acuerdo con las características de las deficiencias nutricionales de las hojas de cafeto

Una vez caracterizado las deficiencias nutricionales, se construyó un prototipo para la adquisición de las imágenes en ambiente controlado, el cual permitió ajustar la luminosidad, eliminar el brillo y posibles sombras. Los materiales utilizados para su elaboración fueron: doce varillas de metal de cuarenta centímetros de longitud y uno de

diámetro, tela adhesiva color blanco, cartulina blanca, una pistola de silicona, y dos focos led de 30 watts cada uno. A continuación, se muestra el modelado del prototipo propuesto en 3D, construido en software AutoCAD.

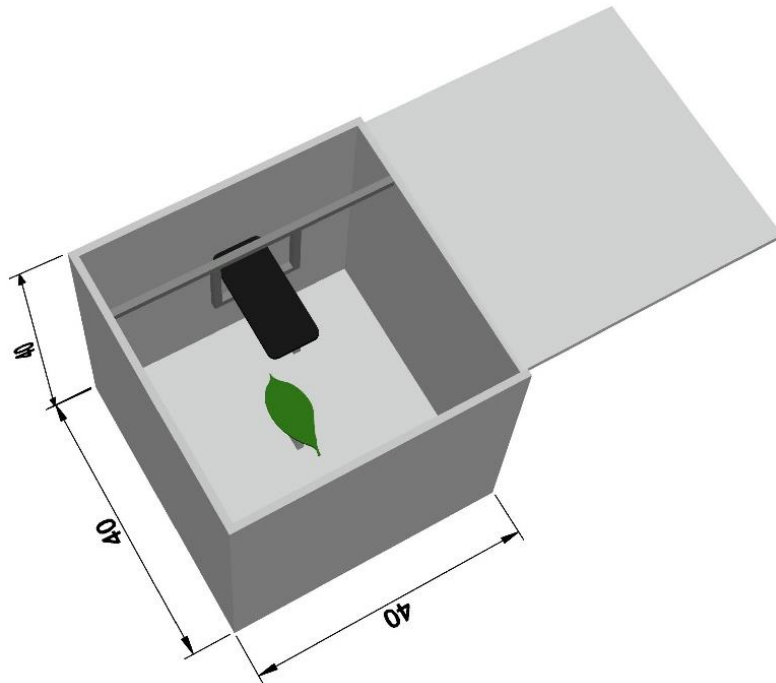


Fig. 24: Vista superior prototipo propuesto para la adquisición de las imágenes. Fuente elaboración propia.

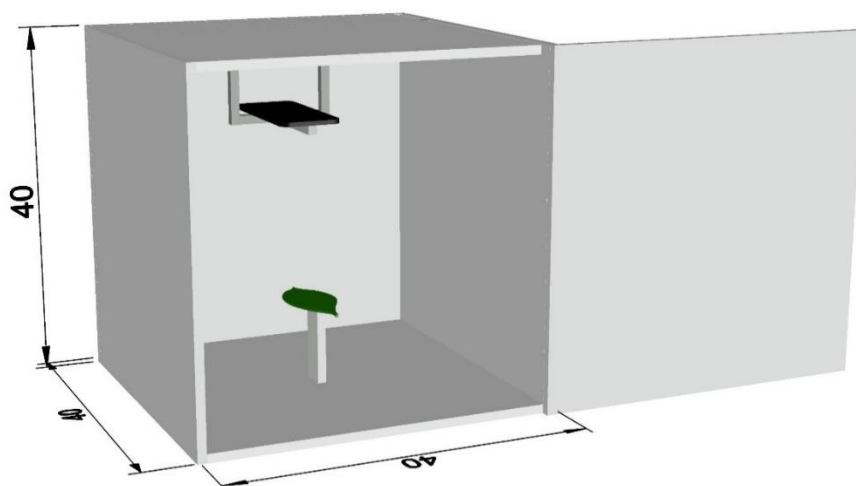


Fig. 25: Vista frontal del prototipo propuesto para la adquisición de las imágenes. Fuente elaboración propia.

A continuación, se muestra el prototipo implementado, teniendo en cuenta las consideración y especificaciones de los materiales y modelado anteriormente mencionados.



Fig. 26: Prototipo construido para la adquisición de las imágenes

En la etapa siguiente, se elaboró un protocolo detallado para la captura de las imágenes, con el objetivo de determinar los mejores parámetros de adquisición y que garantizaran la calidad y consistencia de los datos visuales obtenidos. Esto se estableció de la siguiente manera:

- Los focos LED a una distancia de 25 cm del prototipo, uno a cada extremo, el cual proporcionó la iluminación adecuada para captura de las imágenes.
- La hoja se colocó en una posición plana sobre una base diseñada específicamente para evitar sombras, para esta base se estableció una dimensión de 8x1 cm y de material transparente.
- La cámara se posicionó en la parte superior del prototipo, a una distancia de 12 cm

de la hoja, enfocando hacia abajo.

- Se utilizó un celular Red mi Note 10 Pro con su cámara principal de 108 MP, configurada en modo profesional y con la vista en cuadrícula activada. La vista en cuadrícula se estableció con el fin de evitar puntos de enfoque de la cámara.
- Adicionalmente, se ajustaron algunos otros parámetros de la cámara, esto incluyó el uso del lente wide, un fotograma de 1.9, un tiempo de exposición de 1/50s, una distancia focal de 6 mm, una apertura máxima de lente de 1.85 y una longitud focal de 25 mm.
- En cuanto a la captura de las imágenes se hizo de manera remota, ya que el celular se incluyó de manera fija en el interior del prototipo cerrado, esto con el fin de evitar en los posibles ruidos en las imágenes, como el aumento de luminosidad, sombra o distorsión de las imágenes. Se utilizó la aplicación AnyDesk para controlar el celular desde una computadora. Tal como se muestra a continuación.

Vista del celular



Vista desde la computadora

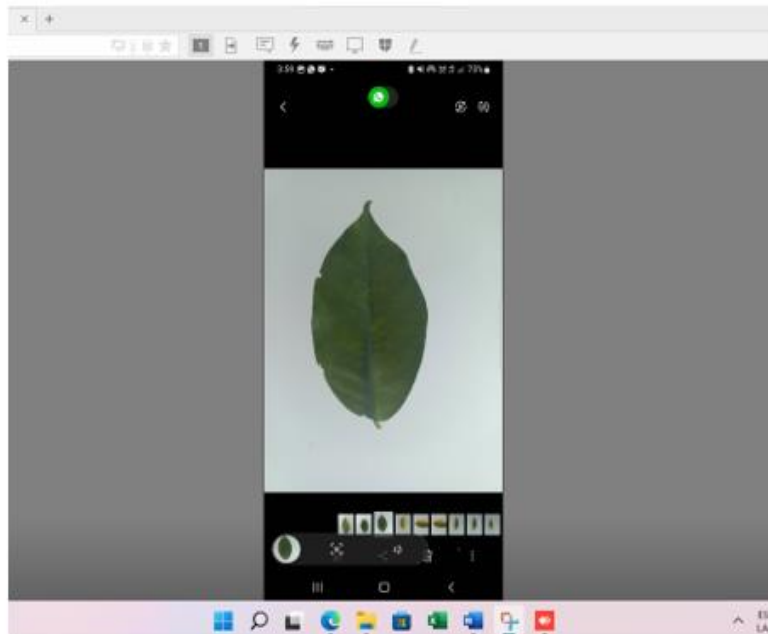


Fig. 27: Vista celular controlado por un computador de manera remota utilizando la aplicación AnyDesk

Una vez establecido el protocolo de adquisición, se tomaron 3366 imágenes, 561 para las deficiencias de fósforo (P), 561 para las deficiencias de potasio (K), 561 para las deficiencias de nitrógeno (N), 561 para las deficiencias de Magnesio (Mg), 561 para hojas sanas y 561 para las deficiencias de hierro (Fe). Posteriormente con las imágenes ya adquiridas se pasó a elaborar el conjunto de datos llamado NDCL, dentro del cual se incluyó un subdirectororio llamado train el cual contiene las seis clases propuestas anteriormente. Tal como se observa a continuación.

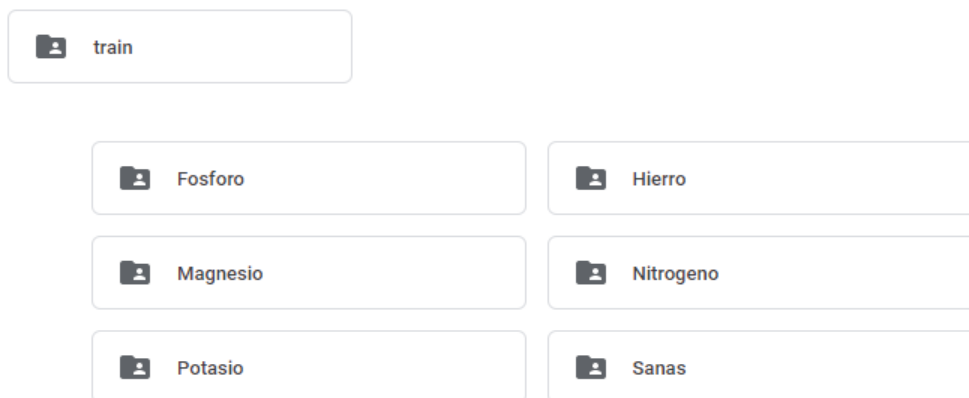


Fig. 28: Organización de los directorios para el conjunto de datos

Para cada clase, las imágenes fueron etiquetadas haciendo referencia al número y a la clase a la que pertenecen. De la siguiente manera.

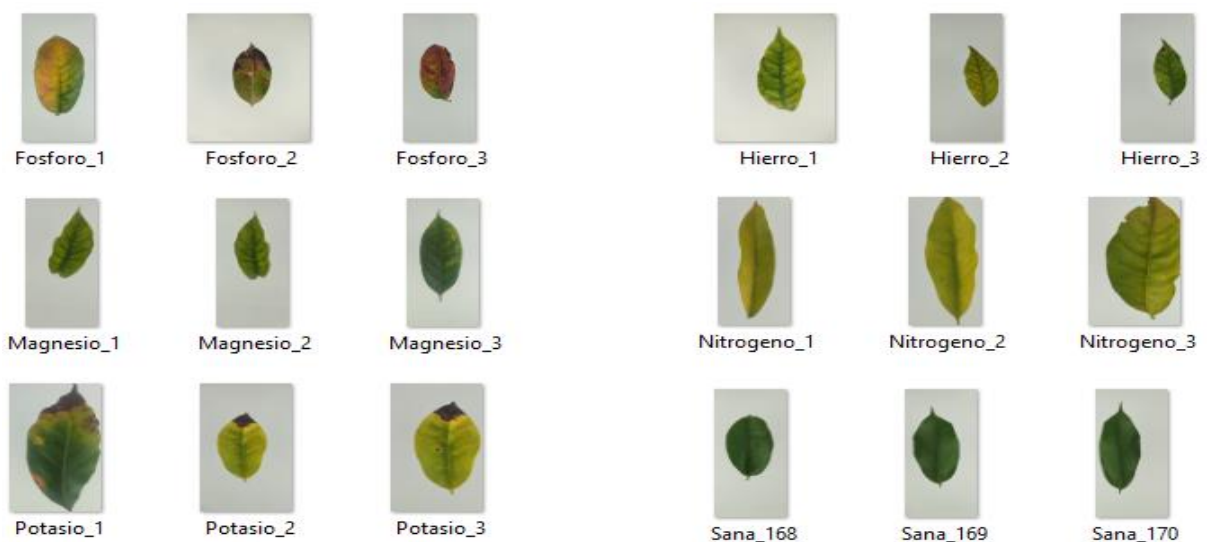


Fig. 29: Etiquetado de las imágenes, considerando el número y clase a la que pertenecen

Adicionalmente, se tomaron 666 imágenes, mezcladas entre todas clases propuestas. Estas se incluyeron en otro subdirectorio de nuestro conjunto de datos, al cual se le llamo test. Asimismo, estas imágenes no fueron etiquetas, con el fin de que sean utilizadas para realizar las pruebas de nuestro modelo planteado. Tal como se muestra a continuación.

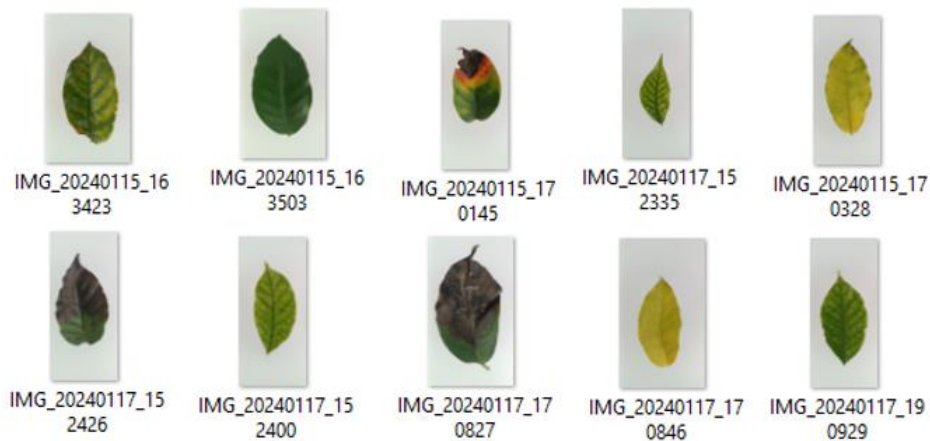


Fig. 30: *Imágenes del subconjunto test, no etiquetadas*

3. Realizar el procesamiento digital de las imágenes de las hojas de café

Durante esta fase, se realizaron varias transformaciones en las imágenes del conjunto de datos con el fin de prepararlas adecuadamente para la clasificación de deficiencias nutricionales en hojas de café. Para llevar a cabo este proceso, se empleó el entorno de Google Colaboratory, el cual facilitó la escritura y ejecución del código en Python directamente desde el navegador web. En primera instancia se cargaron las librerías necesarias para poder realizar el procesamiento. Asimismo, se cargó el subconjunto de datos llamada train y se hizo la lectura de la ruta de las clases esta que esta contiene y utilizando la librería opencv se asignó a una variable de donde se lee todas las imágenes. Dado que las imágenes tienen un tamaño excesivamente grande, estas fueron redimensionadas a un tamaño considerable de 900x1800 píxeles, esto con el fin de mantener en lo posible la información que estas contienen. Seguido de hizo la conversión de color RGB

a escala de grises, con objetivo de simplificar el procesamiento, reduciendo la información de colores a una sola dimensionalidad.

A continuación, se realizó la segmentación de las imágenes, utilizando la segmentación por umbralización. Para ello se usó el método Otsu, que es una técnica de umbralización automática, en la que se determina el umbral óptimo para separar los píxeles en dos clases, generalmente basándose en la distribución de los niveles de gris en la imagen. Para este caso, se utilizó la función `cv2.threshold` para binarizar la imagen, es decir convertir la imagen en blanco y negro y junto con el método `cv2.THRESH_OTSU` se calculó automáticamente el umbral óptimo. También con la función `cv2.THRESH_BINARY_INV`, se invirtió los colores de modo que los píxeles que superen el umbral se conviertan en negros y los que estén por debajo sean blancos. Después se usó operaciones morfológicas de apertura y cierre. Para la apertura se utilizó la función `cv2.MORPH_OPEN`, con el fin de eliminar pequeños detalles y discontinuidades en los objetos blanco y para el cierre `cv2.MORPH_CLOSE` para eliminar pequeños agujeros en los objetos blancos y conectar componentes cercanos. En ambas operaciones se utilizó un elemento de estructura elíptica de 5x5 denominado `cv2.getStructuringElement`, con cuatro interacciones para cada una. Posteriormente se aplicó la máscara binaria a la imagen original, esto se hizo usando la función `cv2.bitwise_and`, que conserva solo los píxeles de la imagen original donde la máscara es 1 (blanco), y el resto se vuelve negro. Dando como resultado la imagen original con el fondo eliminado. Todo esto, tal como se ve reflejado en el siguiente código.

```
# Binarización con el método Otsu y aplicando morfología para limpiar ruidos
tresh = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)[1]
kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (5, 5))
otsu = cv2.morphologyEx(tresh, cv2.MORPH_CLOSE, kernel, iterations=4)
otsu = cv2.morphologyEx(otsu, cv2.MORPH_OPEN, kernel, iterations=4)

# Aplicar máscara para quitar fondo
image_mask = cv2.bitwise_and(img, img, mask=otsu)
```

Fig. 31: Segmentación de las imágenes por el método de umbralización

El resultado después de aplicar las operaciones anteriormente expuestas se refleja de la siguiente manera.

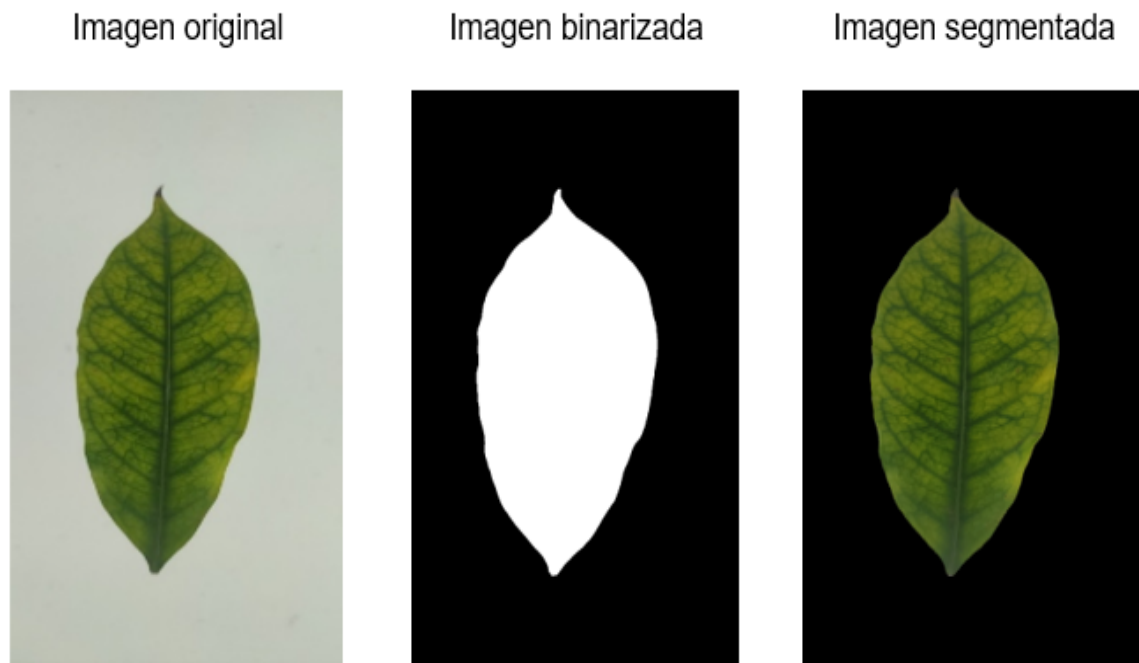


Fig. 32: Representación del proceso de segmentación de las imágenes por el método de umbralización

Después de observar la imagen segmentada, se pudo deducir que tiene áreas que no nos sirven, ya que nuestro objeto de interés es solo la hoja, para ello, se hizo la extracción del área de interés (ROI). Para ello primero se encontró los contornos de la imagen, luego, se calculó el rectángulo delimitador para cada contorno y se selecciona el más grande. Finalmente, se recorta la imagen original según las coordenadas de este rectángulo, obteniendo así la región de interés, que en esta investigación es la hoja de cafeto. Una vez que las imágenes están recortadas de acuerdo con el tamaño de la hoja, se pasó a hacer un redimensionamiento debido que las imágenes tienen un tamaño muy grande para ser pasado a la etapa de clasificación, para ello se utilizó la función `resize` de `opencv`, donde se consideró un alto de 240 por un ancho de 128 pixeles

```

####ROI >>> Extraccion de la region de interes (hoja)#
contours,hierachy=cv2.findContours(otsu,cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
cnt = contours[:]
num = len(cnt)

box = np.zeros((num,4))
for j in range(0,num):
    box[j,:] = cv2.boundingRect(cnt[j])
L = np.zeros((num,4))
Max=[0, 0]
for j in range(0, num):
    L[j,:] = box[j]
    if L[j, 2]> Max[1]:
        Max = [j, L[j,2]]
BOX = box[Max[0],:]

#b =BOX[BOX[1]:BOX[1]+BOX[3],BOX[0]:BOX[0]+BOX[2],:]
imagen_ROI = image_mask[int(BOX[1]):int(BOX[1]+BOX[3]), int(BOX[0]):int(BOX[0]+BOX[2])]

```

Fig. 33: Código para obtener la región de interés (ROI)

Como resultado de aplicar el código anterior se obtiene los siguiente.



Fig. 34: Imagen recortada de acuerdo con el área de interés (ROI) y redimensionada a un tamaño de 128x240 pixeles.

4. Implementar un modelo propio de aprendizaje profundo para clasificar las deficiencias nutricionales de las hojas del cafeto y comparar los resultados con otros modelos.

Una vez que las imágenes ya han sido procesadas y almacenadas en formato array y etiquetadas debidamente, se tomó el rango con el total de imágenes para barajarlas aleatoriamente para que el entrenamiento sea mucho mejor. Seguido se estableció las

variables objetivo del conjunto de datos, considerando 80% para entrenamiento y 20% para validación. Tal como se observa a continuación

```

1 imagenes = np.array(imagenes)
2 np.save("imagenes",imagenes)
3 np.save("labeles",labeles)
4
5 image = np.load("imagenes.npy", allow_pickle=True)
6 labeles = np.load("labeles.npy", allow_pickle=True)
7
8 imagen_shape = np.arange(image.shape[0])
9 np.random.shuffle(imagen_shape)
10 image = image[imagen_shape]
11 labeles = labeles[imagen_shape]
12
13
14 x_train, x_test = image[(int)(0.2*len_data):], image[:(int)(0.2*len_data)]
15 y_train, y_test = labeles[(int)(0.2*len_data):], labeles[:(int)(0.2*len_data)]
16
17 y_train = keras.utils.to_categorical(y_train, numero_clases)
18 y_test = keras.utils.to_categorical(y_test, numero_clases)
19

```

Fig. 35: Distribución de imágenes previas a la implementación de la red propuesta.

La implementación del modelo propio se planteó de la siguiente manera.

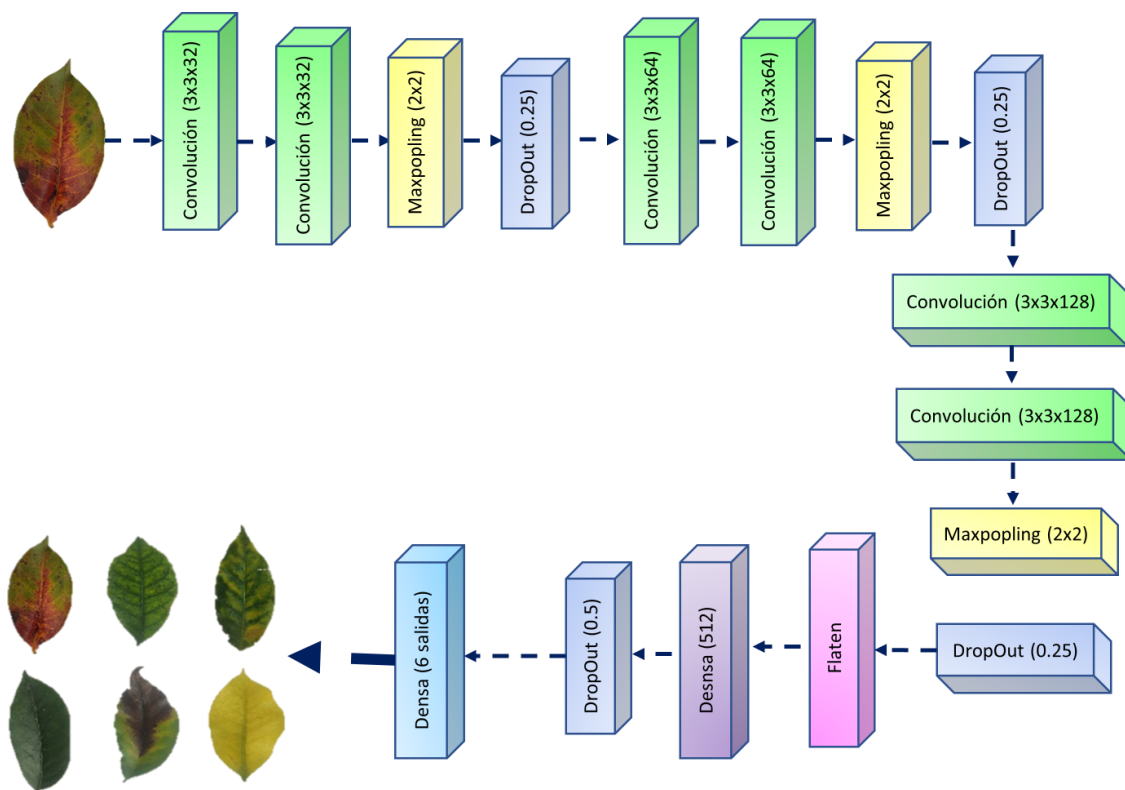


Fig. 36: Diseño de la red propuesta para el modelo propio. Fuente: Elaboración propia

Se construyó el extractor de características en tres bloques, cada uno con dos capas convolucionales. El primero con filtros de 32, el segundo con filtros de 64 y el tercero con filtros de 128. Cada bloque está intercalado con una capa MaxPooling para reducir la dimensionalidad y una capa de Dropout con una tasa de 0.25 para prevenir el sobreajuste.

```
# Definición del modelo CNN
model = Sequential()

# Primer bloque
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(240, 128, 3)))
model.add(Conv2D(32, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Dropout(0.25))

# Segundo bloque
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Dropout(0.25))

# Tercer bloque
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Dropout(0.25))
```

Fig. 37: Arquitectura CNN -Extracción de características

Una vez ajustado el extractor de características, se definió un clasificador que tomara esas características y generara la clase de la imagen de entrada. Se utilizó una capa Flatten para convertir las características 2D en un vector 1D, seguida de una capa densa con 512 unidades y activación ReLU. Se implementó un DropOut del 50% para prevenir el sobreajuste, y se agregó una capa densa final con activación softmax para la clasificación de las clases.

```
# Capa de aplanamiento y capas densas
model.add(Flatten())
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(numero_clases, activation='softmax'))
```

Fig. 38: Implementación del clasificador propuesto

El modelo se compiló con el optimizador Adam, utilizando una tasa de aprendizaje de 0.0001. Se empleó la función de pérdida de entropía cruzada categórica y se configuró la métrica de exactitud para evaluar el rendimiento del modelo durante el entrenamiento.

```
opt = Adam(learning_rate = 0.0001)
model.compile(loss="categorical_crossentropy",
              optimizer = opt,
              metrics = ['accuracy'])
```

Fig. 39: *Compilación del modelo*

Dado que el conjunto de datos consta de 3366 imágenes, utilizando el 80 % para entrenamiento y el 20 % para validación, las pruebas piloto realizadas en este estudio mostraron problemas de sobreajuste en el modelo. Para abordar este problema, se empleó la técnica de aumento de datos durante la etapa de entrenamiento, utilizando ImageDataGenerator de Keras. Esto implicó aplicar transformaciones como rotación, desplazamiento, zoom, volteo horizontal, entre otras, para agregar variaciones adicionales a las imágenes existentes, lo que resultó en un total de 5405 imágenes.

```
# Aumento de datos avanzado
datagen = ImageDataGenerator(
    rotation_range=30,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    vertical_flip=True)
```

Fig. 40: *Técnica de aumento de datos aplicado a los datos de entrenamiento.*

Para el entrenamiento, se utilizó el método `fit()`, que recibió como entrada los datos aumentados y originales, estableciendo un tamaño de lote de 24 y 110 épocas. Después de cada época de entrenamiento, se evaluó el rendimiento del modelo utilizando los datos de validación. Los resultados del entrenamiento se almacenaron en el objeto `history`, que incluye métricas como la pérdida y la exactitud en cada época.

```
# Ajuste del tamaño del lote
batch_size = 24

# Definir el número de épocas de entrenamiento
epochs = 110 # Número de épocas ajustado según tus necesidades

# Entrenar el modelo
history = model.fit(augmented_data_generator,
                    epochs=epochs, validation_data=(x_test, y_test))
```

Fig. 41: Ejecución del entrenamiento del modelo

Luego se realizó una comparación del modelo CNN propio con otras redes existentes: MobilNet, VGG16 e InceptionV3. Para los tres modelos, se utilizó el aprendizaje transferencia, con los pesos del conjunto de datos ImageNet. Se emplearon las mismas técnicas de procesamiento de imágenes y el mismo conjunto de datos que el modelo propio. Se estableció la configuración de la red de la siguiente manera:

VGG16

- Las imágenes se ingresaron con un tamaño de 240x128 píxeles y 3 canales de color (RGB).
- Se excluyeron las capas densas finales.
- Se tomó la salida del último bloque.

- Se empleó una capa "flatten" para convertir los datos en un vector de una dimensión.
- Se agregaron dos capas densas personalizadas con 64 y 128 filtros, respectivamente
- Se congeló las capas preentrenadas.
- En cuanto al optimizador se utilizó Adam con una tasa de aprendizaje de 0.0001
- De igual manera como en el modelo propio se utilizó aumento de datos.
- Se entreno el modelo, considerando un tamaño de lote de 16 y 110 épocas.

```

image_input = Input(shape=(240,128, 3))
cnn_VGG16 = VGG16(input_tensor=image_input, include_top=False,weights='imagenet')

#model2.summary()

last_layer = cnn_VGG16.get_layer('block5_pool').output
x= Flatten(name='flatten')(last_layer)
x = Dense(64, activation='relu', name='fc1')(x)
x = Dense(128, activation='relu', name='fc2')(x)
out = Dense(numero_clases, activation='softmax', name='output')(x)
vgg16_model = Model(image_input, out)

#custom_model.summary()

# freeze all the layers except the dense layers
for layer in vgg16_model.layers[:-3]:
    layer.trainable = False

vgg16_model.summary()

```

Fig. 42: Implementación de la red pre entrenada VGG16

InceptionV3 y MobilNet

Se estableció las mismas configuraciones del modelo VGG16, solo a que diferencia de las capas personalizadas se consideraron filtros de 128 y 256.

```

image_input = Input(shape=(240,128, 3))
cnn_inceptionv3 = InceptionV3(input_tensor=image_input, include_top=False,weights='imagenet')

last_layer = cnn_inceptionv3.layers[-1].output
x= Flatten(name='flatten')(last_layer)
x = Dense(128, activation='relu', name='fc1')(x)
#x=Dropout(0.3)(x)
x = Dense(256, activation='relu', name='fc2')(x)
#x=Dropout(0.3)(x)
out = Dense(numero_clases, activation='softmax', name='output')(x)
inceptionv3_model = Model(image_input, out)

# freeze all the layers except the dense layers
for layer in inceptionv3_model.layers[:-6]:
    layer.trainable = False

inceptionv3_model.summary()

```

Fig. 43: Implementación de la red pre entrenada InceptionV3

```

image_input = Input(shape=(240,128, 3))
cnn_mobilenet = MobileNet(input_tensor=image_input, include_top=False,weights='imagenet')

last_layer = cnn_mobilenet.layers[-1].output
x= Flatten(name='flatten')(last_layer)
x = Dense(128, activation='relu', name='fc1')(x)
#x=Dropout(0.3)(x)
x = Dense(256, activation='relu', name='fc2')(x)
#x=Dropout(0.3)(x)
out = Dense(numero_clases, activation='softmax', name='output')(x)
mobilenet_model = Model(image_input, out)

# freeze all the layers except the dense layers
for layer in mobilenet_model.layers[:-6]:
    layer.trainable = False

```

Fig. 44: Implementación de la red pre entrenada MobileNet

Finalmente, para comprobar que tan bueno es el modelo CNN propuesto clasificando deficiencias nutricionales de cafeto con imágenes que el modelo no haya visto, se implementó una sección de pruebas, utilizando un subconjunto llamado test, el cual consta de 666 imágenes no etiquetadas contenidas entre las 5 clases de deficiencias y una clase hojas buenas. Para ello se utilizó, el aprendizaje exportado del modelo. Se realizó las pruebas y los resultados muestran que el modelo propio es capaz de clasificar de manera correcta las

clases deficiencias nutricionales de las hojas de cafeto. Tal como se evidencia a continuación.

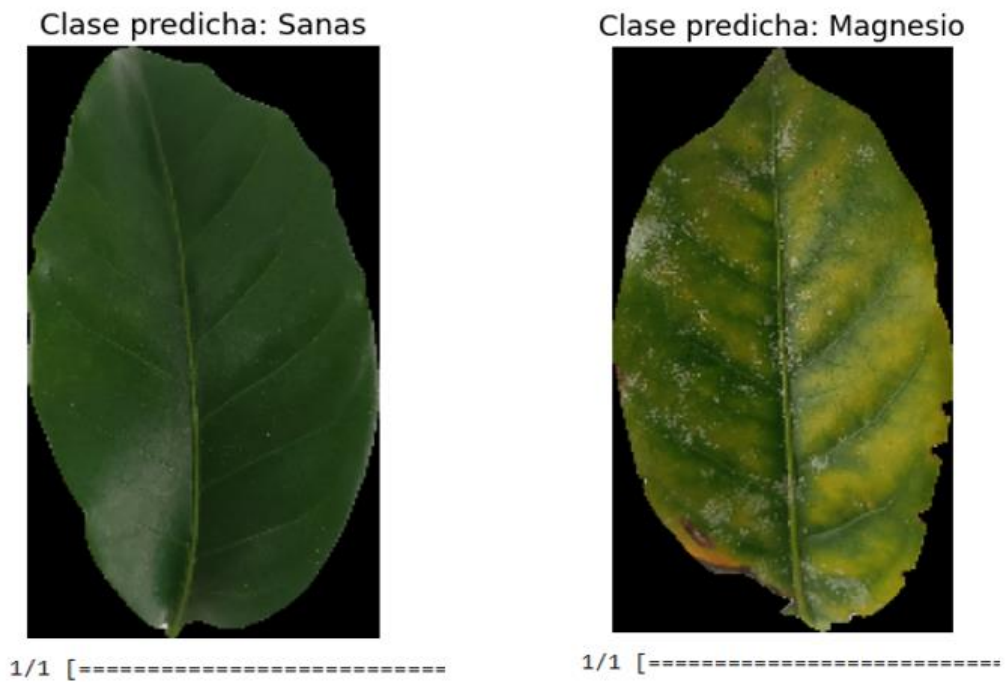


Fig. 45: Prueba realizada con el modelo propio, para clasificar una imagen de café sana y con deficiencia de magnesio.

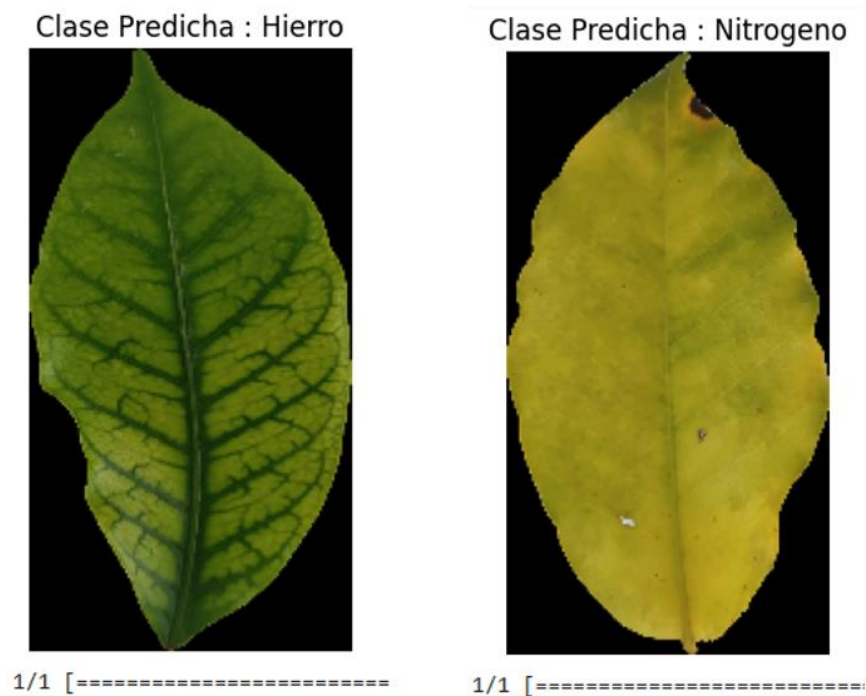


Fig. 46: Prueba realizada con el modelo propio, para clasificar una imagen con deficiencia de hierro y deficiencia de nitrógeno.

IV. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

En este estudio, se desarrolló un método para la clasificación automática de deficiencias nutricionales en hojas de café utilizando procesamiento de imágenes digitales y aprendizaje profundo. Esto incluyó la caracterización de las deficiencias, la creación de un conjunto de datos representativo y la implementación de un modelo de aprendizaje profundo.

En esta investigación, se logró caracterizar de manera efectiva las deficiencias nutricionales en las hojas del cafeto mediante un análisis visual detallado. Este proceso fue fundamental para el desarrollo de nuestro método de clasificación, ya que permitió identificar y diferenciar las manifestaciones visuales asociadas con cada tipo de carencia nutricional. Además, al profundizar en la caracterización de estas deficiencias, se ha establecido una base sólida para futuras investigaciones que estén orientadas a este campo.

Se elaboró un conjunto de datos de acuerdo con las características de las deficiencias nutricionales de las hojas del cafeto, con un total de 3366 imágenes para el entrenamiento y la validación del modelo, junto con 666 imágenes adicionales reservadas para las pruebas. Esto permite comprobar que contar con datos suficientes ayuda a que el modelo obtenga mejores resultados, debido a que, si se tienen muy pocos datos, el modelo tiende a tener un sobreajuste en el entrenamiento. De igual forma, el aumento de datos puede ser una buena técnica para mejorar el entrenamiento del modelo, sobre todo cuando se posee un conjunto de datos limitado.

Se realizó el procesamiento de las imágenes utilizando diversas técnicas y algoritmos de procesamiento, los cuales permitieron dar un adecuado tratamiento a los datos. Estas

técnicas incluyen la eliminación de ruido, la supresión del fondo y la identificación de las regiones de interés en las imágenes. Este proceso permitió mejorar la calidad de las imágenes antes de pasar a la etapa de clasificación, lo que a su vez condujo a mejores resultados en la detección de deficiencias nutricionales.

Se implementó un modelo propio de aprendizaje profundo para la clasificación de deficiencias nutricionales en hojas de cafeto, para el cual se realizaron diversas pruebas. Se experimentó con diferentes configuraciones de parámetros, ya que estas variaciones influyen en el rendimiento del modelo. Además, se compararon los resultados obtenidos con los modelos pre entrenados VGG16, Inception V3 y MobileNet, utilizando transferencia de aprendizaje. Se pudo deducir que la transferencia de aprendizaje es una opción viable, ya que los modelos pre entrenados tienen muchos de sus pesos establecidos, lo que puede reducir el tiempo de ejecución del modelo y el consumo de recursos computacionales. Además, requieren un menor conjunto de datos para su entrenamiento. Sin embargo, aunque se obtuvieron buenos resultados utilizando la transferencia de aprendizaje con modelos pre entrenados, el modelo CNN propio demostró un excelente desempeño en el consumo de recursos computacionales, mostrando resultados similares en comparación con los modelos pre entrenados. Además, superó a los demás modelos en todas las métricas evaluadas, obteniendo una exactitud de 99.11%, precisión y recall también del 99.11%, lo que indica un rendimiento destacado. Esto demuestra que el modelo CNN propio puede ser una opción viable para su aplicación en entornos de producción reales.

4.2. Recomendaciones

Si bien es cierto que el modelo propuesto, mostró muy satisfactorios, es muy importante realizar investigaciones sobre otras deficiencias nutricionales ya que en el presente documento sólo se abordaron cinco tipos, para de este modo se pueda desarrollar

herramientas computacionales capaces de realizar tareas que actualmente realizan los agricultores en el campo, ya que, actuando como observadores humanos, son muy susceptible a que se puedan cometer errores.

Dado que, en la presente investigación, el modelo propuesto se comparó solo con los modelos VGG16, Inception V3 y MobileNet, se sugiere realizar otras investigaciones sobre nuevas arquitecturas para la clasificación de estas deficiencias, esto con el fin de perfeccionar y encontrar mejores resultados.

Una mayor cantidad de datos permite utilizar modelos más robustos y sobre todos más generalizables y con mayores rendimientos, por lo que se sugiere ampliar la cantidad del conjunto de datos, incluyendo diversas variedades y condiciones de crecimiento.

REFERENCIAS

- [1] Banco Mundial, 31 Marzo 2023. [En línea]. Available: <https://www.bancomundial.org/es/topic/agriculture/overview?msckid=e1785180cf>. [Último acceso: 06 marzo 2024].
- [2] Food and Agriculture Organization of the United Nations - (FAO), [En línea]. Available: <https://www.fao.org/state-of-food-security-nutrition/2021/en/>. [Último acceso: 6 Marzo 2024].
- [3] Programa de las Naciones Unidas para el Desarrollo - (PNUD), 22 Septiembre 2020. [En línea]. Available: <https://www.undp.org/es/peru/noticias/el-cafe-peruano-en-una-sola-plataforma>. [Último acceso: 3 Marzo 2024].
- [4] Organización Internacional del Trabajo - (OIT), 9 Octubre 2020. [En línea]. Available: https://www.ilo.org/sanjose/WCMS_757602/lang--es/index.htm?msckid=a5310a4bcfde11ec823ce197c450de82. [Último acceso: 5 Marzo 2024].
- [5] Programa de las Naciones Unidas Para el Desarrollo - (PUND), 13 Mayo 2021. [En línea]. Available: <https://www.undp.org/es/peru/noticias/consolidando-la-cadena-de-valor-del-cafe-peruano>. [Último acceso: 5 Marzo 2024].
- [6] U. d. I. C. -. SSE, «Plataforma digital única del Estado Peruano,» Diciembre 2021. [En línea]. Available: <https://www.gob.pe/institucion/sse/informes-publicaciones/1090659-reporte-estadistico-de-cafe>. [Último acceso: 5 Febrero 2024].
- [7] J. Sosa, J. Ramirez, L. Vives y G. Kemper, «An Algorithm For Detection of Nutritional Deficiencies from Digital Images of Coffee Leaves Based on Descriptors and Neural Networks,» de *2019 XXII Symposium on Image, Signal Processing and Artificial Vision (STSIVA)*, Bucaramanga, Colombia, 2019.
- [8] E. Murali y S. M. Anuncia, «A Survey on Computational Aptitudes towards Precision

- Agriculture using Data Mining,» de *2022 3ª Conferencia Internacional sobre Electrónica y Comunicación Inteligentes (ICOSEC)*, Trichy, India, 2022.
- [9] Q. Zhai, C. Ye, L. Jizhong, Z. Ruzhi y J. Hua, «Rice nitrogen nutrition monitoring classification method based on the convolution neural network model: Direct detection of rice nitrogen nutritional status,» *Plos One*, 2022.
- [10] M. Sobhana, R. Vallabhaneni, T. Vasireddy y D. Polavarpu, «Deep Ensemble Mobile Application for Recommendation of Fertilizer Based on Nutrient Deficiency in Rice Plants Using Transfer Learning Models,» *International Journal of Interactive Mobile Technologies*, vol. 16, pp. 100-112, 2022.
- [11] V. Gokula, J. Deepa, P. Venkateswara, V. Divya y S. Kaviarasan, «An automated segmentation and classification model for banana leaf disease detection,» *Journal of Applied Biology & Biotechnology*, 2022.
- [12] M. Yebasse, B. Shimelis, H. Warku, J. Ko y K. Cheoi, «Coffee Disease Visualization and Classification,» *Plants*, 2021.
- [13] T. Soo y R. Ngadiran, «Plant Diseases Classification using Machine Learning,» *Journal of Physics: Conference Series*, 2021.
- [14] A. M. Khattak, M. U. Asghar, U. Batool, M. Z. Asghar, H. Ullah, M. S. Al-Rakhami y A. H. Gumaei, «Automatic Detection of Citrus Fruit and Leaves Diseases Using Deep Neural Network Model,» *IEEE Access*, 2021.
- [15] K. Lewis y J. Espineli, «Classification And Detection Of Nutritional Deficiencies In Coffee Plants Using Image Processing And Convolutional Neural Network (CNN),» *International Journal of Scientific and Technology Research*, pp. 2076-2081, 2020.
- [16] W. A. Ezat, M. M. Dessouky y N. A. Ismail, «Multi-class Image Classification Using Deep Learning Algorithm,» *Journal of Physics: Conference Series*, 2020.
- [17] J. Bose, S. Priya, Welekar y Z. Gao, «Hemp Disease Detection and Classification Using Machine Learning and Deep Learning,» de *2020 IEEE Intl Conf on Parallel &*

Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking, Exeter, Reino Unido, 2020.

- [18] H. Mejía, R. Antón, J. Nicolás, V. Tuesta y M. Forero, «New method for subject identification based on palm print,» *The International Society for Optical Engineering*, 2020.
- [19] T. Tran, J. Choi, T. Le y J. Kim, «A Comparative Study of Deep CNN in Forecasting and Classifying the Macronutrient Deficiencies on Development of Tomato Plant,» *Appl. Sci.*, 2019.
- [20] M. V.-G. L. T.-M. V. M.-C. H. T. R. Vassallo-Barco, «Automatic detection of nutritional deficiencies in coffee tree leaves through shape and texture descriptors,» *Journal of Digital Information Management*, vol. 15, nº 1, pp. 7-18, 2017.
- [21] N. M. Bacigalupo, E. L. Cabral y R. M. Salas, *Flora de San Juan*, Fundación Argentina, 2017.
- [22] INTAGRI, «Diagnóstico Visual de Deficiencias Nutrimientales,» *artículos Técnicos de INTAGRI*, 2017.
- [23] S. Sadeghian, «Avances Técnicos Cenicafé,» 2020. [En línea]. Available: https://publicaciones.cenicafe.org/index.php/avances_tecnicos/article/view/125. [Último acceso: 7 Marzo 2024].
- [24] R. López y P. Meseguer, *Inteligencia artificial*, Madrid: Consejo Superior de Investigaciones Científicas; Los Libros de la Catarata, 2017, p. 157.
- [25] J. Palma y R. Marín, *Inteligencia Artificial. Técnicas, Métodos y Aplicaciones*, Madrid: MCGRAW-HILL (PAPEL), 2008.
- [26] A. Bosch, J. Casas y T. Lozano, *Deep learning : principios y fundamentos*, Barcelona: UOC, 2020.
- [27] Microsof. [En línea]. Available: <https://azure.microsoft.com/es-es/resources/cloud->

computing-dictionary/what-is-machine-learning-platform/. [Último acceso: 4 Marzo 2024].

- [28] A. Zapeta, G. A. Galindo, H. Juan Santiago y M. Martinez, «Métricas de rendimiento para evaluar el aprendizaje automático en la clasificación de imágenes petroleras utilizando redes neuronales convolucionales,» *Ciencia Latina Revista Científica Multidisciplinar*, 2022.
- [29] J. Giacomantone, 2018. [En línea]. Available: <https://www.info.unlp.edu.ar/wp-content/uploads/2018/03/Fundamentos-y-Aplicaciones-de-Aprendizaje-Automatico.pdf>. [Último acceso: 5 Marzo 2024].
- [30] J. Béjar Alonso, «Repositorio universidad politecnica de Catalunya,» 223. [En línea]. Available: <https://www.upc.edu/content/grau/guiadocent/pdf/esp/270029>. [Último acceso: 2 Febrero 2024].
- [31] E. Soria, P. Rodriguez y Q. García, *Inteligencia Artificial: casos prácticos con aprendizaje profundo*, Madrid: RA-MA, 2022.
- [32] J. VenderPlass, *Python Data Science Handbook*, 2017.
- [33] C. Pineda Pertuz, *Aprendizaje automático y profundo en Python: una mirada hacia la inteligencia artificial*, RA-MA, 2022.
- [34] J. Bobadilla, *Machine Learning y Deep Learning: Usando Python, Scikit y Keras*, RA-MA, 2020.
- [35] D. Calvo, 20 Julio 2017. [En línea]. Available: <https://www.diegocalvo.es/red-neuronal-convolucional/>. [Último acceso: 24 Enero 2024].
- [36] Z. Yufen, Y. Cifford y M. Aleksey, «Breast cancer screening using convolutional neural network and follow-up digital mammography,» 2018.
- [37] B. Qochuk, «OpenGenus,» [En línea]. Available: <https://iq.opengenus.org/inception-v3-model-architecture/>.
- Z. Zuopeng, Z. Zhonxing, X. Xinzheng, X. Yi, Y. Hualin y Z. Lan, «A Lightweight Object

- [38] Detection Network for Real-Time Detection of Driver Handheld Call on Embedded Devices,» *Computational Intelligence and Neuroscience*, pp. 1-12, 15 Diciembre 2020.
- [39] M. G. Forero Vargas, *Introducción al procesamiento de imágenes*, Bogotá, Colombia: La Silueta Ediciones Ltda, 2002.
- [40] R. Gonzales, R. Woods y B. Masters, *Digital Image Processing, Third Edition*, 2009.
- [41] R. Gonzales y R. Woods, *Procesamiento de imágenes digitales*, 2006.
- [42] R. C. Gonzales y R. E. Woods, *Digital Image Processing*, 2008.
- [43] M. Sonka, V. Hlavac y R. Boyle, *Image processing, analysis, and machine vision (4th ed.)*. Cengage Learning, 2014.
- [44] W. Burger y M. J. Burge, *Principles of digital image processing: Core algorithms*, 2009.
- [45] J. R. Parker, *Algorithms for image processing and computer vision*, 2010.
- [46] MathWorks, «graythresh. Centro de ayuda.» [En línea]. Available: <https://la.mathworks.com/help/images/ref/graythresh.html>. [Último acceso: 10 Marzo 2024].
- [47] ICHI.PRO, 2024. [En línea]. Available: <https://ichi.pro/es/segmentacion-de-roi-deteccion-de-contornos-y-umbrales-de-imagen-mediante-opencv-212012106233735>. [Último acceso: 10 Marzo 2024].
- [48] UNAM. [En línea]. Available: https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/1023/mod_resource/content/1/contenido/index.html. [Último acceso: 4 Marzo 2024].
- [49] Python, [En línea]. Available: <https://www.python.org/about/>. [Último acceso: 8 Marzo 2024].
- [50] B. V. Ridge, «mediummultimedia,» 20 Diciembre 2023. [En línea]. Available: <https://www.mediummultimedia.com/apps/como-se-usa-una-libreria-en-programacion/>. [Último acceso: 6 Marzo 2024].

- [51] TensorFlow, «Documentación,» [En línea]. Available: <https://www.tensorflow.org/guide?hl=es-419>. [Último acceso: 8 Marzo 2024].
- [52] D. Johnson, «Guru99,» 9 Diciembre 2023. [En línea]. Available: <https://www.guru99.com/tensorflow-vs-keras.html>. [Último acceso: 8 Marzo 2024].
- [53] Python, [En línea]. Available: <https://docs.python.org/3/library/os.html>. [Último acceso: 2 Marzo 2024].
- [54] Python, [En línea]. Available: <https://docs.python.org/3/library/pathlib.html>. [Último acceso: 9 Marzo 2024].
- [55] Matplotlib, «Matplotlib: Visualización con Python,» [En línea]. Available: <https://matplotlib.org/>. [Último acceso: 5 Marzo 2024].
- [56] Python, [En línea]. Available: <https://docs.python.org/es/3/library/time.html>. [Último acceso: 7 Marzo 2024].
- [57] PyPI, [En línea]. Available: <https://pypi.org/project/psutil/>. [Último acceso: 9 Marzo 2024].
- [58] E. Martín, «Cibernos Grupo,» [En línea]. Available: <https://www.grupocibernos.com/blog/como-funcionan-los-servicios-en-la-nube>. [Último acceso: 7 Marzo 2024].
- [59] Google, [En línea]. Available: <https://research.google.com/colaboratory/intl/es/faq.html>. [Último acceso: 15 Enero 2024].
- [60] O. d. I. N. U. p. I. A. y. I. (. Agricultura. [En línea]. Available: <https://www.fao.org/faostat/es/#data/QCL>. [Último acceso: 6 Marzo 2024].
- [61] M. Suárez, «Programa de las Naciones Unidas para el Desarrollo (PNUD),» 18 Octubre 2018. [En línea]. Available: <https://www.undp.org/es/peru/noticias/se-presenta-plan-nacional-de-accion-del-cafe-para-aumentar-produccion-y-conquistar-mas->

ANEXOS

Anexo 1: Resolución de aprobación del proyecto de investigación



FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO

RESOLUCIÓN N° 0758-2022/FIAU-USS

Pimentel, 05 de diciembre de 2022

VISTOS:

El Acta de reunión N° 01711 – 2022 del Comité de investigación de la Escuela profesional de INGENIERÍA DE SISTEMAS remitida mediante oficio N° 0261-2022/FIAU-II-USS de fecha 30 de noviembre de 2022, 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, según documentos de vistos el Comité de investigación de la Escuela profesional de INGENIERÍA DE SISTEMAS acuerda aprobar el tema tesis en el extremo de la tesis a cargo de los estudiantes o egresados que se detallan en el anexo de la presente Resolución.

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

SE RESUELVE:

ARTÍCULO 1°: APROBAR, el tema tesis en el extremo de la tesis perteneciente a la línea de investigación de INFRAESTRUCTURA, TECNOLOGÍA Y MEDIO AMBIENTE, a cargo de los estudiantes o egresados del Programa de estudios de INGENIERÍA DE SISTEMAS según se detalla en el anexo de la presente Resolución.

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





ANEXO

APROBACION DE TEMA DE TESIS

SECCION A

	APELLIDOS	TESIS
1	TAPIA FUENTES MARIANO HERNANDO	IMPLEMENTACIÓN DE UN MODELO DE GESTIÓN DE SERVICIOS DE TI PARA UNA PEQUEÑA EMPRESA PERUANA BASADA EN ITIL V4
2	RIVAS PLATA CASAS CARLOS GUALBERTO	CLASIFICACIÓN DE CÁNCER DE PULMÓN EN IMÁGENES DE TOMOGRAFÍAS MEDIANTE PROCESAMIENTO DE IMÁGENES Y APRENDIZAJE AUTOMÁTICO
3	FARROÑAN TERAN NIXON PAUL	APLICACIÓN DEL ESTÁNDAR ISO 27001:2017 PARA MEJORAR EL PROCESO DE GESTIÓN DE RIESGOS TI EN UN INSTITUTO TECNOLÓGICO PÚBLICO PERUANO
4	LUNA BECERRA JHERSON ISAC	MÉTODO DE CLASIFICACIÓN AUTOMÁTICA DE DEFICIENCIAS NUTRICIONALES EN HOJAS DE CAFETO MEDIANTE PROCESAMIENTO DE IMÁGENES DIGITALES Y APRENDIZAJE PROFUNDO
5	CABREJOS SEVERINO LUZ ANTONELLA	DESARROLLO DE UN MÉTODO DE LECTURA AUDIBLE DE LIBROS DE TEXTO BASADO EN PROCESAMIENTO DIGITAL DE IMÁGENES Y DEEP LEARNING
6	RABANAL SENMACHE MARRY CECY SÁNCHEZ RUBIO OMAR ALBERTO	DISEÑO DE UN MODELO DE AUDITORÍA DE TI PARA EL CUMPLIMIENTO DE NORMAS Y POLÍTICAS DE UNA EMPRESA RETAIL PERUANA
7	JULCA BRUNO HOLVER KABIR VILCHEZ BAILA ERICKSON YOVANI	EVALUACIÓN DE LA CALIDAD EN USO DE UN SISTEMA DE GESTIÓN EDUCATIVA PARTICULAR BASADO EN LA NORMA ISO/IEC 25022
8	FIESTAS TELLO, TATIANA MERCEDES TEJADA PAREDES, JORGE LUIS	IMPLEMENTACIÓN DE UN MODELO BASADO EN ITIL 4, PARA MEJORAR LA EFICIENCIA DE ITSM EN UNA INSTITUCIÓN REGIONAL DEL ESTADO.
9	AYALA SÁNCHEZ MARIO FRANKLIN TEPE LEON EDWIN ANTONIO	MÉTODO DE DETECCIÓN DE ATAQUES A UNA RED DEFINIDA POR SOFTWARE DE UNA EMPRESA PERUANA BASADO EN MACHINE LEARNING

SECCION B

	APELLIDOS	TESIS
1	SANCHEZ PARDO SAMUEL MORE VILLEGAS FIORELLA JHAJIRA	DESARROLLO DE MÉTODO PARA LA CLASIFICACIÓN POR MADUREZ DE LA FRESA UTILIZANDO PROCESAMIENTO DE IMÁGENES DIGITALES Y MACHINE LEARNING
2	LOPEZ VALLEJOS ROBER YUBELDER	IMPLEMENTACIÓN DE TECNOLOGÍA BLOCKCHAIN ASOCIANDO USUARIO Y DISPOSITIVO PARA MEJORAR LA SEGURIDAD EN LA AUTENTICACIÓN DE CREDENCIALES DE ACCESO
3	LOZANO DELGADO KELKIN HEIMINN GUERRERO VEGA ERICKS TITO	IMPLEMENTACIÓN DE ALGORITMOS CRIPTOGRÁFICOS PARA MEJORAR LA SEGURIDAD EN EL ENVÍO DE TEXTO PLANO POR INTERNET
4	GARCIA CHOZO DIANA KATHERINE MAQUEN MUJICA MIGUEL ANGEL	DESARROLLO DE APLICATIVO MOVIL DE REALIDAD AUMENTADA PARA MEJORAR EL APRENDIZAJE DE INGLÉS EN ESTUDIANTES DE PRIMERO DE SECUNDARIA





	APELLIDOS	TESIS
5	MONJA VASQUEZ FERNANDO JOEL	IMPLEMENTACIÓN DE UNA APLICACIÓN WEB PARA COMPARTIR Y EJECUTAR COMPLEMENTOS DE IMAGEJ
6	BENAVIDES CIURLIZZA OSCAR JAVIER DIAZ CORONADO JENNIFFER GERALDINE	IMPLEMENTACIÓN DE UN MODELO BASADO EN LA TECNOLOGÍA BLOCKCHAIN PARA LA GESTIÓN DE HISTORIAS CLÍNICAS ELECTRÓNICAS
7	HERNA LOJA BRYAM ALEXANDER	DESARROLLO DE UN ENFOQUE ÁGIL PARA OPTIMIZAR LA CONSTRUCCIÓN DE SOFTWARE EN UN EMPRESA LAMBAYECANA
8	PACHECO CONTRERAS NICOLETTE ISIS SANTISTEBAN OSTOS ANDY JOSUE	IMPLEMENTACIÓN DE UN MODELO DE APRENDIZAJE PROFUNDO PARA LA TRADUCCIÓN DEL LENGUAJE DE SEÑAS
9	ASENJO SAAVEDRA SEGUNDO VICTOR PAICO SANTOS PERLA PAOLA	EVALUACIÓN DE SOFTWARE DE UN LABORATORIO CLÍNICO BAJO LA NORMA ISO/IEC 25000 PARA ASEGURAR LA CALIDAD ÓPTIMA DEL SISTEMA
10	TARRILLO CHIRINOS ANGELA CAROLINA GUARNIZ PAREDES CARLOS ALONSO	DESARROLLO DE UN MÉTODO PARA DETECTAR PERSONAS ARMADAS UTILIZANDO MACHINE LEARNING EN VIDEOVIGILANCIA
11	PISFIL SUGARAY ALBERTO ESTUARDO	IMPLEMENTACION DE MODELO DE SERVICIOS DE TI MEDIANTE ISO/IEC 20000 EN EMPRESAS DE DESARROLLO DE SOFTWARE
12	LEYVA CRUZ SILVANA YAOSKELLINEY VASQUEZ TORRES YAMIR	ALGORITMOS DE DATA MINING PARA PREDECIR LA DESERCIÓN Y DESAPROBACIÓN ESTUDIANTIL EN LA ENSEÑANZA Y APRENDIZAJE EN ENTORNOS VIRTUALES
13	ALBARRAN SALAZAR HARVY JEFFERSON DELGADO FARRO JAVIER MANUEL	EVALUACION DE ENFOQUES PARA LA OPTIMIZACIÓN DE PROYECTOS DE DESARROLLO DE SOFTWARE: CASO DE ESTUDIO EMPRESA SOFTCLOUD
14	DAMIAN DIAZ CESAR ALBERTO	IMPLEMENTACIÓN DE UNA APLICACIÓN WEB HÍBRIDA PARA LA INTEGRACIÓN DE DATOS EN LA MICRORED DE SALUD OLMOS, LAMBAYEQUE
15	DAVILA GONZALES GIOVANNY ALEXIS AYMA VELASQUEZ SEBASTIAN ALEJANDRO	MODELO DE GESTIÓN DE RIESGOS DE SEGURIDAD DE INFORMACIÓN PARA MITIGAR EL IMPACTO NEGATIVO EN UNA PYME PERUANA
16	CHUMAN LLUEN DAGNER ANIBAL	DESARROLLO DE UNA APLICACIÓN MÓVIL BASADA EN MICROSERVICIOS PARA AGUILIZAR EL PROCESO DE VENTAS DE UN MINIMARKET
17	CALDERON PIZANGO FLAVIO CESAR CARRANZA MORALES JUAN TOMAS GERALD	IMPLEMENTACIÓN DE TECNOLOGIA BLOCKCHAIN PARA GESTIONAR LAS TRANSACCIONES DE PAGOS Y VENTAS EN UNA EMPRESA LAMBAYECANA

INVESTIGACION II

	APELLIDOS	TESIS
1	CIRILO ESCUDERO FRANCK	COMPARACIÓN DE HERRAMIENTAS DE VIRTUALIZACIÓN DE SOFTWARE LIBRE PARA MEJORAR LA DISPONIBILIDAD DE LOS SERVICIOS DE RED EN UNA EMPRESA MICROFINANCIERA
2	MENDOZA HUANGAL WALTER MARTINEZ CASUSOL ENRIQUE VALENTIN	SISTEMA WEB PARA MEJORAR LA GESTIÓN DE ESTUDIO JURÍDICO UTILIZANDO WEB SCRAPING Y ARQUITECTURA DE MICRO SERVICIOS EN CLOUD COMPUTING
3	CASTRO FERNANDEZ IRVIN GREGORY	MODELO DE GESTIÓN DE RIESGOS PARA MEJORAR LA CONFIDENCIALIDAD DE PROYECTOS DE UNA EMPRESA DE AUDITORÍA DE SISTEMAS DE INFORMACIÓN
	CAYACA CAJUSOL MARIA JUANA	DESARROLLO DE UN MÉTODO DE ADQUISICIÓN BASADO EN NORMAS INTERNACIONALES PARA LA MEJORA DEL PROCESO ADQUISITIVO DE SOFTWARE EN MYPES PERUANAS

APROBACION DE TEMA DE TESIS

	APELLIDOS	TESIS
1	BARRERA TORRES JORGE LUIS	GESTIÓN DEL SERVICIO BASADO EN ITIL V4 PARA EVALUAR LA CALIDAD DEL SOFTWARE DE RECAUDACIÓN DE AGUA POTABLE Y ALCANTARILLADO DE LA EPS JUCUSBAMBA EIRL

DR. VICTOR ALEXCI TUESTA MONTEZA
DECANO (E) FACULTAD DE INGENIERÍA,
ARQUITECTURA Y URBANISMO
UNIVERSIDAD SEÑOR DE SIPÁN SAC.
CHICLAYO




DR. HALYN ALVAREZ VÁSQUEZ
SECRETARIO ACADÉMICO | FACULTAD
DE INGENIERÍA, ARQUITECTURA Y URBANISMO
UNIVERSIDAD SEÑOR DE SIPÁN SAC.
CHICLAYO

REGÍSTRESE, COMUNÍQUESE Y ARCHÍVESE

Cc: Interesado, Archivo

Anexo 2: Imágenes evidencia de la base datos

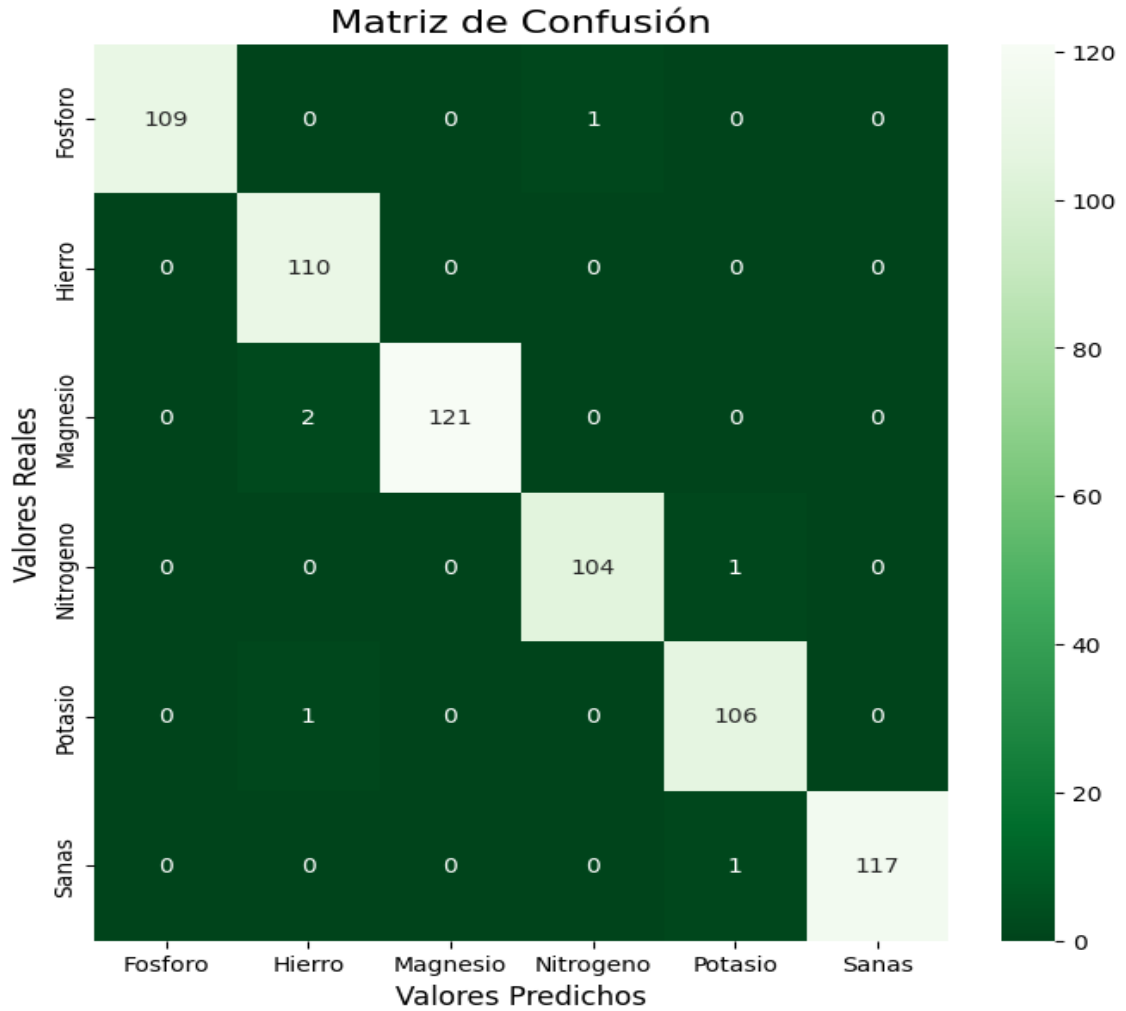






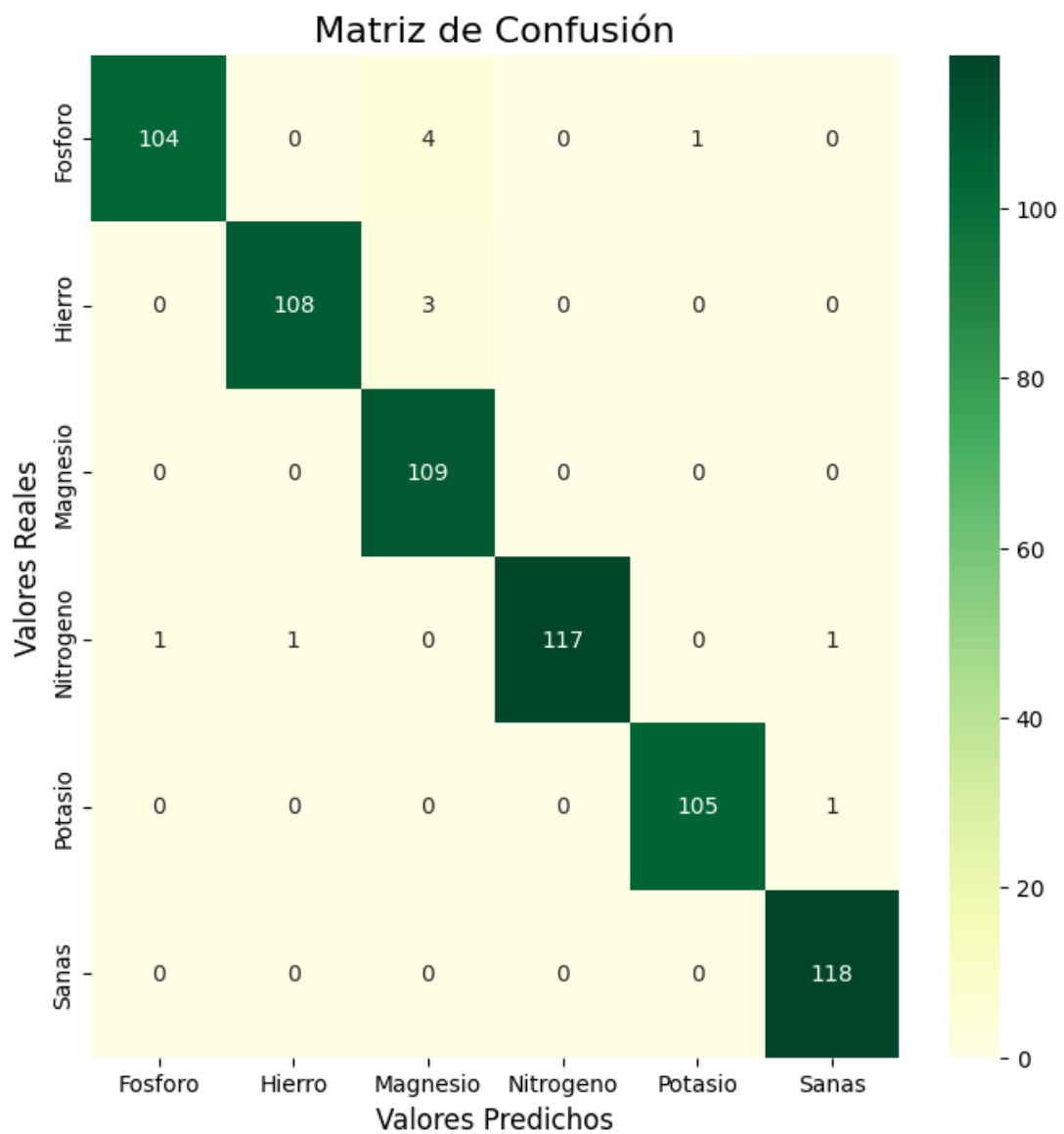
Anexo 3: Matriz de confusión y reporte de métricas de los modelos abordados

Modelo CNN propio



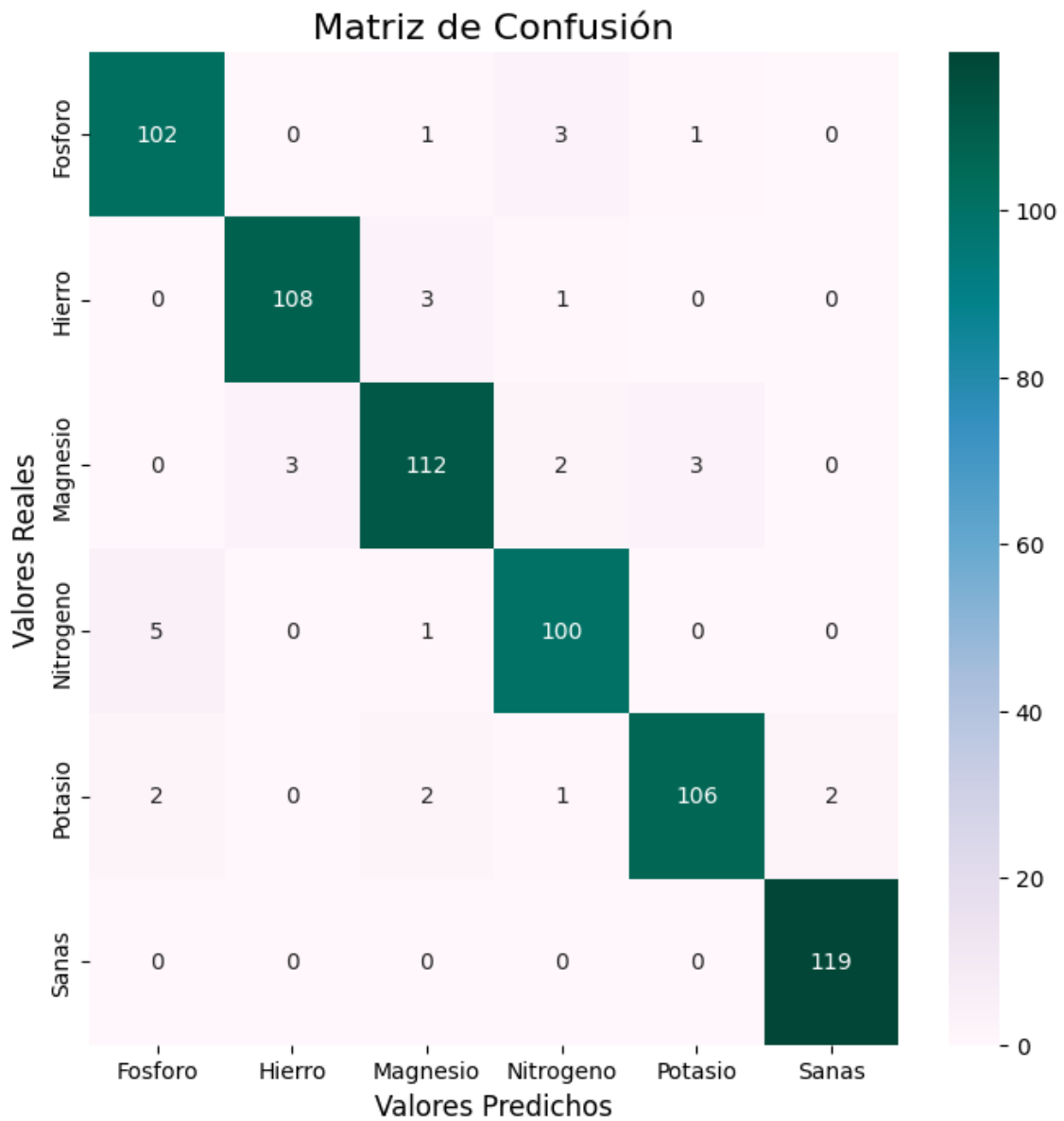
	precision	recall	f1-score
0	1.0000	0.9909	0.9954
1	0.9735	1.0000	0.9865
2	1.0000	0.9837	0.9918
3	0.9905	0.9905	0.9905
4	0.9815	0.9907	0.9860
5	1.0000	0.9915	0.9957
accuracy			0.9911
macro avg	0.9909	0.9912	0.9910
weighted avg	0.9912	0.9911	0.9911

Modelo MobileNet



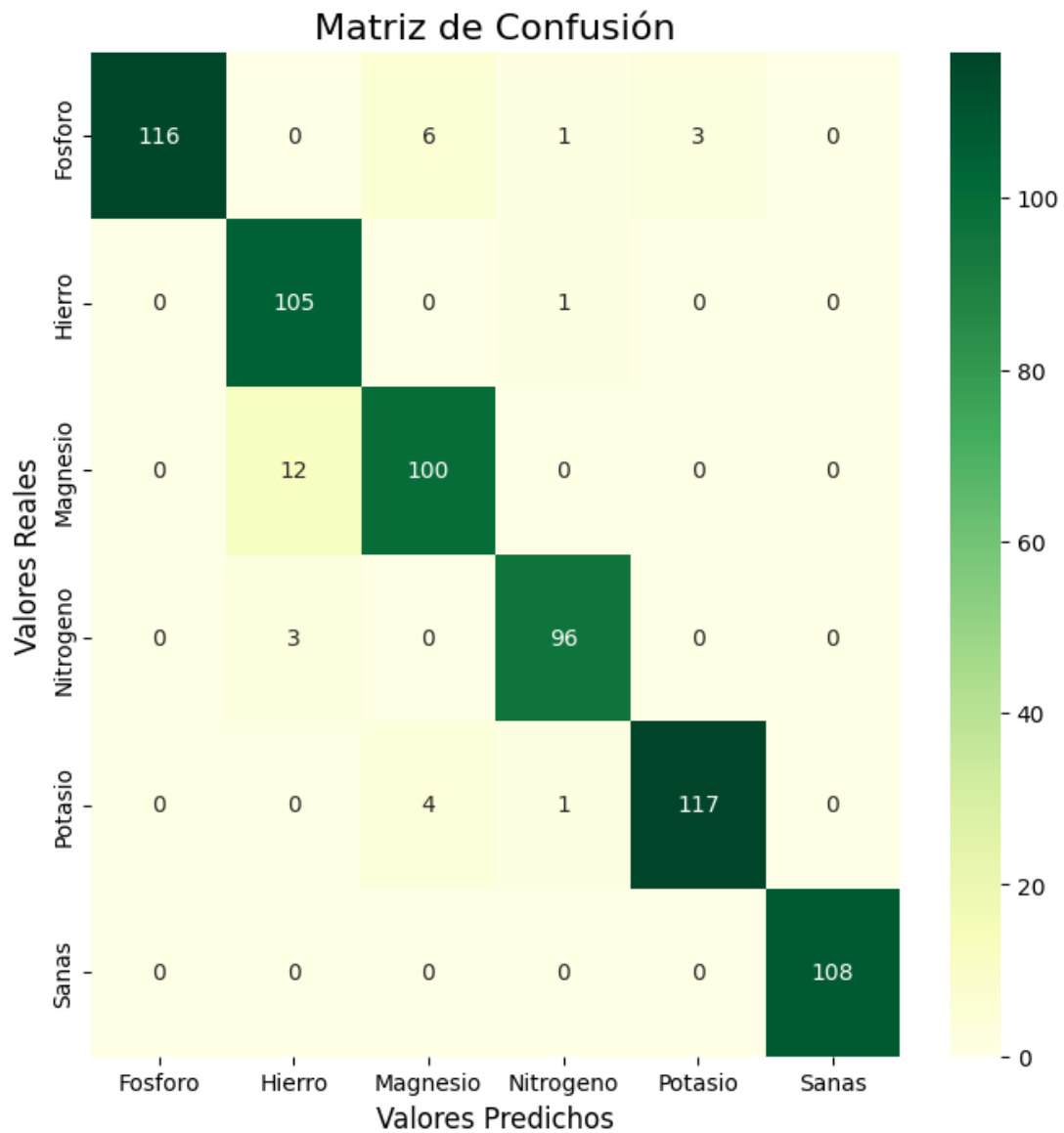
	precision	recall	f1-score
0	0.9905	0.9541	0.9720
1	0.9908	0.9730	0.9818
2	0.9397	1.0000	0.9689
3	1.0000	0.9750	0.9873
4	0.9906	0.9906	0.9906
5	0.9833	1.0000	0.9916
accuracy			0.9822
macro avg	0.9825	0.9821	0.9820
weighted avg	0.9828	0.9822	0.9822

Modelo InceptionV3



	precision	recall	f1-score
0	0.9358	0.9533	0.9444
1	0.9730	0.9643	0.9686
2	0.9412	0.9333	0.9372
3	0.9346	0.9434	0.9390
4	0.9636	0.9381	0.9507
5	0.9835	1.0000	0.9917
accuracy			0.9557
macro avg	0.9553	0.9554	0.9553
weighted avg	0.9557	0.9557	0.9556

Modelo VGG16



	precision	recall	f1-score
0	1.0000	0.9206	0.9587
1	0.8750	0.9906	0.9292
2	0.9091	0.8929	0.9009
3	0.9697	0.9697	0.9697
4	0.9750	0.9590	0.9669
5	1.0000	1.0000	1.0000
accuracy			0.9539
macro avg	0.9548	0.9555	0.9542
weighted avg	0.9562	0.9539	0.9542