

UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN

Escuela de Posgrado

**MAESTRÍA EN INGENIERÍA DE SISTEMAS E INFORMÁTICA -
ADMINISTRACIÓN DE TECNOLOGÍAS DE INFORMACIÓN**

**CLASIFICACIÓN DE ENFERMEDADES DE OLIVO
(*Olea europea*) Y ORÉGANO (*Origanum vulgare*)
MEDIANTE MODELOS DE REDES NEURONALES
CONVOLUCIONALES EN TACNA - PERÚ**

TESIS

PRESENTADA POR:

ISRAEL NAZARETH CHAPARRO CRUZ

Para optar el Grado Académico de:

**MAESTRO EN CIENCIAS (*MAGISTER SCIENTIAE*) CON MENCIÓN EN
INGENIERÍA DE SISTEMAS E INFORMÁTICA - ADMINISTRACIÓN DE
TECNOLOGÍAS DE INFORMACIÓN**

TACNA - PERÚ

2024

UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN

Escuela de Posgrado

**MAESTRÍA EN INGENIERÍA DE SISTEMAS E INFORMÁTICA -
ADMINISTRACIÓN DE TECNOLOGÍAS DE INFORMACIÓN**

**CLASIFICACIÓN DE ENFERMEDADES DE OLIVO (*Olea europea*) Y ORÉGANO
(*Origanum vulgare*) MEDIANTE MODELOS DE REDES NEURONALES
CONVOLUCIONALES EN TACNA - PERÚ**

Tesis sustentada y aprobada el 24 de enero de 2024; estando el jurado calificador integrado por:

PRESIDENTE : 
Dr. Nataniel Mario Linares Gutiérrez

SECRETARIO : 
Dra. Ana Silvia Cori Morón

MIEMBRO : 
Dr. Erbert Francisco Osco Mamani

ASESOR : 
Dr. Erbert Francisco Osco Mamani

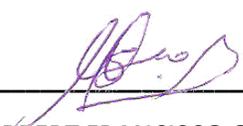
CERTIFICADO DE SIMILITUD

Yo, **Dr. Erbert Francisco Osco Mamani** en mi condición de Asesor de la tesis titulada: **“Clasificación de Enfermedades de Olivo (*Olea europea*) y Orégano (*Origanum vulgare*) mediante Modelos de Redes Neuronales Convolucionales en Tacna - Perú”**, desarrollado por el egresado de la Maestría en Ingeniería de Sistemas e Informática, **ISRAEL NAZARETH CHAPARRO CRUZ**, para optar el Grado Académico de Maestro en Ciencias con mención en Ingeniería de Sistemas e Informática, conforme a la **Directiva de Autenticidad de Documentos Académicos, de Investigación y Tesis de Maestría y Doctorado** disponible en la página web de la ESPG-UNJBG -> Normas, aplicando el software de similitud textual TURNITIN y las **exclusiones del numeral 5.5 de dicha directiva**, se obtiene los siguientes resultados:

Código de identificación de reporte : oid:23228:293063249
Porcentaje de similitud : 2%
Nivel : PERMITIDO - Trabajo Auténtico
(Según Art. 5.6 de la Directiva)

Por lo que **CERTIFICO** que la **SIMILITUD** del Informe Final de Tesis está de acuerdo al nivel **PERMITIDO (Trabajo Auténtico)**. El tesista puede continuar con el proceso de grado respectivo. Se emite el presente certificado para los finales correspondientes.

Tacna, 06 de diciembre de 2023



DR. ERBERT FRANCISCO OSCO MAMANI

ASESOR DE TESIS

Dedicatoria

À la Gloire du Grand Architecte de l'Univers.

AGRADECIMIENTOS

Me gustaría expresar mi reconocimiento y agradecimiento al NVIDIA Deep Learning Institute (DLI) por proporcionar el Taller Gratuito de Desarrollo Docente: "Fundamentos de Deep Learning" el 18 de febrero de 2022 y luego permitirme brindar, como Instructor Certificado y Embajador Universitario, talleres y certificaciones gratuitas de "Fundamentos de Deep Learning" a mis estudiantes de la **ESIS-UNJBG** durante los semestres académicos 2022-I y 2023-I.

En esta investigación se hace uso de conjuntos de datos del proyecto "Visión computacional para la detección y diagnóstico de enfermedades en cultivos relevantes de la Región Tacna" financiado con fondos de Canon, SobreCanon y Regalías Mineras de la Universidad Nacional Jorge Basadre Grohmann, Tacna - Perú.

Índice general

RESUMEN	XII
ABSTRACT	XIII
INTRODUCCIÓN	XIV
I PLANTEAMIENTO DEL PROBLEMA DE INVESTIGACIÓN	15
1.1 PLANTEAMIENTO DEL PROBLEMA	15
1.1.1 Identificación del problema	15
1.2 FORMULACIÓN DEL PROBLEMA	18
1.2.1 Problema principal	18
1.2.2 Problemas secundarios	18
1.3 JUSTIFICACIÓN E IMPORTANCIA DE LA INVESTIGACIÓN	18
1.4 OBJETIVOS	20
1.4.1 Objetivo general	20
1.4.2 Objetivos específicos	20
1.5 HIPÓTESIS	21
1.5.1 Hipótesis general	21
1.5.2 Hipótesis específicas	21

1.6	VARIABLES	22
1.6.1	Identificación de variables	22
1.6.2	Definición conceptual de las variables	22
1.6.3	Definición operacional de las variables	22
1.7	LIMITACIONES DE LA INVESTIGACIÓN	24
II	MARCO TEÓRICO	25
2.1	ANTECEDENTES DEL ESTUDIO	25
2.2	BASES TEÓRICAS	26
2.2.1	Deep Learning	26
2.2.2	Redes neuronales convolucionales	26
2.2.3	Arquitectura VGG16	28
2.2.4	Arquitectura MobileNet	31
2.2.5	MobileNetV2	35
2.2.6	NASNetLarge	36
2.2.7	Subajuste y sobreajuste	37
2.2.8	Aprendizaje por transferencia	37
2.2.9	Ajuste fino	38
2.2.10	Aumento de datos	38
2.3	DEFINICIÓN DE TÉRMINOS	39
2.4	MARCO FILOSÓFICO	40
III	METODOLOGÍA DE LA INVESTIGACIÓN	50
3.1	TIPO Y DISEÑO DE LA INVESTIGACIÓN	50

3.2	POBLACIÓN Y MUESTRA DE ESTUDIO	52
3.2.1	Población	52
3.2.2	Muestra	53
3.3	ACCIONES Y ACTIVIDADES PARA LA EJECUCIÓN DEL PROYECTO	56
3.4	MATERIALES E INSTRUMENTOS	57
3.5	TRATAMIENTO DE DATOS	58
3.5.1	Procedimiento de recolección de datos	58
3.5.2	Análisis y procesamiento de datos	61
IV	RESULTADOS DE LA INVESTIGACIÓN	63
4.1	DISEÑO DE ARQUITECTURAS DE RED NEURONAL CONVOLUCIONAL	63
4.2	CLASIFICACIÓN DE ENFERMEDADES DE ORÉGANO	65
4.2.1	Entrenamiento: Etapa de aprendizaje por transferencia	65
4.2.2	Entrenamiento: Etapa de ajuste fino	67
4.2.3	Selección de modelo	69
4.3	CLASIFICACIÓN DE ENFERMEDADES DE OLIVO	74
4.3.1	Entrenamiento: Etapa de aprendizaje por transferencia	74
4.3.2	Entrenamiento: Etapa de ajuste fino	76
4.3.3	Selección de modelo	78
4.4	COMPARACIÓN MODELOS POR CONJUNTO DE DATOS	83
4.5	CORRELACIÓN ENTRE MODELOS Y SU RENDIMIENTO	87
	DISCUSIONES	88
	CONCLUSIONES	89

	VIII
RECOMENDACIONES	91
REFERENCIAS BIBLIOGRÁFICAS	92
ANEXOS	99
Anexo 1: Código principal de experimentación	99

Índice de tablas

Tabla 1. <i>Entendimiento vs razonamiento</i>	41
Tabla 2. <i>Conjunto de datos de enfermedades de olivo</i>	55
Tabla 3. <i>Conjunto de datos de enfermedades de orégano</i>	56
Tabla 4. <i>Métricas de los modelos en ImageNet</i>	60
Tabla 5. <i>Métricas de las arquitecturas modificadas</i>	64
Tabla 6. <i>Olivo: Prueba de U de Mann-Whitney para mejor rendimiento</i>	71
Tabla 7. <i>Orégano: Métricas sobre todos los experimentos</i>	72
Tabla 8. <i>Orégano: Matriz de confusión</i>	73
Tabla 9. <i>Orégano: Métricas de clasificación por enfermedad</i>	73
Tabla 10. <i>Olivo: Prueba de U de Mann-Whitney para mejor rendimiento</i>	80
Tabla 11. <i>Olivo: Métricas sobre todos los experimentos</i>	81
Tabla 12. <i>Olivo: Matriz de confusión</i>	82
Tabla 13. <i>Olivo: Métricas de clasificación por enfermedad</i>	82
Tabla 14. <i>Prueba de U de Mann-Whitney VGG16 y NASNetLarge</i>	85
Tabla 15. <i>Prueba de U de Mann-Whitney Olivo vs Orégano con VGG16</i>	85
Tabla 16. <i>Prueba de U de Mann-Whitney con NASNetLarge Olivo vs Orégano</i> ...	86

Índice de figuras

Figura 1. <i>Estructura típica de una Red Neuronal Convolutiva</i>	27
Figura 2. <i>Arquitectura del modelo VGG-16</i>	28
Figura 3. <i>Arquitectura del modelo MobileNet</i>	31
Figura 4. <i>Convolución estándar vs convolución separable en profundidad</i>	34
Figura 5. <i>Arquitectura del modelo MobileNetV2</i>	35
Figura 6. <i>Arquitectura del modelo NASNetLarge</i>	36
Figura 7. <i>Ejemplo de image captioning</i>	46
Figura 8. <i>Diseño experimental</i>	51
Figura 9. <i>Muestras de Olivo</i>	54
Figura 10. <i>Muestras de orégano</i>	55
Figura 11. <i>Flujo de trabajo experimental</i>	62
Figura 12. <i>Orégano: Exactitud de los experimentos</i>	65
Figura 13. <i>Orégano: Pérdida de los experimentos</i>	66
Figura 14. <i>Orégano: Exactitud del mejor experimento</i>	69
Figura 15. <i>Orégano: Diagrama de caja de la exactitud en validación</i>	70
Figura 16. <i>Olivo: Exactitud de los experimentos</i>	74
Figura 17. <i>Olivo: Pérdida de los experimentos</i>	75

Figura 18. <i>Olivo: Exactitud del mejor experimento</i>	78
Figura 19. <i>Olivo: Diagrama de caja de la exactitud en validación.</i>	79
Figura 20. <i>Comparación modelos por conjunto de datos.</i>	83
Figura 21. <i>Criterios de selección vs rendimiento del modelo</i>	87

RESUMEN

Perú es el mayor productor y exportador de orégano de Sudamérica, destacando la región de Tacna como la mayor productora nacional (65%). Tacna también lidera la producción nacional de aceitunas (81,4%), siendo Perú el segundo exportador de aceitunas y el tercero de aceite de oliva en Sudamérica. A pesar de su importancia económica, el cultivo de olivo y orégano en Tacna se gestiona de manera tradicional, con un incipiente uso de datos.

El estudio aborda la necesidad de desarrollo tecnológico para reducir las enfermedades que afectan la cosecha de orégano y olivo. La clasificación exacta de enfermedades puede prevenir pérdidas de calidad y rendimiento.

Se llevó a cabo un extenso estudio experimental comparativo utilizando modelos como VGG16, MobileNet, MobileNetV2 y NASNetLarge, aplicando técnicas como Aumento de Datos, Aprendizaje por Transferencia y Ajuste Fino. Se concluyó que: (i) No existen diferencias significativas (p -valor=0,999) entre VGG16 y NASNetLarge para los conjuntos de datos de olivo y orégano, (ii) No existen diferencias significativas (p -valor=0,168) al comparar los conjuntos de datos de olivo vs orégano, (iii) Una fuerte relación significativa para orégano ($R^2 = 0,84$) y olivo ($R^2 = 0,66$) entre el número de parámetros y tamaño (en MB) con la exactitud en validación del modelo.

Finalmente, se publicaron conjuntos de datos y modelos de redes neuronales convolucionales de alta exactitud para clasificar enfermedades en orégano y olivo. Los resultados están disponibles en GitHub para garantizar la reproductibilidad.

Palabras clave: Aprendizaje profundo, orégano, olivo, enfermedades.

ABSTRACT

Peru is the largest producer and exporter of oregano in South America, with the region of Tacna being the largest national producer (65%). Tacna also leads the national production of olives (81,4%), being Peru the second largest exporter of olives and the third largest exporter of olive oil in South America. Despite its economic importance, olive and oregano cultivation in Tacna is managed traditionally, with incipient use of data.

The study addresses the need for technological development to reduce diseases affecting the oregano and olive crop. Accurate disease classification can prevent quality and yield losses, but the use of technologies and data in this area is limited.

An extensive comparative experimental study was carried out using models such as VGG16, MobileNet, MobileNetV2, and NASNetLarge, applying techniques such as Data Augmentation, Transfer Learning, and Fine Tuning. It was concluded that: *(i)* There are no significant differences (p -value=0,999) between VGG16 and NASNetLarge for olive and oregano datasets, *(ii)* There are no significant differences (p -value=0. 168) when comparing olive vs oregano datasets, *(iii)* A strong significant relationship for oregano ($R^2 = 0,84$) and Olive ($R^2 = 0,66$) between the number of parameters and size (in MB) with accuracy in model validation.

Finally, high accuracy Convolutional Neural Network data sets and models were published. The results are available on GitHub to ensure reproducibility.

Keywords: Deep learning, oregano, olive, diseases.

INTRODUCCIÓN

En el capítulo I se ha descrito la realidad problemática origen de esta tesis, así como los objetivos, hipótesis y variables. En el capítulo II se ha revisado el estado del arte a través de artículos científicos publicados en revistas de alto impacto, además se describe detalladamente la base teórica necesaria para comprender la investigación. En el capítulo III se ha presentado el marco filosófico de la investigación. En el capítulo IV se ha descrito y detallado la metodología, tipo y diseño de la investigación. En el capítulo V se presentan los resultados obtenidos a nivel descriptivo, separados por conjunto de datos (orégano y olivo) y la comparación y correlación de estos con los modelos entrenados. En el capítulo VI se realiza el contraste de los resultados con los resultados obtenidos en los antecedentes descritos en el capítulo III. Finalmente, se plantean las conclusiones y recomendaciones resultantes de esta investigación.

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA DE INVESTIGACIÓN

1.1 PLANTEAMIENTO DEL PROBLEMA

1.1.1 Identificación del problema

El orégano es una planta originaria de Europa y Asia, existen más de 50 especies pertenecientes a diferentes familias botánicas y géneros, que se denominan y comercializan en todo el mundo con el nombre común de orégano (Salamanca García and Sánchez Bermúdez, 2009).

Los principales países productores y exportadores de orégano se localizan en tres áreas geográficas: El sur de Europa (España, Francia, Albania, Grecia y Turquía), oriente próximo (Israel y Egipto) y América (principalmente México en América del Norte y Perú en América del Sur) (Rengifo, 2007).

Los principales productores de orégano en Sudamérica son Perú, Chile, Argentina y Bolivia, siendo Perú el mayor productor y exportador (Chagua and Amador, 2016). En 2021, Perú exportó 7 355 toneladas de orégano por un valor de US\$ 20,1 millones, lo que representa un incremento del 30% respecto a 2020 (US\$ 15,4 millones), pero respecto a 2019 (US\$ 13,4 millones) este incremento fue del 50% (Lopez, 2022).

La región Tacna es por lejos la mayor productora de orégano (*Origanum vulgare* L.) en el Perú con 2 499 ha y una producción de 11 946 t, representando el 65% de la producción nacional (Condori Tintaya and Fernández Cutire, 2022). El 70% de las

áreas de producción se ubican en los valles interandinos en altitudes entre 2000 y 3500 m.s.n.m., donde el cultivo del orégano adquiere una particular connotación económica en la agricultura familiar y una importante participación e incidencia en la agroindustria y agroexportación (Contreras et al., 2021).

Tacna es el primer productor nacional de aceitunas (MINCETUR, 2022). Perú es el segundo exportador de aceitunas y el tercero de aceite de oliva de Sudamérica (MINCETUR, 2022). Con más de 22 897 hectáreas de olivar y un rendimiento medio de 7 995 kg/ha, la producción y transformación de aceitunas es fundamental para la economía local y regional. Tacna representa el 81,4% de la superficie olivarera que existe en Perú (Baumann, 2022).

A pesar de su importancia, el cultivo del olivo y orégano aún se maneja de forma tradicional y el uso de datos es incipiente. Plagas y enfermedades como la *Orthezia olivicola* y el barrenador del fruto pueden reducir significativamente el número de frutos por cosecha (SENASA, 2021). Existen diversas enfermedades en el cultivo del orégano: hongos como la *Alternaria* o el Oidio ennegrecen y secan la planta, y un virus como el Mosaico decolora las hojas. Este y otros problemas se verán agravados por el inminente cambio climático (FAO, 2019). Los problemas derivados de plagas y enfermedades afectan a la producción y al ciclo productivo ya que afectan a hojas, flores y frutos y pueden iniciar ciclos vecinos (Casilla García, 2011).

Además, la escasez de herramientas de diagnóstico en los países subdesarrollados tiene un impacto devastador en su desarrollo y calidad de vida. Por lo tanto, existe una necesidad urgente de clasificar automáticamente las enfermedades del olivo y orégano con soluciones asequibles y fáciles de usar.

Para reducir las enfermedades de la cosecha del orégano y olivo, es necesario crear y modernizar tecnologías para una productividad eficiente. Una clasificación adecuada y rápida de las enfermedades de la hoja del olivo y orégano podría evitar pérdidas de calidad y de cosecha.

Deep Learning se ha convertido en el método de inteligencia artificial por

excelencia para resolver una gran variedad de problemas, entre ellos los relacionados con la visión por computador (Ekman, 2021). El área de visión por computador abarca un gran número de tareas como segmentación, detección y clasificación, entre las que se encuentran las relacionadas con enfermedades. En problemas de enfermedades, el Deep Learning ha demostrado ser puntero tanto en aplicaciones agrícolas como médicas utilizando arquitecturas CNN (Thomkaew and Intakosum, 2022; Iparraguirre-Villanueva et al., 2022; Yandouzi et al., 2022; Osco-Mamani and Chaparro-Cruz, 2023). Los modelos de Deep Learning aprenden directamente de los datos y requieren grandes conjuntos de datos para obtener buenas precisiones.

Las redes neuronales convolucionales preentrenadas en Imagenet (Deng et al., 2009) han demostrado dar excelentes resultados en la tarea de clasificación (Teo et al., 2022; Al-Dhabyani et al., 2019; Govathoti et al., 2022; Jaiswal et al., 2022; Alam et al., 2022; Kumar et al., 2022; Abubakar et al., 2022; Alharbi et al., 2020) utilizando técnicas como Aumento de Datos (Teo et al., 2022; Al-Dhabyani et al., 2019; Govathoti et al., 2022), Aprendizaje por Transferencia (Jaiswal et al., 2022; Alam et al., 2022; Kumar et al., 2022), y Ajuste Fino (Abubakar et al., 2022; Alharbi et al., 2020) y podrían utilizarse para clasificar las enfermedades de las hojas de orégano y olivo.

Sin embargo, los investigadores han probado experimentalmente varias redes neuronales convolucionales con resultados diversos, pero no han establecido una relación entre sus características y su rendimiento. La necesidad de un amplio estudio comparativo experimental para revelar posibles correlaciones entre las arquitecturas y su rendimiento se convierte en evidente.

Es así que se plantea la siguiente pregunta:

1.2 FORMULACIÓN DEL PROBLEMA

1.2.1 Problema principal

¿Es posible clasificar enfermedades de olivo (*Olea europea*) y orégano (*Origanum vulgare*) mediante modelos de redes neuronales convolucionales en Tacna - Perú?

1.2.2 Problemas secundarios

1. ¿Es posible diseñar arquitecturas de redes neuronales convolucionales para clasificar enfermedades de olivo y orégano?
2. ¿Es posible entrenar modelos de redes neuronales convolucionales para clasificar enfermedades de olivo y orégano utilizando aumento de datos, aprendizaje por transferencia y ajuste fino?
3. ¿Es posible comparar los modelos de redes neuronales convolucionales para clasificar enfermedades de olivo y orégano?
4. ¿Es posible correlacionar las características de las arquitecturas y su rendimiento?

1.3 JUSTIFICACIÓN E IMPORTANCIA DE LA INVESTIGACIÓN

Existen investigaciones relacionadas con la clasificación de enfermedades de las hojas de olivo y orégano (Othman et al., 2022; Prajapati et al., 2023; Lachgar et al., 2022; Alruwaili et al., 2019; Ksibi et al., 2022; Uğuz and Uysal, 2021; Alshammari et al., 2022; Alshammari et al., 2023); sin embargo, para las enfermedades más comunes de las hojas de orégano en La Yarada - Tacna - Perú: Alternaria, Mosaico y Oidio y las enfermedades más comunes del olivo en Susapaya - Tarata - Perú: virosis, fumagina y deficiencias nutricionales, no existe un conjunto de datos para entrenar modelos de visión por ordenador para la clasificación.

Adicionalmente, existen las siguientes razones que justifican y demuestran la importancia del siguiente estudio en base al impacto significativo en la agricultura y la producción de alimentos:

- **Detección temprana de enfermedades:** Los modelos de clasificación de enfermedades en las hojas permiten la detección temprana de enfermedades en las plantas, lo que es esencial para tomar medidas preventivas y evitar la propagación de enfermedades. Cuanto antes se identifique una enfermedad, más efectivas serán las estrategias de control.
- **Reducción de pérdidas de cultivos:** Las enfermedades en las plantas pueden causar daños significativos a los cultivos, lo que lleva a pérdidas económicas para los agricultores y la industria agrícola en su conjunto. La identificación rápida y precisa de las enfermedades ayuda a minimizar estas pérdidas al permitir la aplicación oportuna de tratamientos adecuados.
- **Uso eficiente de recursos:** La clasificación automatizada de enfermedades en hojas de olivo y orégano permite un uso más eficiente de los recursos agrícolas, como pesticidas y fertilizantes. En lugar de aplicar tratamientos de manera indiscriminada, los agricultores pueden dirigir sus esfuerzos solo a las áreas afectadas, reduciendo así el costo y el impacto ambiental de la agricultura.
- **Mejora de la calidad de los productos:** La calidad de los productos agrícolas, como el aceite de oliva y el orégano, está estrechamente relacionada con la salud de las plantas. La detección y el control tempranos de las enfermedades pueden mejorar la calidad de los productos finales al prevenir la contaminación y el deterioro de los cultivos.
- **Sostenibilidad agrícola:** La agricultura sostenible es fundamental para la seguridad alimentaria a largo plazo y la conservación del medio ambiente. La detección y el control precisos de las enfermedades ayudan a promover prácticas agrícola-

las sostenibles al reducir la dependencia de productos químicos y minimizar el impacto negativo en el ecosistema.

- **Investigación y desarrollo:** Los modelos de clasificación de enfermedades en hojas también son una herramienta valiosa para la investigación agrícola. Ayudan a los científicos a comprender mejor las enfermedades de las plantas, su propagación y su impacto, lo que a su vez puede llevar al desarrollo de mejores estrategias de manejo.

1.4 OBJETIVOS

Los objetivos establecen qué se pretende con la investigación, y dado que las investigaciones buscan contribuir a resolver un problema en especial, los objetivos deberán mencionar cuál es y de qué manera se piensa que el estudio ayudará a resolverlo (Hernández Sampieri et al., 2018, p. 36).

Además, los objetivos señalan los objetivos que se esperan, siendo que los objetivos específicos se derivan del objetivo general y contribuyen al logro de este. (Arias, 2012, p. 43-45).

1.4.1 Objetivo general

Clasificar enfermedades de olivo (*Olea europea*) y orégano (*Origanum vulgare*) mediante modelos de redes neuronales convolucionales en Tacna - Perú.

1.4.2 Objetivos específicos

1. Diseñar arquitecturas de redes neuronales convolucionales para clasificar enfermedades de olivo y orégano.
2. Entrenar modelos de redes neuronales convolucionales para clasificar enfermedades de olivo y orégano utilizando aumento de datos, aprendizaje por transferencia

y ajuste fino.

3. Comparar los modelos de redes neuronales convolucionales para clasificar enfermedades de olivo y orégano.
4. Correlacionar las características de las arquitecturas y su rendimiento.

1.5 HIPÓTESIS

Los objetivos y preguntas de investigación son susceptibles de reafirmarse o mejorarse durante el desarrollo del estudio, dando origen a otras hipótesis que no estaban planteadas originalmente, producto de nuevas reflexiones, ideas o experiencias. En las investigaciones cuantitativas es posible plantear hipótesis que intenten pronosticar un hecho. Sin embargo, al respecto de las hipótesis, no hay reglas universales. Los estudios que contienen análisis de datos cuantitativos suelen incluir hipótesis de investigación únicamente. Las hipótesis de investigación son proposiciones tentativas, cuya utilidad es servir de guías de investigación y ayudan a saber lo que tratamos de probar, proporcionando orden y lógica al estudio, pudiendo ser soluciones a los problemas de investigación (Hernández Sampieri et al., 2018, p. 92,93,96,106,108).

1.5.1 Hipótesis general

Sí es posible clasificar enfermedades de olivo (*Olea europea*) y orégano (*Origanum vulgare*) mediante modelos de redes neuronales convolucionales en Tacna - Perú.

1.5.2 Hipótesis específicas

1. Sí es posible diseñar arquitecturas de redes neuronales convolucionales para clasificar enfermedades de olivo y orégano.

2. Sí es posible entrenar modelos de redes neuronales convolucionales para clasificar enfermedades de olivo y orégano utilizando aumento de datos, aprendizaje por transferencia y ajuste fino.
3. Sí es posible comparar modelos de redes neuronales convolucionales para clasificar enfermedades de olivo y orégano.
4. Sí es posible correlacionar las características de las arquitecturas y su rendimiento.

1.6 VARIABLES

1.6.1 Identificación de variables

Variable Independiente: Modelo de red neuronal convolucional

Variable Dependiente: Clasificación de enfermedades de olivo y orégano

1.6.2 Definición conceptual de las variables

Variable Independiente: Modelo de Red Neuronal Convolucional

Por su naturaleza: Cuantitativa

Por su escala de medición: Numérica de razón

Variable Dependiente: Clasificación de enfermedades de olivo y orégano

Por su naturaleza: Cuantitativa

Por su escala de medición: Numérica de razón

1.6.3 Definición operacional de las variables

En la presente investigación se midió el efecto de la variable independiente sobre la dependiente para luego correlacionar las características de la variable independiente

con el efecto en la variable dependiente.

Variable independiente: Modelo de red neuronal convolucional

Dimensiones e indicadores:

- Criterios de selección
 - Tamaño (MB).
 - Exactitud Top-1 en Imagenet.
 - Exactitud Top-5 en Imagenet.
 - Número de parámetros.
 - Profundidad.
 - Tiempo (ms) por inferencia.

Variable dependiente: Clasificación de enfermedades de olivo y orégano

Dimensiones e indicadores:

- Rendimiento
 - Exactitud (accuracy).
 - Precisión (precision).
 - Memoria (recall).
 - F1-Score.
- Pérdida (loss)
- Criterios de selección
 - Tamaño (MB).
 - Exactitud Top-1 en Imagenet.
 - Exactitud Top-5 en Imagenet.

- Número de parámetros.
- Profundidad.
- Tiempo (ms) por inferencia.

1.7 LIMITACIONES DE LA INVESTIGACIÓN

Entre las limitaciones más relevantes se menciona las siguientes:

- En cuanto a la muestra, está constituida por conjuntos de datos de fotografías tomadas en cultivos de la región, específicamente de La Yarada - Tacna para olivo y de Susapaya - Tarata - Tacna para el orégano.
- En cuanto a las enfermedades que se clasifican, solo se han considerado las 03 más comunes: Virosis, fumagina y deficiencia nutricional para olivo y alternaria, mosaico y oidiosis para orégano.
- En los conjuntos de datos de orégano y olivo no se tomó en consideración la clase "Ninguna" (ausencia de enfermedad). La toma de imágenes para el conjunto de datos fue posterior al planteamiento de esta investigación.

CAPÍTULO II

MARCO TEÓRICO

2.1 ANTECEDENTES DEL ESTUDIO

Existen investigaciones relacionadas con la clasificación de enfermedades de las hojas de orégano utilizando redes neuronales convolucionales (Prajapati et al., 2023; Othman et al., 2022). Sin embargo, para las enfermedades más comunes de las hojas de orégano en la región de Tacna, Perú: Alternaria, Mosaico y Oidio no existe un conjunto de datos disponibles para el entrenamiento de modelos de clasificación.

Existen investigaciones relacionadas con la clasificación de enfermedades de la hoja del olivo (Lachgar et al., 2022) con conjuntos de datos recogidos en países como Arabia Saudí (Alruwaili et al., 2019; Ksibi et al., 2022) o Turquía (Uğuz and Uysal, 2021; Alshammari et al., 2022; Alshammari et al., 2023). Sin embargo, para las enfermedades más comunes en la región de Tacna-Perú: virosis, fumagina y deficiencias nutricionales, no existe un conjunto de datos para entrenar modelos de visión por ordenador para la clasificación.

Los resultados experimentales de algunos estudios (Yandouzi et al., 2022; Osco-Mamani and Chaparro-Cruz, 2023; Abubakar et al., 2022) de clasificación de enfermedades en hojas de orégano y olivo muestran a la arquitectura VGG16 con resultados superiores en comparación con Inceptionv3, ResNet50V2, InceptionResNetV2, Xception, Dense-Net, MobileNet, MobileNetV2 y NASNetMobile. Adicionalmente, otros estudios (Prajapati et al., 2023; Teo et al., 2022; Jaiswal et al., 2022; Alam et al., 2022;

Kumar et al., 2022) obtienen resultados diversos en cuanto a qué arquitectura es mejor.

2.2 BASES TEÓRICAS

2.2.1 Deep Learning

El aprendizaje profundo es un subcampo del aprendizaje automático que consiste en utilizar múltiples capas de procesamiento no lineal para aprender representaciones de datos con múltiples niveles de abstracción (Goodfellow et al., 2016).

2.2.2 Redes neuronales convolucionales

El origen de las redes neuronales convolucionales se remonta a la década de 1950, cuando David Hubel y Torsten Wiesel describieron células simples y complejas en el córtex visual de un gato (Hubel and Wiesel, 1959). En la década de 1980, el Dr. Kunihiko Fukushima, inspirado por el experimento de Hubel y Wiesel, propuso el modelo de neocognición (Fukushima, 1980) que capta patrones mediante “células” (simples) y “células” (complejas) a través de un modelo computacional.

En la década de 1990, Yann LeCun demostró que un modelo de red neuronal convolucional (LeCun et al., 1998), que combina características básicas en otras cada vez más complejas, podía realizar eficazmente la tarea de clasificación de reconocimiento de caracteres manuscritos. El boom de la IA se produjo en 2012, después de que una CNN llamada AlexNet (Krizhevsky et al., 2012) consiguiera un rendimiento puntero clasificando imágenes en el Desafío de Reconocimiento Visual a Gran Escala (ILSVRC) (Deng et al., 2009).

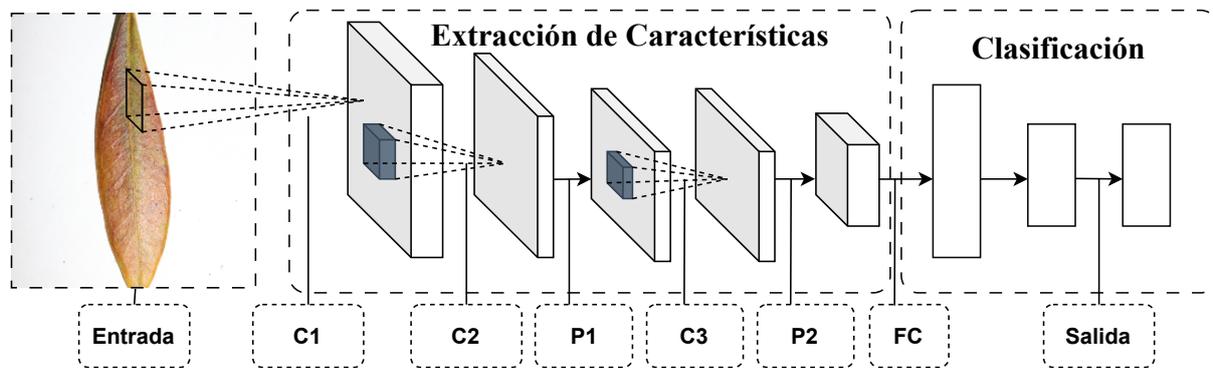
Las redes neuronales convolucionales (CNN) están diseñadas para procesar múltiples tipos de datos, especialmente imágenes bidimensionales, y se inspiran directamente en la corteza visual del cerebro. En la corteza visual existe una jerarquía de células básicas: simples y complejas (Min et al., 2017). Una CNN puede extraer

las características de la imagen sin necesidad de realizar este proceso a mano como antes de 2011, finalmente, añadiendo una red neuronal al final permite realizar la tarea de clasificación.

La Figura 1 muestra una arquitectura típica de CNN. La extracción de características consta de capas de convolución (C1, C2, C3) y capas de agrupación (P1, P2). La clasificación consta de capas totalmente conectadas (FC) y una capa de salida (Zhang et al., 2019).

Figura 1

Estructura típica de una Red Neuronal Convolutiva



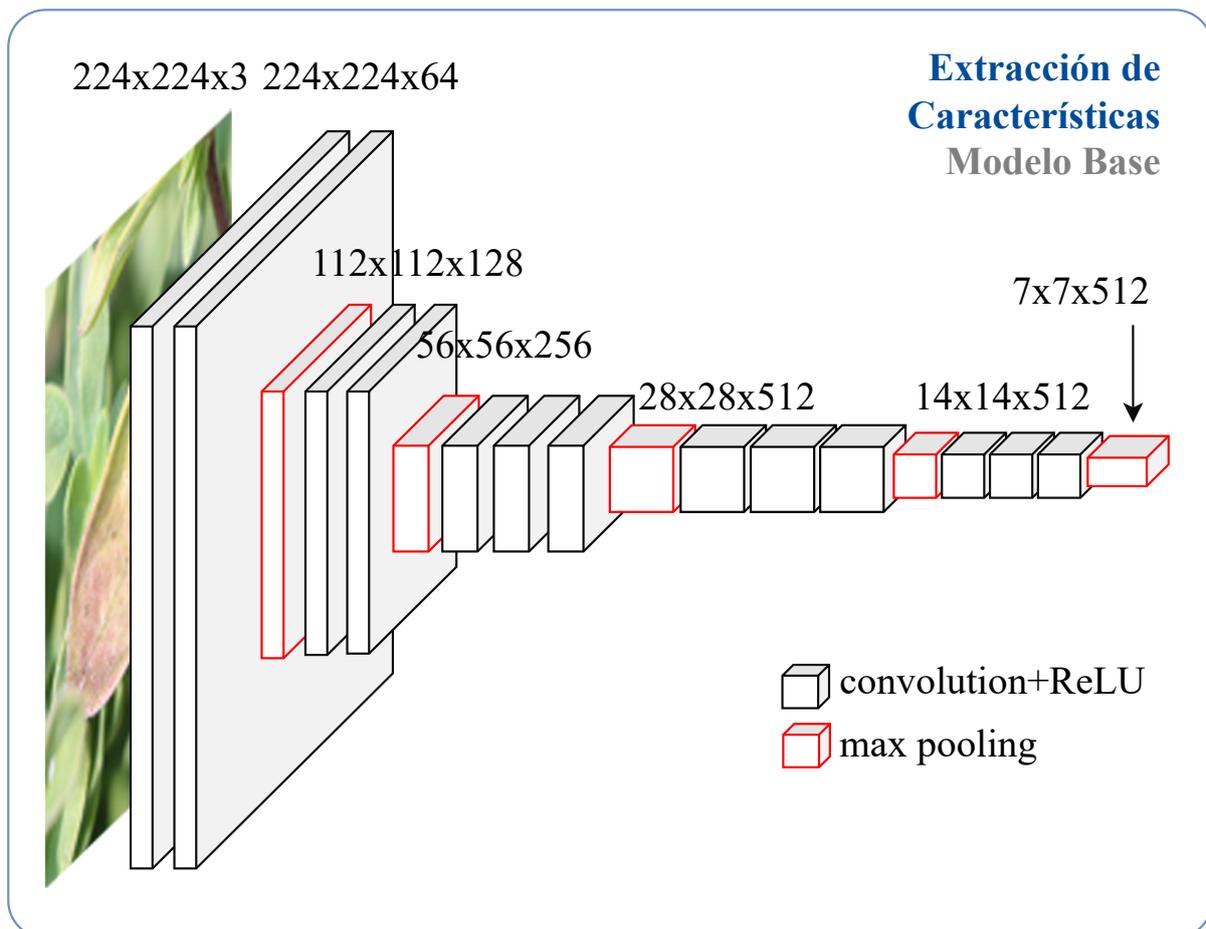
Nota: Elaboración propia

En los años siguientes, las CNN han obtenido excelentes resultados en el conjunto de datos de referencia ImageNet (Deng et al., 2009), y se han propuesto muchas arquitecturas a lo largo de los años. A continuación, se explora algunas de ellas:

2.2.3 Arquitectura VGG16

Figura 2

Arquitectura del modelo VGG-16



Nota: Elaboración propia

La arquitectura VGG16 (Simonyan and Zisserman, 2014) consta de 13 capas convolucionales para la extracción de características, como se muestra en la Figura 2. VGG16 es un modelo de clasificación capaz de clasificar 14M de imágenes de 1000 categorías diferentes (Imagenet dataset - ILSVRC 2014) con un 92,7% de exactitud. A pesar de la existencia de modelos más recientes, la simplicidad de esta arquitectura la hace perfecta para aplicar técnicas como Aprendizaje por Transferencia y Ajuste Fino.

La red (modelo y pesos entrenados) está disponible en Keras.

Las convoluciones se realizan con núcleos 3x3, seguidos de una función de activación ReLU. A cada grupo de capas de convolución le sigue una capa de max-pooling, y las operaciones de max-pooling se realizan con núcleos de 2x2, reduciendo el tamaño del mapa de características a la mitad. La arquitectura original de VGG16 incluye tres capas totalmente conectadas + una capa softmax como modelo de clasificación o superior. De ahí que el nombre incluya 16.

Las 13 capas convolucionales de VGG16 son convoluciones estándar. Las convoluciones estándar son uno de los bloques de construcción fundamentales en Deep Learning para tareas de procesamiento de imágenes. Consiste en deslizar un filtro (un núcleo) sobre la imagen de entrada para realizar una multiplicación y una suma de elementos, lo que genera mapas de características. Estos mapas de características capturan diferentes patrones y jerarquías espaciales dentro de la imagen de entrada.

Standard Convolution

El artículo original que introdujo las redes neuronales convolucionales fue de Yann LeCun et al: "Gradient-Based Learning Applied to Document Recognition" (LeCun et al., 1998).

Dada una imagen o mapa de características I y un filtro o kernel K , donde (x, y) representa la posición espacial en el mapa de características de salida, y a y b representan la media anchura y la media altura del filtro, respectivamente, el resultado de aplicar una convolución estándar es:

$$(I * K)(x, y) = \sum_{i=-a}^a \sum_{j=-b}^b I(x+i, y+j) \cdot K(i, j) \quad (1)$$

El valor resultante en la posición (x, y) del mapa de características de salida es la suma ponderada de los valores de los píxeles de la imagen de entrada que se

solapan con el filtro, y esta operación se realiza para todas las posiciones del mapa de características de salida. El número de mapas de características de salida es igual al número de filtros aprendidos en la capa de convolución.

Para las convoluciones estándar, el coste computacional puede expresarse mediante la siguiente fórmula:

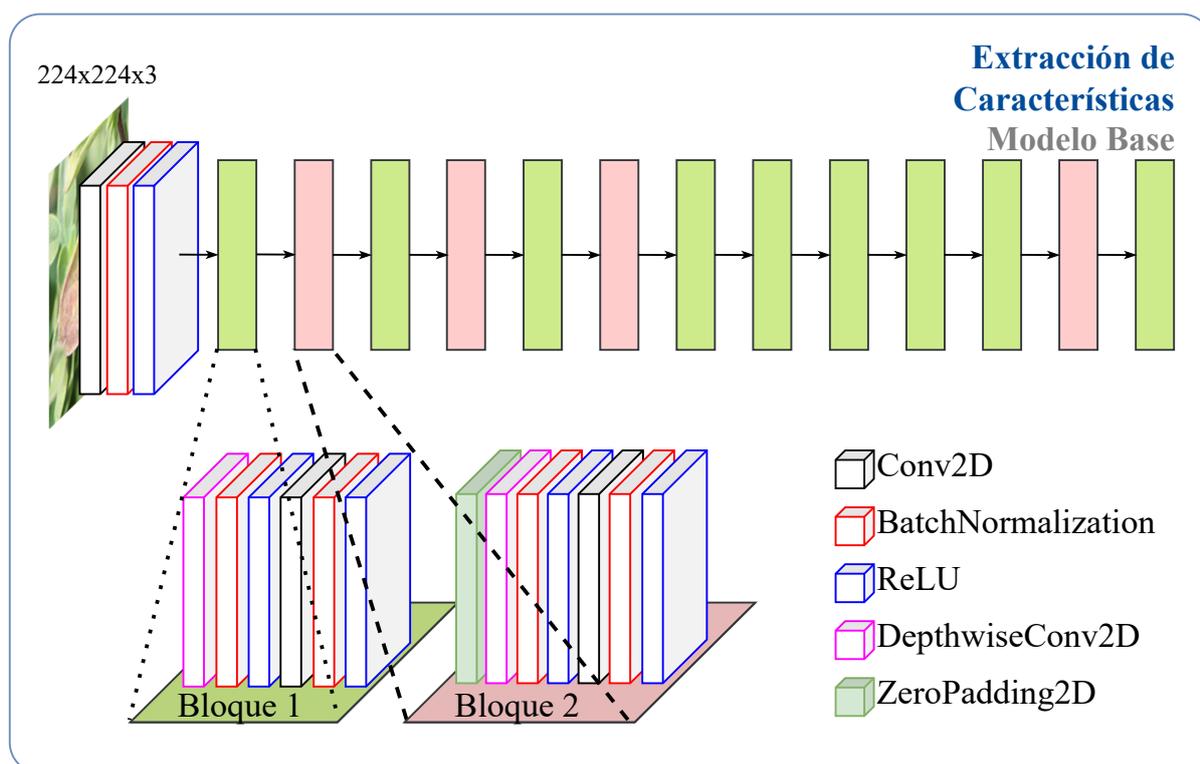
$$D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F \quad (2)$$

Donde M es el número de canales de entrada de I , N es el número de canales de salida, $D_k \times D_k$ es el tamaño del kernel, y $D_F \times D_F$ es el tamaño del mapa de características de salida.

2.2.4 Arquitectura MobileNet

Figura 3

Arquitectura del modelo MobileNet



Nota: Elaboración propia

La arquitectura MobileNet fue propuesta por Google Engineers en 2017 (Howard et al., 2017). Antes de 2017, la tendencia en las arquitecturas de redes neuronales era hacerlas más eficaces siendo más profundas y complicadas aumentando el número de parámetros para obtener precisiones más mejoradas. Sin embargo, esto no era bueno para el despliegue en aplicaciones de visión móviles e integradas, donde es necesario ser eficiente.

Para resolver este problema, MobileNet introduce convoluciones separadas en profundidad, que dividen la operación convolucional en dos pasos: convolución en profundidad y convolución puntual, sin sacrificar exactitud.

Depthwise convolution

Aplica un único filtro a cada canal de entrada.

Dada c la representación del índice de la imagen de entrada I , el resultado de aplicar una convolución en profundidad es:

$$(I * K_c)(x, y) = \sum_{i=-a}^a \sum_{j=-b}^b I(x+i, y+j, c) \cdot K_d(i, j) \quad (3)$$

El coste computacional de las convoluciones en profundidad se muestra en la siguiente ecuación:

$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F \quad (4)$$

Pointwise convolution

Es una convolución estándar 1x1 que combina estas características reducidas en un nuevo conjunto de características más complejas.

Dada la salida de la convolución en profundidad I' , el resultado de aplicar una convolución puntual es:

$$(I' * K)(x, y) = I(x+i, y+j) \cdot K(i, j) \quad (5)$$

El coste computacional de las convoluciones puntuales se muestra en la siguiente ecuación:

$$M \cdot N \cdot D_F \cdot D_F \quad (6)$$

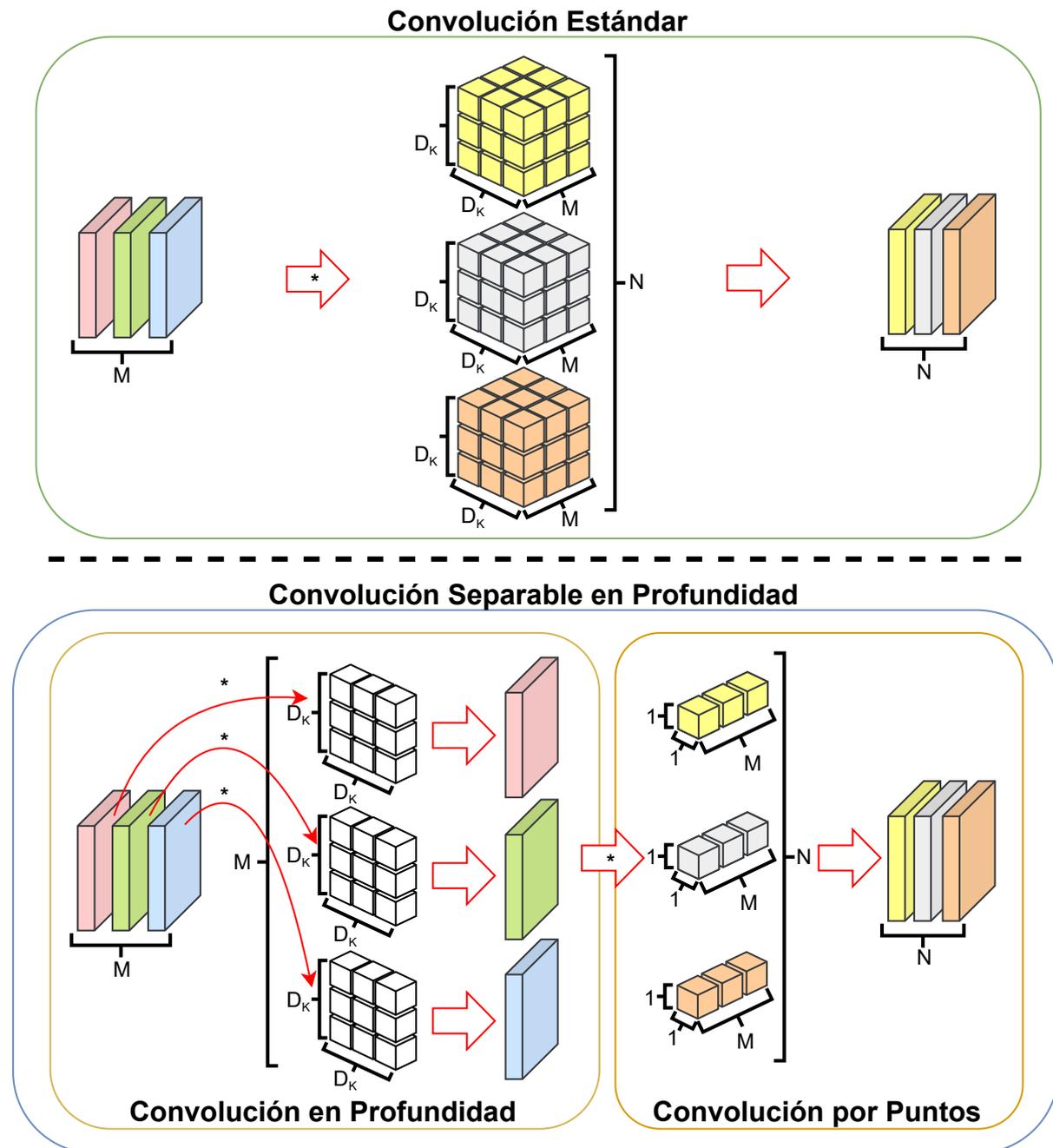
Estos dos componentes reducen la complejidad computacional al separar las operaciones espaciales y de canal. Matemáticamente, la suma de los costes compu-

tacionales de la convolución en profundidad mostrada en la Ecuación 4 y la convolución en puntos mostrada en la Ecuación 6 son de 8 a 9 veces menos que el coste computacional de las convoluciones estándar mostrada en la Ecuación 2. La Figura 4 muestra una comparación visual entre estas operaciones.

Cabe mencionar que, después de cada una de estas convoluciones (depthwise y pointwise), se aplica una normalización por lotes y una activación de capas ReLU como se puede ver en el modelo base de la Figura 3.

Figura 4

Convención estándar vs convención separable en profundidad

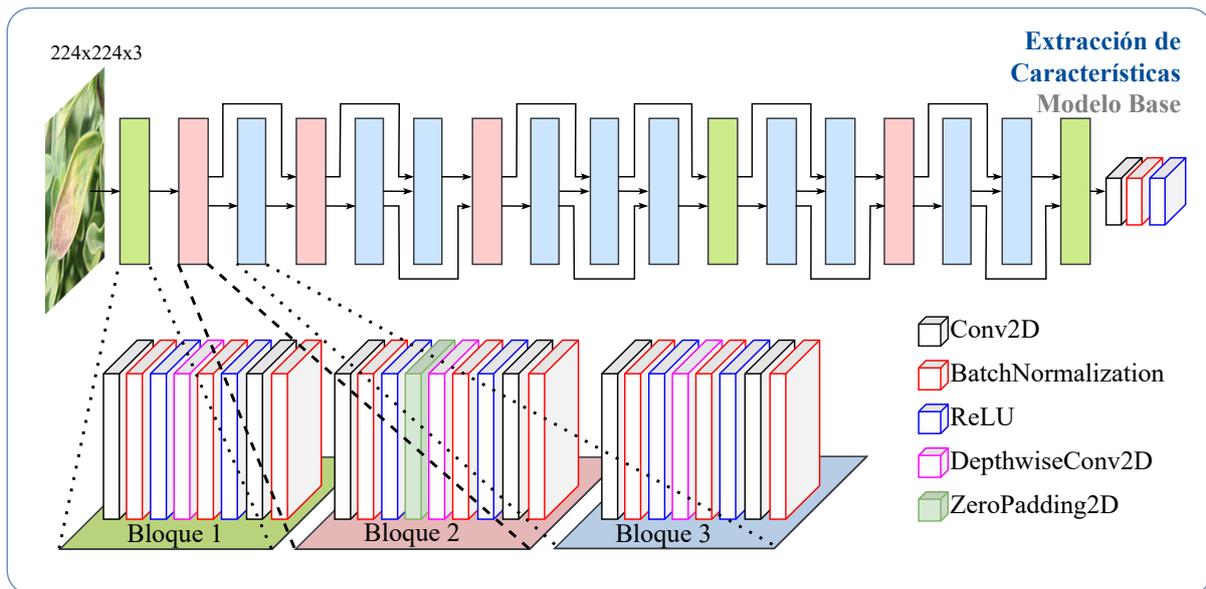


Nota: Elaboración propia

2.2.5 MobileNetV2

Figura 5

Arquitectura del modelo MobileNetV2



Nota: Elaboración propia

El modelo MobileNetV2 propuesto en 2018 (Sandler et al., 2018), fue una versión mejorada de la arquitectura MobileNet original donde se introdujeron optimizaciones como los residuales invertidos y los cuellos de botella lineales para mejorar las capacidades de representación de características, como se puede ver en el modelo base de la Figura 5.

Los bloques residuales invertidos son una variante de los bloques residuales tradicionales propuestos en ResNet (He et al., 2016). A diferencia de los bloques residuales tradicionales que aumentan la dimensionalidad, los bloques residuales invertidos primero disminuyen la dimensionalidad antes de aumentarla. Estos bloques residuales invertidos buscan capturar primero información más detallada antes de disminuir la dimensionalidad, lo que les permite ser más eficientes computacionalmente.

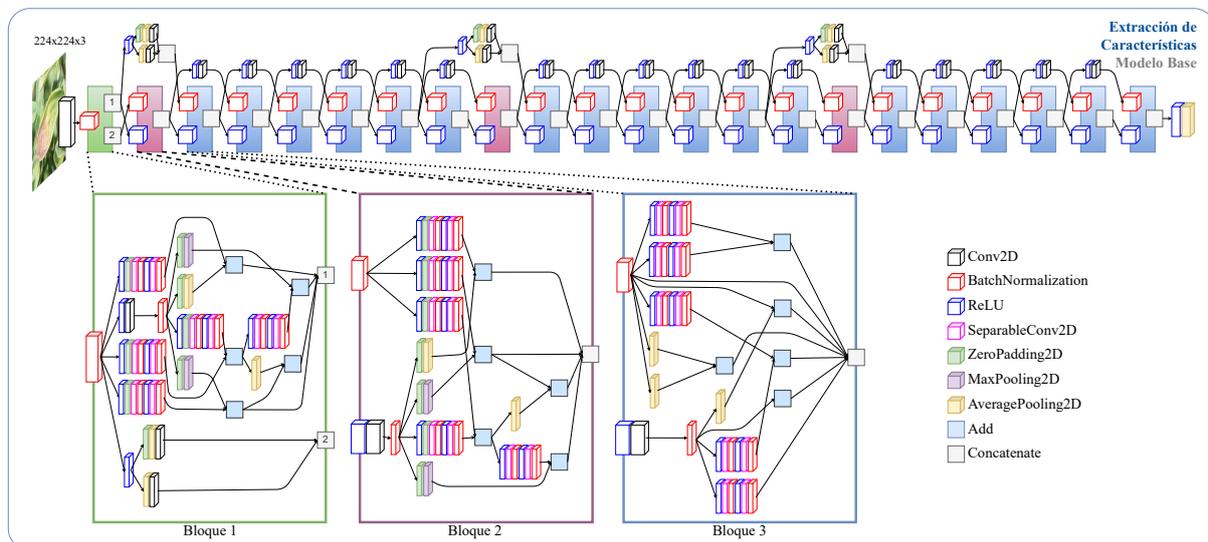
Además, los cuellos de botella lineales permiten controlar la cantidad de información transmitida entre capas, lo que da lugar a una arquitectura más eficiente.

Estas modificaciones, junto con las convoluciones separadas en profundidad, mejoraron la eficacia y la capacidad de representación de características del modelo.

2.2.6 NASNetLarge

Figura 6

Arquitectura del modelo NASNetLarge



Nota: Elaboración propia

La arquitectura NASNetLarge lanzada en 2017 (Zoph et al., 2018) fue desarrollada por el marco de búsqueda de arquitectura neuronal (NASNet) (Zoph and Le, 2016) que utiliza el aprendizaje de refuerzo para encontrar la estructura óptima de una red neuronal para una tarea específica en lugar de diseñar manualmente una arquitectura.

El conjunto de operaciones utilizadas como bloques de construcción para crear arquitecturas de redes neuronales complejas mediante un proceso de búsqueda de arquitecturas neuronales incluye convoluciones estándar y convoluciones separables

en profundidad, como se muestra en la Figura 6, donde algunas capas se agruparon en bloques para representar toda la arquitectura base.

2.2.7 Subajuste y sobreajuste

El subajuste es un escenario en el que el modelo no se ajusta al conjunto de datos de entrenamiento, lo que provoca una elevada tasa de error. En el entrenamiento a través de épocas, esto ocurre cuando el modelo no es suficientemente capaz de modelar la relación entre la entrada y la salida.

Cuando un modelo funciona bien con datos de entrenamiento, es necesario considerar un grupo de control para garantizar que el modelo funciona bien con datos con los que no ha sido entrenado. Ese grupo de control se conoce como conjunto de validación. Es posible que el modelo funcione muy bien con los datos de entrenamiento, pero mal con los datos de validación, lo que se conoce como sobreajuste o sobreajuste (falta de generalización).

2.2.8 Aprendizaje por transferencia

El aprendizaje por transferencia es una técnica de aprendizaje automático que trata de aprovechar un modelo ya entrenado para una tarea (preentrenado) en otra (Ekman, 2021), basándose en el supuesto de que los datos con los que se ha entrenado inicialmente el modelo se encuentran en el mismo espacio de características y tienen la misma distribución que los nuevos datos. En los casos en que esto último sea cierto, se podría utilizar la base de extracción de características congeladas de un modelo preentrenado y vincularla a una nueva red de clasificación (con pesos inicializados aleatoriamente) para entrenar sólo esta última y obtener buenos resultados.

Este proceso da lugar a una rápida convergencia del modelo, ya sea debido a las restricciones de extracción de características que se han congelado o debido a la buena compatibilidad de los dominios del conjunto de datos. Cuando esto no es

suficiente, también es necesario entrenar la base de extracción de características, mediante una técnica denominada Ajuste Fino.

2.2.9 Ajuste fino

El ajuste fino es una técnica que se aplica sobre el aprendizaje por transferencia para terminar de ajustar el modelo al nuevo conjunto de datos (Ekman, 2021), para ello se descongelan las capas de extracción de características y se entrena el modelo completo. Es importante que este proceso se lleve a cabo después de haber entrenado las capas de clasificación del modelo y haber seleccionado la mejor época para el ajuste fino.

El ajuste fino permite al modelo mejorar la extracción de características para obtener un mejor resultado en la tarea de clasificación. La necesidad de utilizar el Ajuste Fino, indica que el espacio de características y la distribución de los conjuntos de datos (preentrenados y nuevos) no son exactamente iguales, la distancia entre el espacio de características y la distribución de los conjuntos de datos puede reflejarse en la mejora del modelo.

2.2.10 Aumento de datos

El aumento de datos es la generación de datos sintéticos mediante la perturbación de los datos originales (Ekman, 2021), esto permite disponer artificialmente de una mayor cantidad de datos y evita sesgos relacionados con las perturbaciones realizadas. El uso de esta técnica reduce la necesidad de disponer de grandes cantidades de datos para entrenar modelos de Deep Learning. Además, cuando los datos son difíciles de obtener, es posible utilizar el aumento de datos como técnica de regularización para generalizar mejor.

El aumento de datos también evita el sobreajuste del modelo al añadir diversidad y cierta aleatoriedad al conjunto de datos, lo que permite entrenar los modelos durante

más épocas. Las deformaciones o transformaciones más utilizadas están relacionadas con la rotación, el zoom, el desplazamiento de anchura y altura, y los giros horizontales y verticales. Es muy importante utilizar estas transformaciones teniendo en cuenta el dominio del problema para no obtener resultados no deseados.

2.3 DEFINICIÓN DE TÉRMINOS

Matriz de confusión

Una matriz de confusión es una tabla NxN que resume el número de predicciones correctas e incorrectas que hizo un modelo de clasificación, donde N es el número de clases. Se calculan los siguientes valores: *TP* son verdaderos positivos, *FP* falsos positivos, *FN* falsos negativos y *TN* son verdaderos negativos. Teniendo en cuenta estos valores, se pueden calcular diferentes métricas de clasificación.

Exactitud (Accuracy)

Número de predicciones correctas sobre el número total de predicciones.

Pérdida (Loss)

Es una medida de cuánto error se está cometiendo en la predicción. En concreto, este valor será 0 cuando la predicción sea igual a la salida deseada. Para la clasificación multiclase, se suele utilizar la pérdida de entropía cruzada categórica, cuya definición es la siguiente:

$$L_{CE} = - \sum_{i=1}^n t_i \log(p_i), \text{ for } n \text{ classes} \quad (7)$$

donde t_i es la etiqueta de verdad y p_i es la probabilidad Softmax para la clase i^{th} .

Precisión (Precision)

Intente responder a la siguiente pregunta: ¿Qué proporción de identificaciones positivas fue correcta? En el caso de la clasificación multiclase, se obtiene una métrica por clase. Se define como sigue:

$$Precision = \frac{TP}{TP + FP} \quad (8)$$

Sensibilidad (Recall)

Intente responder a la siguiente pregunta: ¿Qué proporción de positivos reales se identificó correctamente? En el caso de la clasificación multiclase, se obtiene una métrica por clase. Se define como sigue:

$$Sensibilidad = \frac{TP}{TP + FN} \quad (9)$$

F1-Score

Se trata de una única puntuación definida como la media armónica de la precisión y la sensibilidad. En el caso de la clasificación multiclase, se obtiene una métrica por clase. Se define como sigue:

$$F_1 = 2 \frac{precision \cdot sensibilidad}{precision + sensibilidad} = \frac{2TP}{2TP + FP + FN} \quad (10)$$

2.4 MARCO FILOSÓFICO

El interés por la inteligencia artificial ha incrementado exponencialmente en los últimos años y es innegable que la inteligencia artificial se ha vuelto parte de nuestra vida cotidiana. Las preguntas en cuanto a inteligencia artificial que se abordarán en el

presente marco filosófico son: ¿Cómo funciona la IA? ¿Por qué funciona? ¿Cuáles son sus límites?

Epistemología es la base general para todo lo relacionado con el conocimiento científico y la resolución de problemas. Dentro de la ciencia se discute que tipo de ciencia es mejor, fuera de la ciencia se discuten estrategias precientíficas de resolución de problemas, que es lo que los cerebros utilizan la mayor parte del tiempo.

En el libro de Kahneman titulado “Thinking, fast and slow” se discute la idea de que las mentes humanas usan dos procesos diferentes y complementarios, dos modos diferentes de pensar, que llamamos comprensión y razonamiento. La idea ha sido discutida por décadas y ha sido verificada usando estudios psicológicos y por neurociencia.

Tabla 1

Entendimiento vs razonamiento

Entendimiento	Razonamiento
Rápido	Lento
Paralelo	Paso a paso
Intuitivo	Lógico
Subconsciente	Consciente
Caro	Eficiente
Libre de modelos	Basado en modelos

“Entendimiento intuitivo subconsciente” es el nombre completo del pensamiento “Pensamiento rápido” o “Sistema 1”. Es rápido porque el cerebro puede realizar muchas partes de esta tarea en paralelo. El cerebro dedica mucho esfuerzo a esta tarea.

“Razonamiento lógico consciente” es el nombre completo de pensamiento “Pensamiento lento” o “Sistema 2”. Para sorpresa de muchas personas, esto se usa muy poco en la práctica. Casi todo lo que hacemos diariamente en nuestra rica realidad se hace sin necesidad de razonar al respecto; solo repetimos lo que funcionó la última vez que realizamos esta tarea y somos impulsados por la experiencia.

“Intuitivo” significa que el sistema puede “proporcionar soluciones” muy rápidamente a problemas muy complejos, pero esas soluciones pueden no ser correctas en todo momento.

“Lógico” significa que las respuestas siempre son correctas siempre que los datos de entrada sean correctos y suficientes. Lo cual no es verdad en nuestra rica realidad; solo puede ser verdad en un espacio de “Modelo” matemáticamente puro.

“Subconsciente” significa que no tenemos acceso consciente (“introspectivo”) a estos procesos. El proceso de leer una oración y comprenderla completamente no puede ser explicado ni entender en sí cómo o por qué se entiende.

“Consciente” significa que somos conscientes del pensamiento; podemos acceder a él a través de la introspección y podemos encontrar razones por las que creemos una idea determinada.

“Caro” está en la lista porque los cerebros dedican la mayor parte de su esfuerzo a esta parte de la Comprensión. Realmente no deberíamos sorprendernos de que la IA ahora requiera computadoras muy poderosas.

Por el contrario, el razonamiento es “eficiente”. Es más útil cuando está atrapado en una situación novedosa donde la experiencia y la comprensión ayudan. O tal vez se necesita planificar por adelantado, o necesita encontrar razones para explicar por qué sucedió algo después del hecho. Se utiliza a nivel formal en las ciencias. El razonamiento es importante, pero rara vez se necesita o se usa.

Finalmente, la comprensión es “libre de modelos” y el razonamiento está “basado en modelos”. Esta es probablemente la distinción más importante para las personas que están implementando sistemas inteligentes, ya que proporciona una manera de mantener la implementación en el camino correcto cuando las cosas se ponen difíciles.

Sabemos desde hace mucho tiempo que los cerebros usan estos dos modos. Pero la comunidad de investigación de inteligencia artificial ha realizado un esfuerzo excesivo en la parte de razonamiento y ha estado ignorando la parte de entendimiento

durante sesenta años.

Se tenían varias buenas razones para esto. Hasta hace poco, nuestras máquinas eran demasiado pequeñas para ejecutar cualquier red neuronal de tamaño útil. Además, no teníamos ni idea de cómo implementar este Entendimiento. Pero eso es exactamente lo que cambió en 2012 cuando un grupo de investigadores de inteligencia artificial de Toronto encabezados por Geoffrey Hinton demostró efectivamente que Deep Neural Networks podría proporcionar un tipo simple de compresión al ganar la competencia Imagenet con una red llamada AlexNet (basada en Redes Neuronales Convolucionales Profundas).

Esta nueva fase de la inteligencia artificial tardó décadas en desarrollarse, pero nunca habría ocurrido sin personas como el grupo liderado por Geoffrey Hinton en la Universidad de Toronto, que pasó más de 34 años desarrollando la tecnología de la red neuronal profunda que ahora llamamos “Aprendizaje profundo”. Una serie de avances desde 1997 hasta 2006 condujeron a una serie de demostraciones exitosas (incluidos los primeros premios en competiciones de inteligencia artificial) en 2012 y, por lo tanto, contamos ese año como el año de nacimiento de Machine Understanding.

Para un forastero en la materia, puede parecer que un programa o una aplicación de teléfono puede “comprender” lo que sea que esté haciendo la aplicación, pero ese entendimiento realmente solo sucedió en la mente del programador que crea la aplicación. El programador primero simplificó el problema en su propia cabeza al descartar una gran cantidad de detalles irrelevantes usando “entendimiento del programador”. Este “modelo” mental simplificado del dominio del problema podría luego explicarse a una computadora en forma de un programa de computadora.

Lo que está cambiando es que las computadoras ahora están haciendo estos modelos.

La inteligencia artificial, desde 1955, ha proporcionado muchos algoritmos novedosos y brillantes que ahora utilizamos en todos los programas. Pero cuando se compara la inteligencia artificial de estilo antiguo con los sistemas de entendimiento, el

viejo tipo de inteligencia artificial es básicamente indistinguible de cualquier otro tipo de programación que hacemos hoy en día.

Las redes neuronales profundas son tan diferentes de los programas regulares que tenemos que reconocerlas como un paradigma computacional diferente. Por eso tardaron casi cuatro décadas en desarrollarse. Y el paradigma, por ser precientífico y sin modelos, es difícil de entender si se recibe una sólida educación reduccionista y basada en modelos; toma mucho tiempo para que un practicante de IA establecido o un programador experimentado lo cambie. Las personas que recién están comenzando en la IA tienen un tiempo más fácil para asimilar este nuevo paradigma, ya que no han tenido la experiencia y el éxito de una carrera completa utilizando técnicas de IA de estilo antiguo.

La cantidad de trabajo que se tiene que hacer para obtener una Deep Neural Network to Understand es sorprendentemente pequeña, y compañías como Google y Facebook están trabajando para eliminar el esfuerzo restante de la programación de Neural Networks. Aquí es donde las cosas se pondrán realmente extrañas: cuando la Red neuronal profunda (DNN) comprende lo suficiente sobre el mundo y sobre el problema al que se enfrenta, ya no necesitamos un programador para adquirir este entendimiento.

Los programadores son empleados para unir dos dominios diferentes. Primero tienen que estudiar cualquier dominio de aplicación en el que estén trabajando. Por ejemplo, si están escribiendo un sistema de reservación de boletos de avión, tendrán que aprender mucha información detallada sobre aerolíneas, boletos de avión, vuelos, equipaje, etc., y luego se les proporcionará funciones para casos inusuales, como vuelos cancelados. Y luego el programador usa su comprensión del dominio del problema para explicar a una computadora cómo puede razonar sobre estas cosas, pero el programador no puede hacer que el sistema entienda, solo puede poner un tipo de razonamiento hueco y frágil, como un programa con muchos casos de if-else. Y cualquier malentendido que el programador tenga sobre el dominio del problema se

convertirá en “errores” en el programa de computadora.

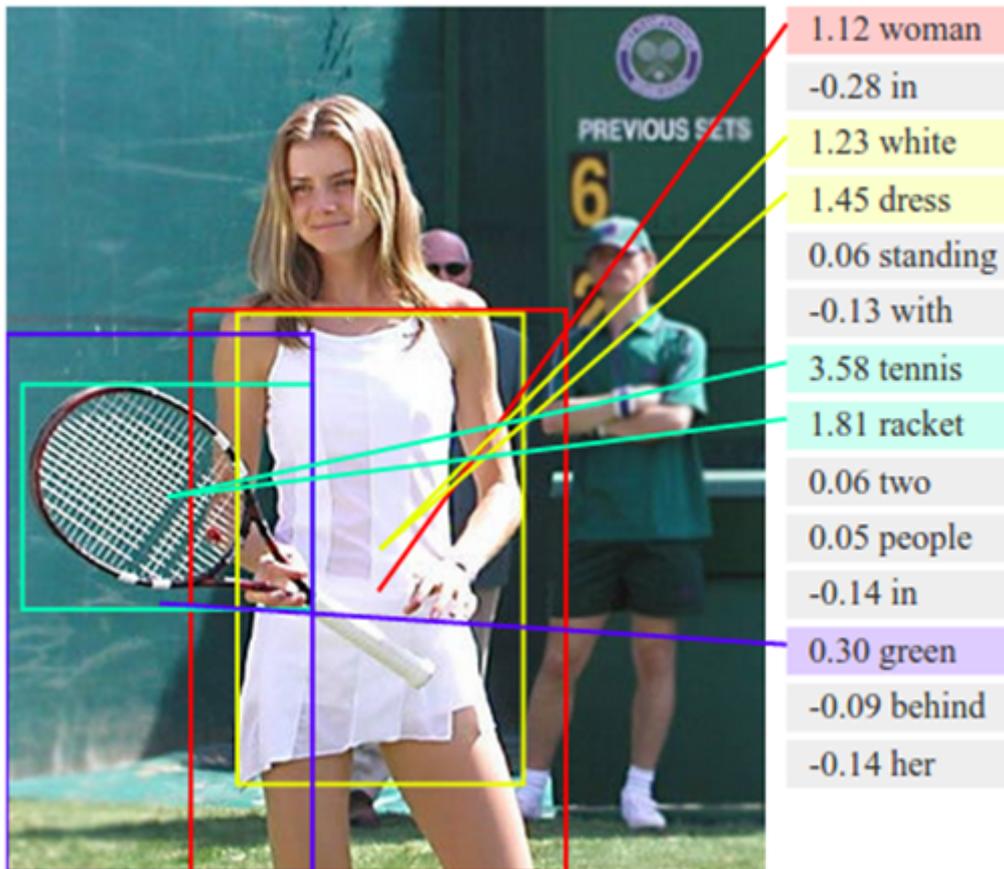
Pero hoy, para ciertas clases de problemas moderadamente complejos, podemos usar un DNN para aprender automáticamente por sí mismos cómo entender el problema. Lo que significa que ya no necesitamos un programador para entender el problema. Hemos delegado nuestro entendimiento a una máquina.

Eso es exactamente lo que debería hacer una IA. Debe comprender todo tipo de cosas para que los humanos no tengan que hacerlo. Y hay dos situaciones comunes en las que esta será una muy buena idea. Una es que cuando tenemos un problema que no podemos explicarlo nosotros mismos.

El otro caso común será cuando entendamos bien el problema, pero hacer que una máquina lo entienda lo suficientemente bien como para hacer el trabajo es más barato y más fácil que cualquier otra alternativa.

Figura 7

Ejemplo de image captioning



Nota: Extraído de: <https://ceslava.com/blog/software-de-ia-de-google-capaz-de-describir-lo-que-hay-en-una-fotografia/>

Si le damos una Figura 7 como está a un DNN (Deep Neural Network) entrenado en imágenes, identificará los objetos importantes en la imagen y proporcionará los rectángulos, llamados “cuadros delimitadores”, como aproximaciones a donde están los objetos. El texto a la derecha dice “Mujer con vestido blanco de pie con raqueta de tenis, dos personas en verde detrás de ella”. Lo cual no es una mala descripción de la imagen. Para todos los propósitos prácticos, esto es comprensión.

No teníamos idea de cómo hacer que nuestras computadoras hagan esto antes de 2012. Esto es realmente un gran problema. Esta hazaña requiere no solo un nuevo

algoritmo, requiere un nuevo paradigma computacional.

Una imagen es, para una computadora, una larga secuencia de números que denota valores para los colores rojo, azul y verde en valores de 0 a 255; también sabemos qué tan amplia es la imagen. ¿Cómo se obtiene esta representación de tan bajo nivel para saber que hay una mujer con una raqueta de tenis en la imagen?

A esto se le ha llamado “Un río que fluye cuesta arriba”. Hay muy pocos mecanismos que pueden ir en esta dirección, desde niveles bajos hasta niveles altos. Este término fue usado para describir la evolución, y puede ser usado para describir el entendimiento.

La evaluación puede ser considerada como “entendimiento de la naturaleza” porque los fenómenos son muy similares en varios niveles. La evolución de las especies puede generar especies avanzadas a partir de especies más simples de la misma manera que el entendimiento es el descubrimiento y la reutilización de conceptos de alto nivel en insumos de bajo nivel.

En contraste, el razonamiento procede dividiendo los problemas en subproblemas y resolviéndolos, que es un tipo de estrategia que “fluye cuesta abajo”. En matemáticas aceptamos que necesitamos usar la inducción para hacer “cuesta arriba” en abstracciones. Y eso es un movimiento cuesta arriba muy limitado. La epistemología permite movimientos cuesta arriba mucho más fuertes. Esto se conoce como “saltar a conclusiones sobre evidencia escasa” y está permitido en los sistemas precientíficos basados en epistemología.

La naturaleza/evolución reutiliza todo lo que funciona. Me gusta pensar que el Entendimiento es un fantasma de la Evolución misma. El darwinismo neural ciertamente se extiende a esta brecha. Podría ser una coincidencia, o la única respuesta que funcionará en absoluto.

En este sentido, los últimos años el dominio del Deep Learning sobre IA y Aprendizaje de máquina ha sido rotundo, a tal punto que puede ser considerada una

técnica de IA fuerte capaz de resolver desafíos altamente complejos; sin embargo, es indudable que conlleva varias limitaciones e incluso aplicaciones en las que puede ser inviable o incluso peligroso.

En este marco, no nos enfocaremos en las limitaciones del aprendizaje profundo a un nivel filosófico (por cuanto nos encontramos definiendo un marco filosófico) sin entrar a las profundidades técnicas que demuestran dichas limitaciones.

Una de las primeras limitaciones es que, aunque Deep Learning indica “profundo”, es un algoritmo relativamente “superficial” desde el punto de vista de la “inteligencia general”. Cualquier perturbación del problema original puede invalidar por completo todo el entrenamiento. Este concepto es denominado “transferencia de conocimiento”, es decir, la capacidad de aplicar el conocimiento adquirido previamente a otro problema similar o relacionado; es en esto que el aprendizaje profundo es limitado en la práctica.

La segunda debilidad es probablemente una consecuencia de la primera. El aprendizaje profundo no es completamente transparente. Esto significa que podemos verificar fácilmente si la respuesta es correcta o no, pero al mismo tiempo no tenemos oportunidad de saber por qué el sistema tomó una decisión determinada. Si bien esto no es un problema en muchos casos (no nos importa el complejo razonamiento por el cual Google Photo puede identificar que tenemos un gato en una foto, al menos para el usuario final). Sin embargo, esto puede ser un gran problema para otras aplicaciones.

Si la aplicación es muy compleja, la transparencia puede ser vital para propósitos de depuración. Si estamos diseñando un software de diagnóstico médico, la transparencia es muy importante para permitir que otros médicos (u otro software) entiendan por qué se ha hecho un determinado diagnóstico. O porque, como hemos visto anteriormente, un sistema de aprendizaje automático no transparente puede llevar a un sesgo social grave (¿cómo podemos saber si se ha tomado una decisión determinada por razones legítimas o porque los datos de entrada contienen algún sesgo sutil?).

El problema aquí es que los sistemas de aprendizaje profundo están aprendien-

do un modelo del mundo que no conocemos y que no pueden explicarnos. En algunos casos, esto es una ventaja. Imaginar procesar conscientemente las señales oculares sería un desastre. Gastaríamos mucho cerebro para entender los objetos que tenemos a nuestro alrededor. El hecho de que todo este cálculo se realice automáticamente por las redes neuronales en nuestro cerebro es perfecto. Al mismo tiempo, este no es el mejor enfoque para todos los problemas posibles.

Este no es solo un problema práctico, es un problema filosófico. Podemos resolver problemas usando Deep Learning y, al mismo tiempo, todavía no podemos entenderlos. Esta situación no tiene precedentes a esta escala en la evolución del conocimiento humano.

Esto indica que los sistemas de aprendizaje profundo pueden resolver problemas, pero no “entienden” el problema. Aquí podemos plantear la primera pregunta “profunda” sobre la inteligencia: ¿Cuál es la naturaleza de la “comprensión”?

Hasta este punto, hemos reflexionado respecto a cómo, por qué y cuáles son los límites de la IA, lo que es el sentido de la filosofía que no busca respuestas, sino más bien plantear preguntas.

CAPÍTULO III

METODOLOGÍA DE LA INVESTIGACIÓN

3.1 TIPO Y DISEÑO DE LA INVESTIGACIÓN

La investigación cumple dos propósitos fundamentales: *a*) producir conocimiento y teorías (investigación básica) y *b*) resolver problemas (investigación aplicada) (Hernández Sampieri et al., 2018, p. xxvii). La presente investigación busca resolver el problema de la clasificación de enfermedades en hojas de olivo y orégano, por lo tanto, se caracteriza como aplicada.

Las investigaciones experimentales son aquellas que administran estímulos o tratamientos y/o intervenciones y se caracterizan por la:

- Manipulación intencional de variables (independientes).
- Medición de variables (dependientes).
- Control y validez.
- Dos o más grupos de comparación.
- Participantes asignados al azar o emparejados.

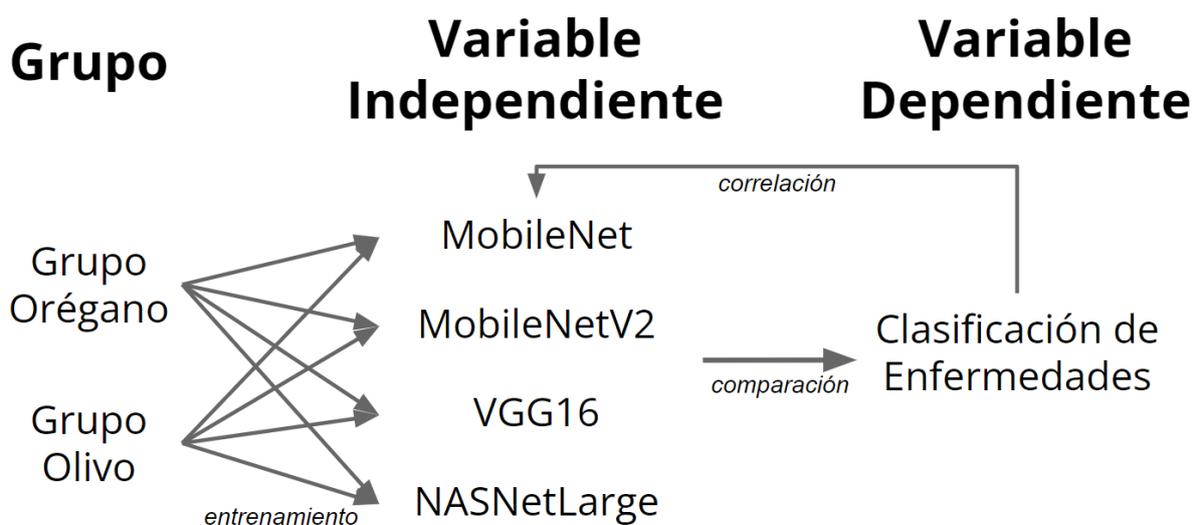
El diseño de una investigación se refiere al plan o estrategia concebida para obtener la información que se desea en el que se debe visualizar la manera práctica y concreta de responder a las preguntas de la investigación y cubrir los objetivos fijados (Hernández Sampieri et al., 2018, p. 120).

En la presente investigación, se manipula la variable independiente que son los modelos de redes neuronales convolucionales (entrenan) y se mide de estos su rendimiento en la clasificación de enfermedades de olivo y orégano (medición de variables). El entrenamiento de los modelos se lleva a cabo usando el conjunto de entrenamiento con las muestras aleatorizadas (asignados al azar) y la medición usando el conjunto de validación (validez), por lo tanto, se caracteriza como un diseño de investigación de tipo experimental y de experimentos puros (Hernández Sampieri et al., 2018, p. 119).

Finalmente, se presenta el resumen del diseño experimental en la Figura 8.

Figura 8

Diseño experimental



Nota: Elaboración propia

Dado el diseño de investigación, podemos caracterizar el alcance del estudio. Tomando en cuenta que una investigación puede incluir diferentes alcances (Hernández Sampieri et al., 2018, p. 84), la presente investigación se caracteriza como descriptiva y correlacional.

3.2 POBLACIÓN Y MUESTRA DE ESTUDIO

3.2.1 Población

La población está definida como el conjunto de todos los casos que concuerdan con una serie de especificaciones (delimitación) (Hernández Sampieri et al., 2018, p. 120).

En el presente estudio se considera como unidad de análisis las hojas de olivo u orégano región Tacna, sin embargo, es muy difícil cuantificar directamente con cuantas de ellas existen debido a: volumen y extensión de los cultivos, variabilidad en la población, ciclo de vida, inviabilidad logística, etc. Sin embargo, el no poder cuantificar la cantidad de hojas que son la población no afecta negativamente la validez y confiabilidad de estudio debido a que se ha delimitado correctamente la población.

La población está constituida por las hojas olivo y orégano que cuenten con las siguientes enfermedades:

Olivo

- **Virosis:** El nombre es descriptivo de esta enfermedad del follaje del olivo debido a la forma curvada o de hoz que muestra la hoja afectada por esta enfermedad. La enfermedad es común en todas las variedades comerciales de aceitunas cultivadas en Sudamérica (Alfaro et al., 2003).
- **Fumagina:** Es una especie de hongo, de color negruzco, que recubre los tejidos vegetales en forma de una capa de hollín, que dificulta la fotosíntesis por lo que los olivos afectados ven disminuida su capacidad productiva. Esta capa se desprende al pasar el dedo por las partes afectadas (Molina de la Rosa et al., 2017).
- **Deficiencia nutricional:** Hojas con coloraciones anormales en el extremo terminal de las hojas como fósforo y potasio (Fernández-Escobar et al., 2016).

Orégano

- **Alternaria:** Alternaria es un género de hongos, son patógenos que afectan a las plantas, caracterizados por la formación de feodiceosporas de color oscuro. Suelen aparecer en forma de anillos o manchas oscuras y concéntricas en las hojas. Estas manchas pueden variar en tamaño y color, pasando del marrón al negro (Woudenberg et al., 2013).
- **Mosaico:** El virus del mosaico de la alfalfa (AMV) está clasificado como un virus ARN de la familia Bromoviridae, que causa patrones de decoloración tipo mosaico en las hojas de las plantas infectadas. Estos patrones suelen consistir en zonas verdes claras y oscuras, parecidas a un mosaico. Las hojas también pueden presentar curvatura, distorsión y crecimiento reducido (Murphy et al., 1995).
- **Oidiosis:** Principal enfermedad del orégano, el síntoma principal es la presencia de polvo blanco en las hojas, primero puntos, luego manchas grandes y finalmente amarilleamiento y secado de las hojas (Huacollo Alvarez, 2014).

3.2.2 Muestra

Se utilizaron 02 conjuntos de datos del proyecto de investigación “Visión Computacional para la Detección y Diagnóstico de enfermedades en Cultivos Relevantes de la Región Tacna” de la Universidad Nacional Jorge Basadre Grohmann que se describen a continuación:

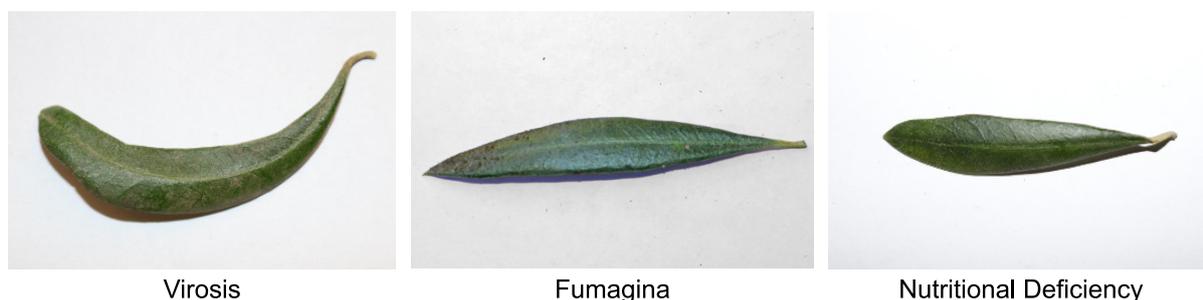
Constan de imágenes RGB que se capturaron con una cámara Canon EOS Rebel T6i de 24,2 MPX, las imágenes poseen unas dimensiones de 3984 x 2656 píxeles. Los ajustes de la cámara se mantuvieron constantes, con una distancia focal de 55 mm, una apertura de 1/200 s y una sensibilidad ISO-400. La fecha de captura fue agosto de 2019 (olivo) y septiembre de 2018 (orégano).

Un método empírico para diferenciar las enfermedades que afectan a las hojas

del orégano consiste en analizar la extensión de la coloración marrón, amarilla y blanca. No obstante, este criterio sólo es aplicable en ocasiones y a menudo requiere los conocimientos de un especialista en agricultura. En la Figura 9 se ilustran ejemplos de cada enfermedad.

Figura 9

Muestras de Olivo



Nota: Elaboración propia

El procedimiento para la construcción del conjunto de datos de enfermedades de la hoja del olivo incluye tres fases. (i) En primer lugar, se capturan imágenes de olivos, seleccionando tres enfermedades según el estudio. Las imágenes de hojas de olivo pertenecen a cultivos de La Yarada - Los Palos en la región Tacna de Perú y fueron recogidas en agosto de 2019. (ii) En segundo lugar, una vez recogidas las imágenes, se entregan a un agrónomo especialista experto en imágenes de enfermedades del olivo. (iii) Finalmente, el especialista filtró las imágenes y las etiquetó manualmente según sus características.

El total de imágenes adquiridas dio como resultado 773 imágenes útiles, de las cuales 258 corresponden a enfermedades por carencias nutricionales, 257 a fumagina y 258 a virosis. Con el fin de tener un equilibrio de clases, se realizó una división 77:19:3 para formar los conjuntos de entrenamiento, validación y prueba. En la Tabla 2 se especifica el número final de imágenes por clase y conjunto de datos.

Tabla 2*Conjunto de datos de enfermedades de olivo*

Clase	Conjunto de entrenamiento	Conjunto de validación	Total
Deficiencias	200	50	258
Fumagina	200	50	257
Virosis	200	50	258
Total	600	150	773

Una forma de distinguir las hojas de orégano enfermas es fijarse en el nivel de color verde, marrón y amarillo. Sin embargo, se trata de un criterio que no se aplica a todos los casos, por lo que suele ser necesario recurrir a un experto agrónomo. En la Fig. 10 se muestran muestras de las imágenes tomadas.

Figura 10*Muestras de orégano***Nota:** Elaboración propia

El procedimiento para la construcción del conjunto de datos de enfermedades de las hojas de orégano incluyó cuatro fases: (i) En primer lugar, se capturaron imágenes de plantas de orégano que presentaban las tres enfermedades mencionadas. Las imágenes de las hojas de orégano fueron recolectadas en septiembre de 2018 en Susapaya - Tarata en Tacna, Perú (ii) Segundo, tras la captura de las imágenes, se

recortaron las imágenes en hojas considerando como criterios de calidad que las imágenes estuvieran enfocadas y que el recorte no fuera inferior a 112x112 píxeles. (iii) En tercer lugar, una vez procesadas las imágenes, se entregaron a un agrónomo experto en enfermedades del orégano. (iv) Finalmente, el especialista procedió a seleccionar las imágenes y asignarles etiquetas en función de sus características distintivas de forma manual.

El total adquirido dio como resultado 1642 imágenes útiles, de las cuales 314 pertenecen a la clase Alternaria, 811 a la clase Mosaico y 512 a la clase enfermedad del Oidio. Para construir los conjuntos de entrenamiento y validación, se realizó una división aleatoria 80:20. La Tabla 3 muestra la composición del conjunto de datos.

Tabla 3

Conjunto de datos de enfermedades de orégano

Clase	Conjunto de entrenamiento	Conjunto de validación	Total
Alternaria	260	54	314
Mosaico	627	184	811
Oidio	426	91	517
Total	1,313	329	1,642

3.3 ACCIONES Y ACTIVIDADES PARA LA EJECUCIÓN DEL PROYECTO

Efectuada la operacionalización de las variables y definidos los instrumentos de medición, las acciones y actividades para llevar a cabo la ejecución de la investigación se encuentran enmarcadas a cumplir los objetivos específicos en estricto orden de prelación.

La validez de los instrumentos se encuentra enmarcada en los instrumentos utilizados en las investigaciones mencionadas en los antecedentes del estudio.

3.4 MATERIALES E INSTRUMENTOS

En el contexto de la metodología de investigación para tesis, los autores del libro Metodología de la investigación (Hernández Sampieri et al., 2018), definen los instrumentos y técnicas de investigación como los recursos que se utilizan para recolectar datos:

Los **instrumentos** son los medios físicos o materiales que se utilizan para recolectar los datos. Por ejemplo, un cuestionario, una entrevista, un test o un registro observacional. Las **técnicas** son los procedimientos o métodos que se utilizan para recolectar los datos. Por ejemplo, la observación directa, la encuesta, la entrevista o el análisis documental.

Adicionalmente, también se define la técnica de investigación como el procedimiento o forma particular de obtener datos necesarios de la investigación, y al instrumento como cualquier dispositivo o formato (en papel o digital) que se utiliza para obtener, registrar o almacenar información (Arias, 2012, p. 67-68).

Sin embargo, es importante tener en cuenta que los autores mencionados se centran en los métodos de investigación tradicionales, como el método experimental o el método descriptivo. El machine learning (aprendizaje de máquina o aprendizaje automático) es un método de investigación relativamente nuevo, y no está explícitamente cubierto en el libro. En el caso de inteligencia artificial, se podría argumentar que el instrumento son el software y hardware que utiliza el investigador y la técnica los algoritmos y métodos propios de la ciencia computacional que aplica.

En tal sentido, en la presente investigación se emplearon las siguientes técnicas e instrumentos:

- **Técnicas:** Machine Learning (Aprendizaje de Máquina): Descenso por Gradiente, redes neuronales convolucionales, aprendizaje por transferencia, aumento de datos, ajuste fino.

- **Instrumento:** Para el diseño de los experimentos y su ejecución, se tuvo a disposición el siguiente hardware informático a través de una suscripción a Google Colab Pro: GPU 1x NVIDIA Tesla T4 con 15GB de RAM y CPU 1x Procesadores Xeon @2,3Ghz con 12,7 GB de RAM. En este estudio se utilizó Keras, una de las principales API de aprendizaje profundo escrita enPython. Es compatible con múltiples motores de computación de redes neuronales y se ejecuta en la plataforma de aprendizaje automático TensorFlow. Las versiones fueron: Keras 2.11.0 y Tensorflow 2.11.0 como backend en un entorno Pythonversión 3.8.

Las técnicas e instrumentos mencionadas son las mismas empleadas y utilizadas en estudios similares abordados en los antecedentes de la presente investigación.

3.5 TRATAMIENTO DE DATOS

3.5.1 Procedimiento de recolección de datos

Los datos a recolectarse consisten en las métricas (indicadores) de la variable dependiente, sin embargo, antes de recolectar estos datos fue necesario llevar a cabo la experimentación que precisamos a continuación:

Para la preparación de los datos, las imágenes RGB del conjunto de datos se redimensionaron al tamaño de entrada de la arquitectura seleccionada (224,224) y cada muestra se escaló a media 0. Además, las etiquetas de cada una de las imágenes se categorizaron como una variable dummy para que coincidan con el tamaño de salida de la arquitectura modificada. Esto se realizó en todos los experimentos.

Para el aumento de datos, que se utilizó en todos los experimentos, las imágenes enviadas al modelo para el entrenamiento se incrementaron artificialmente mediante las siguientes transformaciones aleatorias:

- 0-45 grados de rotación aleatoria.

- 0,8-1,2 zoom.
- 0-0,2 de desplazamiento de ancho total.
- 0-0,2 de desplazamiento de altura total.
- Volteo horizontal.
- Voltar verticalmente.

Se describe esta configuración como “Aumento agresivo de datos” porque consigue una transformación que rota la imagen de 0° a 359° con una variedad de zooms, desplazamientos y giros.

Para el presente estudio, se utilizaron los siguientes criterios para probar el rendimiento de diversas arquitecturas de redes neuronales convolucionales:

- Con el menor peso (en bytes).
- Con la mayor exactitud Top-1 en ImageNet.
- Con la mayor exactitud Top-5 en ImageNet.
- Con el mayor número de parámetros.
- Con el mayor número de capas.
- Con el menor tiempo de inferencia en GPU.

Los modelos seleccionados, según las métricas mencionadas, se resumen en la Tabla 4 (Team, 2023). Se redujo el número de arquitecturas a probar a 4 porque algunas cumplían múltiples criterios de selección: VGG16, MobileNet, MobileNetV2 y NASNetLarge.

Tabla 4*Métricas de los modelos en ImageNet*

Modelo	MobileNetV2	MobileNet	VGG16	NASNetLarge
Size (MB)	14	16	528	343
Top-1 Exactitud	0,713	0,704	0,713	0,825
Top-5 Exactitud	0,901	0,895	0,901	0,96
Parámetros	3,5M	4,3M	138,4M	88,9M
Profundidad	105	55	16	533
Tiempo (ms) de inferencia	3,8	3,4	4,2	20

Las etapas de aprendizaje por transferencia y ajuste fino se aplicaron para entrenar los modelos utilizando los modelos base de ImageNet pre-entrenados disponibles en Keras (Team, 2023) para obtener los mejores resultados.

Finalmente, para el entrenamiento de los modelos, la pérdida fue Categorical Cross Entropy, y el optimizador fue ADAM (Kingma and Ba, 2014), excepto para la etapa Ajuste Fino, donde el optimizador fue RMSProp (Hinton et al., 2012) con una tasa de aprendizaje de 0,00001.

Cada experimento consistirá en entrenar, mediante aprendizaje por transferencia y ajuste fino, un modelo de red neuronal convolucional pre-entrenado en ImageNet al que se le le modificará la arquitectura.

Para obtener resultados más fiables, cada experimento consta de 10 repeticiones.

En cada experimento, la etapa de aprendizaje por transferencia se ejecutó durante 100 épocas. A continuación, se seleccionó la mejor época de cada repetición de cada experimento para la fase de ajuste fino y se entrenó durante otras 100 épocas. De nuevo, la mejor época de cada repetición de cada experimento se almacenó para la selección del modelo en función de la exactitud en el conjunto de validación.

Todos los experimentos deberán pasar por las fases de infraajuste y sobreajuste del entrenamiento del modelo, asegurando que en ninguno de los casos el número de

épocas de entrenamiento fue insuficiente.

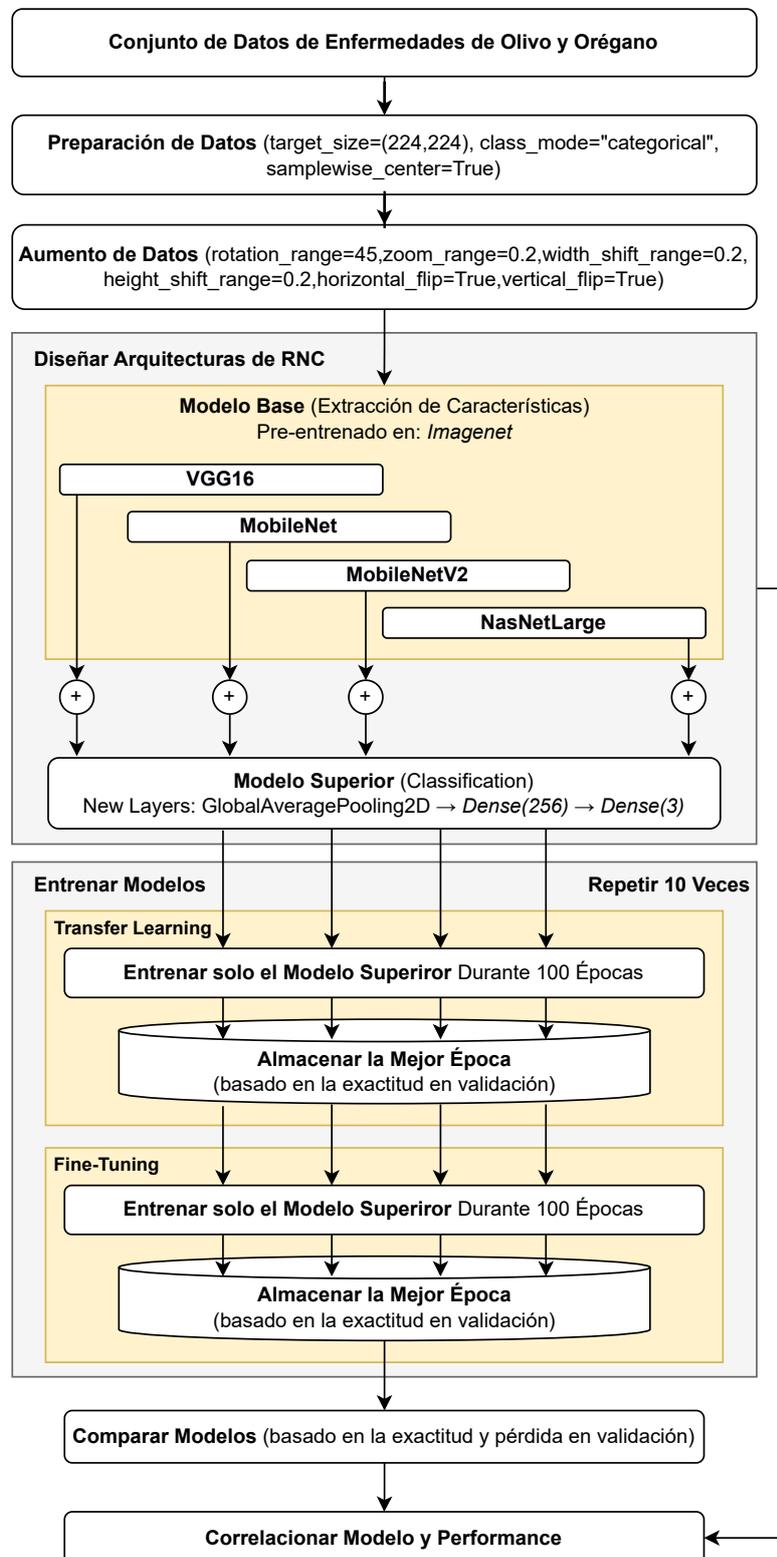
Los experimentos VGG16, MobileNet, MobileNetV2 y NASNetLarge requirieron aproximadamente $\approx 1,41, 1,40, 1,44$ y $3,31$ horas por cada repetición de entrenamiento. Para diez repeticiones, el tiempo total de entrenamiento fue de $\approx 85,41$ horas.

3.5.2 Análisis y procesamiento de datos

A continuación, se presenta el flujo de trabajo o pipeline de metodología aplicada:

Figura 11

Flujo de trabajo experimental



Nota: Elaboración propia

CAPÍTULO IV

RESULTADOS DE LA INVESTIGACIÓN

4.1 DISEÑO DE ARQUITECTURAS DE RED NEURONAL CONVOLUCIONAL

La arquitectura de cada modelo (VGG16, MobileNet, MobileNetV2, NASNetLarge) se modificará conectando el modelo base (que extrae características) a una capa 2D de agrupación de promedios globales para reducir la dimensionalidad de las características extraídas. A continuación, se añadirá secuencialmente una capa densa de 256 neuronas, una función de activación ReLU y una capa densa de 3 neuronas para mapear todo el modelo a las 3 enfermedades.

La Tabla 5 presenta los parámetros entrenables y no entrenables de las arquitecturas modificadas en las etapas de Aprendizaje por Transferencia (TL) y Ajuste Fino (FT).

Tabla 5*Métricas de las arquitecturas modificadas*

Arquitectura (Etapa)	Número de Capas	Total de Parámetros	Parámetros Entrenables	Parámetros No Entrenables
MobileNetV2 (TL)	157	2,586,691	328,707	2,257,984
MobileNetV2 (TL + FT)	157	2,586,691	2,552,579	34,112
MobileNet (TL)	89	3,492,035	263,171	3,228,864
MobileNet (TL + FT)	89	3,492,035	3,470,147	21,888
VGG16 (TL)	22	14,846,787	132,099	14,714,688
VGG16 (TL + FT)	22	14,846,787	14,846,787	0
NASNetLarge (TL)	1042	85,950,037	1,033,219	84,916,818
NASNetLarge (TL + FT)	1042	85,950,037	85,753,369	196,668

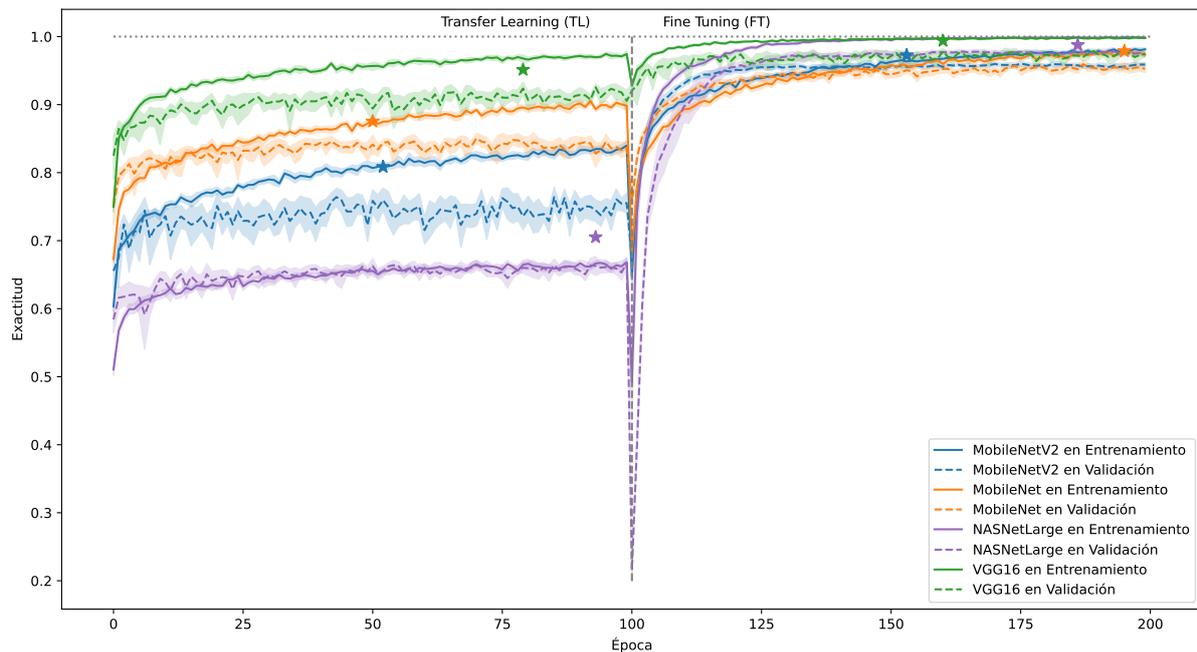
Hemos comprobado experimentalmente que un mapeo directo de las características extraídas por el modelo base a las clases (sin el uso de la capa densa de 256 neuronas + ReLU) no es suficiente para obtener buenos resultados.

4.2 CLASIFICACIÓN DE ENFERMEDADES DE ORÉGANO

4.2.1 Entrenamiento: Etapa de aprendizaje por transferencia

Figura 12

Orégano: Exactitud de los experimentos



Nota: Elaboración propia

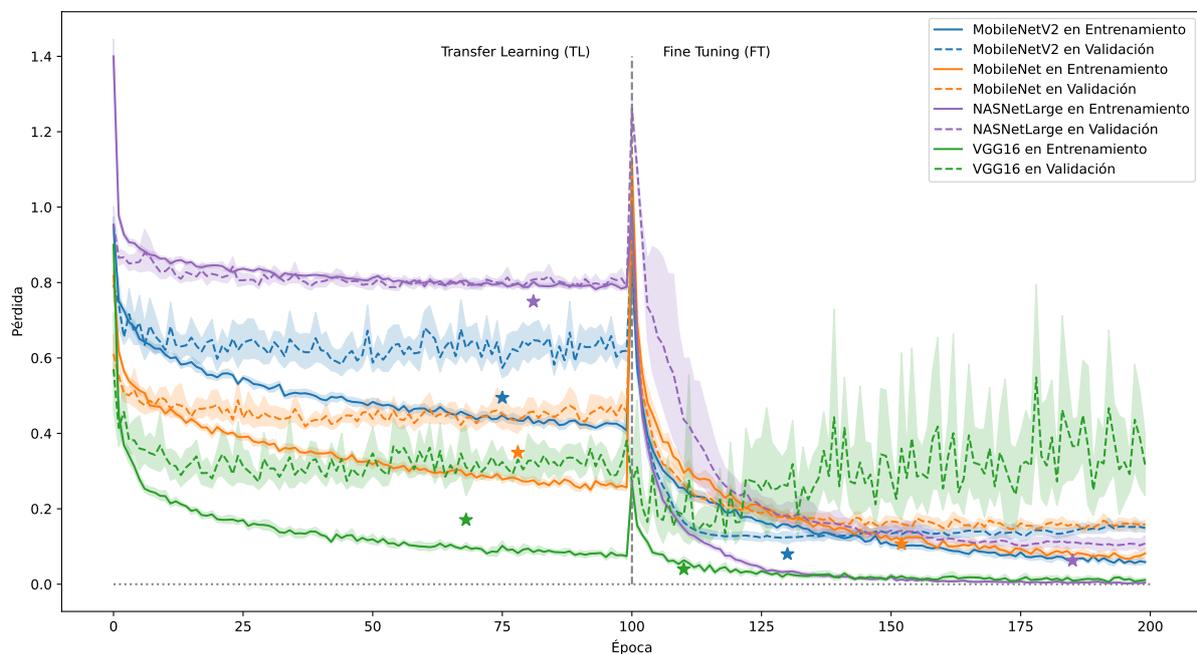
La Figura 12 muestra la exactitud media en los conjuntos de entrenamiento y validación con un rango intercuartílico. Las 100 primeras épocas representan la etapa de aprendizaje por transferencia. Además, las estrellas del gráfico muestran el valor de exactitud más alto obtenido en el conjunto de validación de cada experimento.

La mayor exactitud en el conjunto de validación obtenida en la etapa de aprendizaje por transferencia utilizando el modelo VGG16 fue de 0,951368 en la época 79, utilizando MobileNetV2 fue de 0,87538 en la época 50, utilizando MobileNetV2 fue de 0,808511 en la época 52, y utilizando NASnetLarge fue de 0,705167 en la época 93.

Se puede observar que el modelo VGG16 obtuvo la mejor exactitud del conjunto de validación en la etapa de aprendizaje por transferencia. Además, podemos observar un pobre rendimiento de NASNetLarge en la etapa de aprendizaje por transferencia, en la que incluso el rendimiento para el conjunto de entrenamiento y validación fue similar.

Figura 13

Orégano: Pérdida de los experimentos



Nota: Elaboración propia

La Figura 13 muestra la pérdida media en los conjuntos de entrenamiento y validación con un rango intercuartílico. Las 100 primeras épocas representan la etapa de aprendizaje por transferencia. Además, las estrellas del gráfico muestran el valor de pérdida más bajo obtenido en el conjunto de validación de cada experimento.

La pérdida del conjunto de validación más baja obtenida en la etapa de aprendizaje por transferencia utilizando el modelo VGG16 fue de 0,170563 en la época 68, utilizando MobileNet fue de 0,349056 en la época 78, utilizando MobileNetV2 fue de

0,494586 en la época 75, y utilizando NASnetLarge fue de 0,74966 en la época 81.

Se puede observar que el modelo VGG16 tuvo la mejor pérdida en el conjunto de validación en la etapa de aprendizaje por transferencia. Además, podemos observar un rendimiento pobre de NASnetLarge en la etapa de aprendizaje por transferencia, en la que incluso el rendimiento en el conjunto de entrenamiento fue peor que en el conjunto de validación.

Por último, se observan rendimientos diferenciados, con VGG16 en el primer puesto de los mejores modelos, MobileNet en segundo lugar, MobileNetV2 en tercer lugar, y NASnetLarge en cuarto y último lugar.

4.2.2 Entrenamiento: Etapa de ajuste fino

La Figura 12 muestra, a la derecha de la línea de puntos, las 100 épocas que representan la etapa de ajuste fino. Además, las estrellas del gráfico muestran el valor de exactitud más alto obtenido en el conjunto de validación de cada experimento.

La mayor exactitud en el conjunto de validación obtenida en la etapa de ajuste fino utilizando el modelo VGG16 fue de 0,993921 en la época 160, utilizando NASnetLarge fue de 0,987842 en la época 186, utilizando MobileNet fue de 0,978723 en la época 195, y utilizando MobileNetV2 fue de 0,972644 en la época 153.

Se puede observar que el modelo VGG16 obtuvo la mejor exactitud del conjunto de validación en la fase de ajuste fino. Además, podemos observar un mejor rendimiento de NASnetLarge en la fase de ajuste fino que en la de aprendizaje por transferencia.

En la Figura 13 se muestran las 100 épocas que representan la etapa de Ajuste Fino a la derecha de la línea de puntos. Además, las estrellas del gráfico muestran el valor de pérdida más bajo obtenido en el conjunto de validación de cada experimento.

La pérdida más baja en el conjunto de validación obtenida en la etapa de Ajuste fino utilizando el modelo VGG16 fue de 0,039905 en la época 110, utilizando NASnetLarge fue de 0,062677 en la época 185, utilizando MobileNetV2 fue de 0,07996

en la época 130, y utilizando MobileNet fue de 0,106604 en la época 152.

Se puede observar que el modelo VGG16 tuvo la mejor pérdida en el conjunto de validación en la fase de ajuste fino. Además, podemos observar un mejor rendimiento de NASnetLarge en la etapa Ajuste Fino que en Aprendizaje por Transferencia.

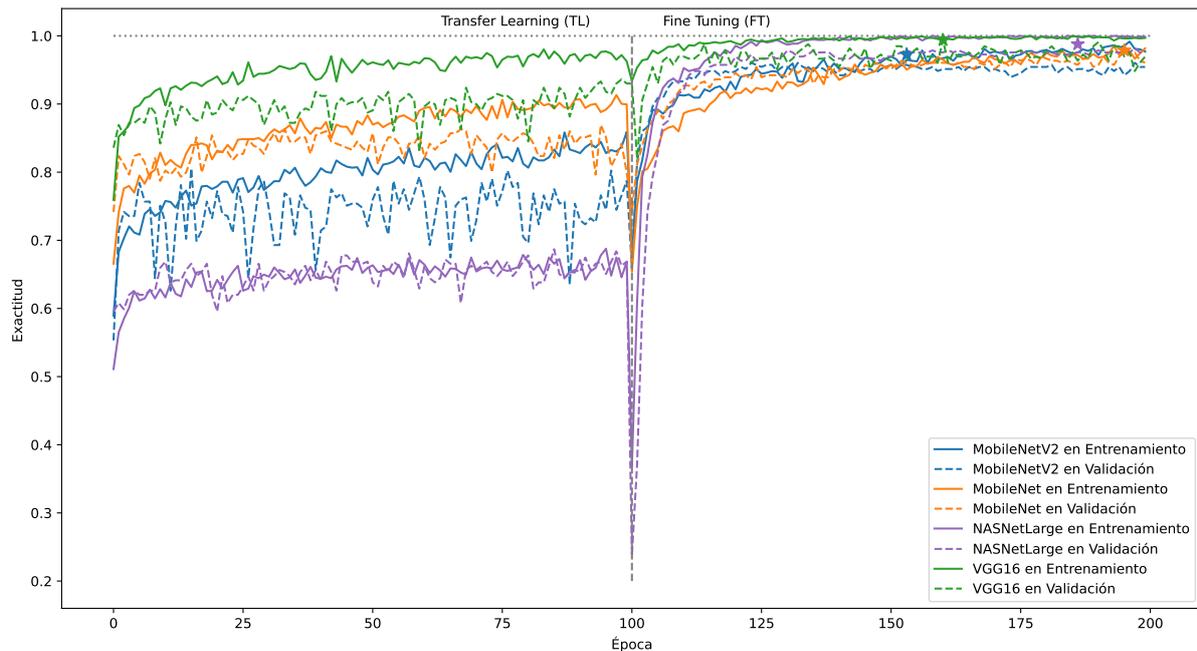
Además, se ha demostrado que la aplicación del ajuste fino ayuda a mejorar la exactitud del conjunto de validación y los valores de pérdida de nuestros modelos. Aplicando solo Aprendizaje por Transferencia, los modelos mostraron una exactitud en validación de 0,7804574477999999 en promedio. Después, aplicando Ajuste Fino, los modelos mostraron una exactitud en validación de 0,94810562183725 en promedio representando una mejora de aproximadamente el 21,48% en la exactitud del conjunto de validación cuando se aplica Ajuste Fino.

Por último, se observan rendimientos diferenciados, con VGG16 en primer lugar de los mejores modelos, NASNetLarge en segundo lugar, MobileNet en tercer lugar, y MobileNetV2 en cuarto y último lugar.

4.2.3 Selección de modelo

Figura 14

Orégano: Exactitud del mejor experimento

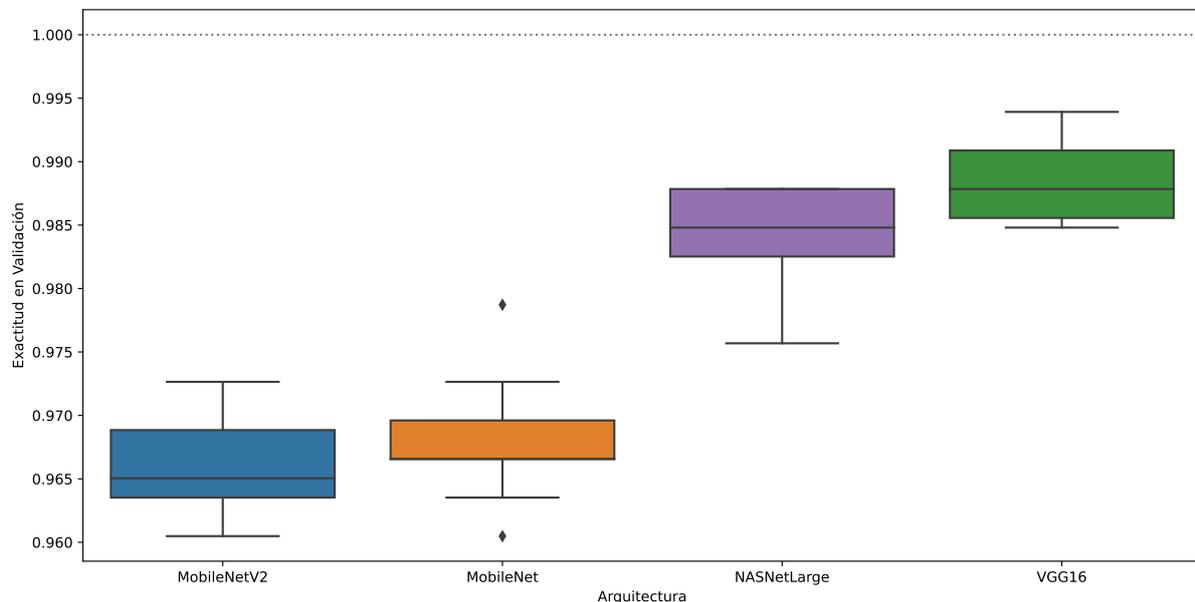


Nota: Elaboración propia

El criterio para seleccionar el modelo fue el modelo de la mejor época basado en la mayor exactitud de validación. La Figura 14 presenta la mejor repetición de todos los experimentos. Los valores de exactitud de validación fueron muy próximos entre los modelos. Por lo tanto, se seleccionó el modelo VGG16 de la repetición 0, época 160, con una pérdida en el conjunto de entrenamiento de 0,000000945, una exactitud en el conjunto de entrenamiento de 1,0, una pérdida en el conjunto de validación de 0,039905 y una exactitud en el conjunto de validación de 0,993921 obtenidos mediante aprendizaje por transferencia y ajuste fino.

Figura 15

Orégano: Diagrama de caja de la exactitud en validación



Nota: Elaboración propia

Además, la Figura 15 muestra la mayor exactitud de validación de todos los experimentos a lo largo de las diez repeticiones, mostrando la superioridad del modelo VGG16 sobre el resto de modelos, encontrando en segundo lugar a NASNetLarge, cuyo Q1 tiene aproximadamente los mismos valores que el Q3+Q4 de VGG16. Muy por debajo, con rendimientos similares, se encuentran MobileNet y MobileNetV2. Esto último valida el modelo seleccionado previamente establecido.

Aplicando la prueba estadística de U de Mann-Whitney presentada en la Tabla 6, encontramos que no hay evidencia de los valores de la distribución de la exactitud obtenida en el conjunto de datos de validación de orégano utilizando el modelo VGG16 sea significativamente mayor que la del modelo NASNetLarge con un p-valor de 0,074.

Tabla 6

Olivo: Prueba de U de Mann-Whitney para mejor rendimiento

Grupo	N	Rango Promedio	Suma de Rangos	U de Mann-Whitney	P valor
VGG16	10	12,4	124,0	69,0	0,074
NASNetLarge	10	8,6	86,0		

La Tabla 7 muestra los valores: mínimo, máximo, mediana, media e IQR (Rango InterCuartílico) de exactitud y pérdida en el conjunto de entrenamiento y validación sobre todos los experimentos tras la etapa de Ajuste Fino.

Los mejores valores de exactitud en el conjunto de entrenamiento corresponden al modelo VGG16 y a NASNetLarge con 1,0. Sin embargo, VGG16 obtiene los mejores valores de pérdida en el conjunto de entrenamiento. VGG16 obtiene los mejores valores de exactitud y pérdida en el conjunto de validación en casi todas las mediciones. Comparando todos los modelos, VGG16 muestra resultados superiores en casi todas las mediciones.

Tabla 7

Orégano: Métricas sobre todos los experimentos

Datos	Métrica	Valor	MobileNetV2	MobileNet	VGG16	NASNetLarge
Entrenamiento	Pérdida	min ↓	0,033839	0,042043	0,000001	0,000333
		max ↓	1,055699	1,284429	0,377679	1,082791
		median ↓	0,110713	0,126563	0,020775	0,013691
		mean ↓	0,145844	0,174924	0,029682	0,061203
		iqr ↓	0,094433	0,110026	0,026243	0,042429
		min ↑	0,593298	0,654227	0,912414	0,351866
	Exactitud	max ↑	0,991622	0,985529	1,000000	1,000000
		median ↑	0,961919	0,955826	0,996192	0,996192
		mean ↑	0,946940	0,939985	0,992790	0,977122
		iqr ↓	0,035796	0,041127	0,005331	0,013709
		min ↓	0,079960	0,106604	0,039905	0,062677
		max ↓	1,334554	1,277737	1,567940	1,558281
Validación	Pérdida	median ↓	0,137759	0,171045	0,233635	0,138150
		mean ↓	0,161784	0,199720	0,291962	0,220457
		iqr ↓	0,035516	0,050521	0,204667	0,097463
		min ↑	0,580547	0,705167	0,820669	0,167173
		max ↑	0,972644	0,978723	0,993921	0,987842
	Exactitud	median ↑	0,954407	0,945289	0,972644	0,972644
		mean ↑	0,945070	0,938112	0,967906	0,941334
		iqr ↓	0,012158	0,021277	0,015198	0,018237

A continuación, el modelo seleccionado se utilizó para generar la Matriz de Confusión que puede verse en la Tabla 8. La Matriz de Confusión nos mostró una clasificación perfecta en el conjunto de entrenamiento, y un pequeño número de errores en el conjunto de validación: una imagen que pertenecía a la clase Alternaria fue clasificada como Mosaico. Además, dos imágenes pertenecientes a la clase Oidio se clasificaron como Alternaria.

Tabla 8*Orégano: Matriz de confusión*

		Conjunto de entrenamiento		
		Clase	Alternaria	Mosaico
Clase verdadera	Alternaria	260	0	0
	Mosaico	0	627	0
	Oidio	0	0	426
	Conjunto de validación			
	Clase	Alternaria	Mosaico	Oidio
	Alternaria	53	1	0
	Mosaico	0	184	0
	Oidio	2	0	86
	Clase predicha			

Las métricas de clasificación típicas, como la exactitud, la precisión, la sensibilidad y la puntuación F1, se tuvieron en cuenta para la evaluación posterior del modelo seleccionado y se muestran en la Tabla 9. La puntuación F1 media de todas las enfermedades fue de 0,9861 en el conjunto de validación. Este valor no coincide con la exactitud del modelo debido al desequilibrio de clases.

Tabla 9*Orégano: Métricas de clasificación por enfermedad*

Conjunto de entrenamiento			
Clase	Precisión	Sensibilidad	F1-Score
Alternaria	1,0000	1,0000	1,0000
Mosaico	1,0000	1,0000	1,0000
Oidio	1,0000	1,0000	1,0000
Conjunto de validación			
Clase	Precisión	Sensibilidad	F1-Score
Alternaria	0,9636	0,9815	0,9725
Mosaico	0,9946	1,0000	0,9973
Oidio	1,0000	0,9780	0,9886

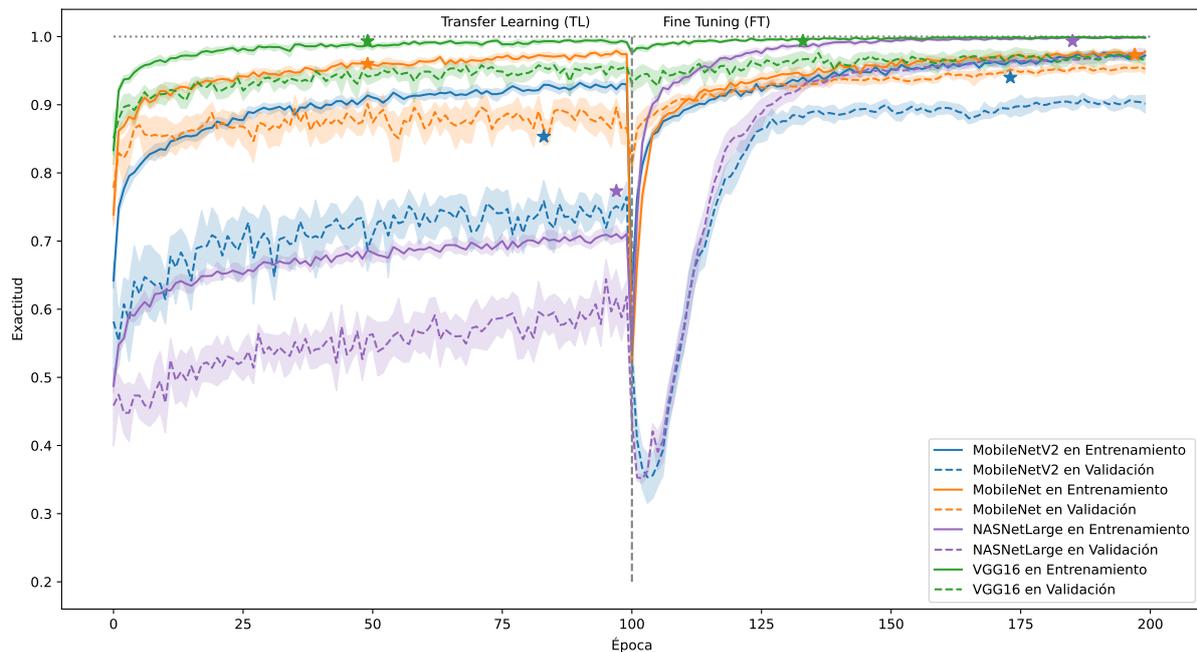
Por último, el modelo seleccionado está disponible en nuestro [GitHub](#) para su uso y evaluación posteriores.

4.3 CLASIFICACIÓN DE ENFERMEDADES DE OLIVO

4.3.1 Entrenamiento: Etapa de aprendizaje por transferencia

Figura 16

Olivo: Exactitud de los experimentos



Nota: Elaboración propia

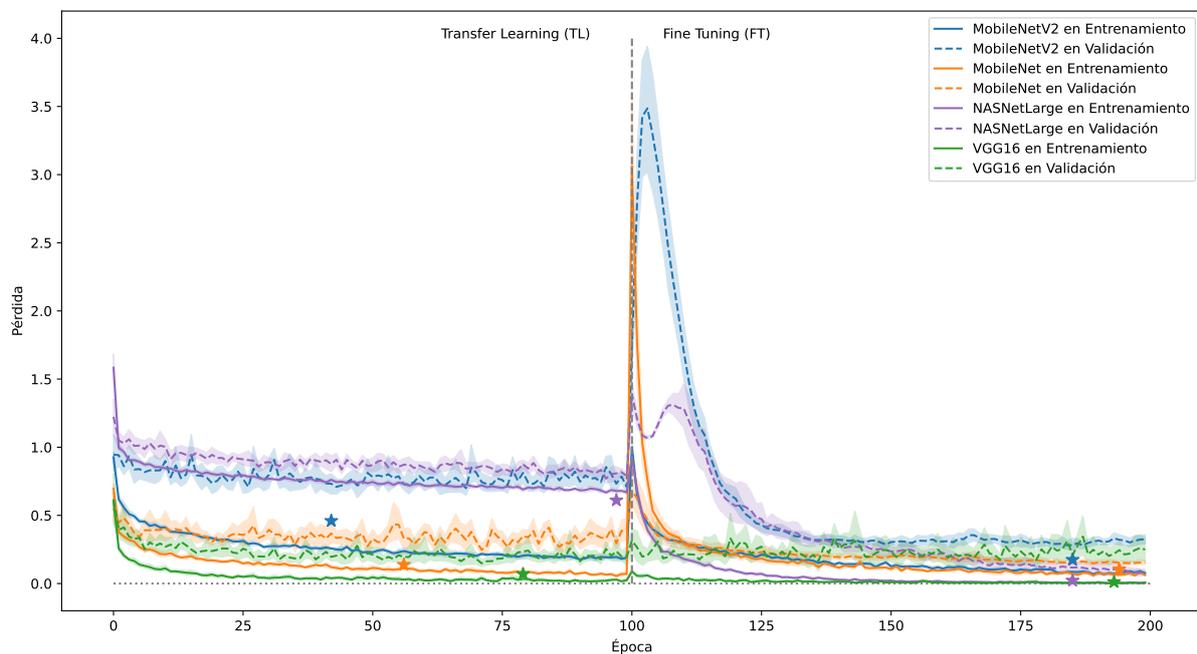
La Figura 16 muestra la exactitud media en los conjuntos de entrenamiento y validación con un rango intercuartílico. Las 100 primeras épocas representan la etapa de aprendizaje por transferencia. Además, las estrellas del gráfico muestran el valor de exactitud más alto obtenido en el conjunto de validación de cada experimento.

La mayor exactitud en el conjunto de validación obtenida en la etapa de Aprendizaje por Transferencia utilizando el modelo VGG16 fue de 0,993333 en la época 49, utilizando MobileNet fue de 0,96 en la época 49, utilizando MobileNetV2 fue de 0,853333 en la época 83, y utilizando NASnetLarge fue de 0,773333 en la época 97.

Se puede observar que el modelo VGG16 obtuvo la mejor exactitud del conjunto de validación en la etapa de aprendizaje por transferencia. Además, podemos observar un pobre rendimiento de NASNetLarge en la etapa de aprendizaje por transferencia, en la que incluso el rendimiento para el conjunto de entrenamiento en NASNetLarge fue menor que el rendimiento en el conjunto de validación de los otros modelos.

Figura 17

Olivo: Pérdida de los experimentos



Nota: Elaboración propia

La Figura 17 muestra la pérdida media en los conjuntos de entrenamiento y validación con un rango intercuartílico. Las 100 primeras épocas representan la etapa de aprendizaje por transferencia. Además, las estrellas del gráfico muestran el valor de pérdida más bajo obtenido en el conjunto de validación de cada experimento.

La pérdida del conjunto de validación más baja obtenida en la etapa de aprendizaje por transferencia utilizando el modelo VGG16 fue de 0,069734 en la época 79, utilizando MobileNet fue de 0,138183 en la época 56, utilizando MobileNetV2 fue de

0,459678 en la época 42, y utilizando NASnetLarge fue de 0,610832 en la época 97.

Se puede observar que el modelo VGG16 tuvo la mejor pérdida en el conjunto de validación en la etapa de aprendizaje por transferencia. Además, podemos observar un rendimiento pobre de NASnetLarge en la etapa de aprendizaje por transferencia, en la que incluso el rendimiento en el conjunto de entrenamiento fue peor que en el conjunto de validación.

Por último, se observan rendimientos diferenciados, con VGG16 en el primer puesto de los mejores modelos, MobileNet en segundo lugar, MobileNetV2 en tercer lugar, y NASNetLarge en cuarto y último lugar.

4.3.2 Entrenamiento: Etapa de ajuste fino

La Figura 16 muestra, a la derecha de la línea de puntos, las 100 épocas que representan la etapa de ajuste fino. Además, las estrellas del gráfico muestran el valor de exactitud más alto obtenido en el conjunto de validación de cada experimento.

La mayor exactitud en el conjunto de validación obtenida en la etapa de ajuste fino utilizando el modelo VGG16 fue de 0,993333 en la época 193, utilizando NASNetLarge fue de 0,993333 en la época 185, utilizando MobileNet fue de 0,966667 en la época 194, y utilizando MobileNetV2 fue de 0,926667 en la época 185.

Se puede observar que los modelos VGG16 y NASNetLarge obtuvo la mejor exactitud del conjunto de validación en la fase de ajuste fino. Además, podemos observar un mucho mejor rendimiento de NASnetLarge en la fase de ajuste fino que en la de aprendizaje por transferencia.

En la Figura 17 se muestran las 100 épocas que representan la etapa de Ajuste Fino a la derecha de la línea de puntos. Además, las estrellas del gráfico muestran el valor de pérdida más bajo obtenido en el conjunto de validación de cada experimento.

La pérdida más baja en el conjunto de validación obtenida en la etapa de Ajuste fino utilizando el modelo VGG16 fue de 0,011384 en la época 193, utilizando

MobileNetV2 fue de 0,174358 en la época 185, utilizando NASnetLarge fue de 0,021931 en la época 185, y utilizando MobileNet fue de 0,104122 en la época 194.

Se puede observar que el modelo VGG16 tuvo la menor pérdida en el conjunto de validación en la fase de ajuste fino. Además, podemos observar un mejor rendimiento de NASnetLarge en la etapa Ajuste Fino que en Aprendizaje por Transferencia.

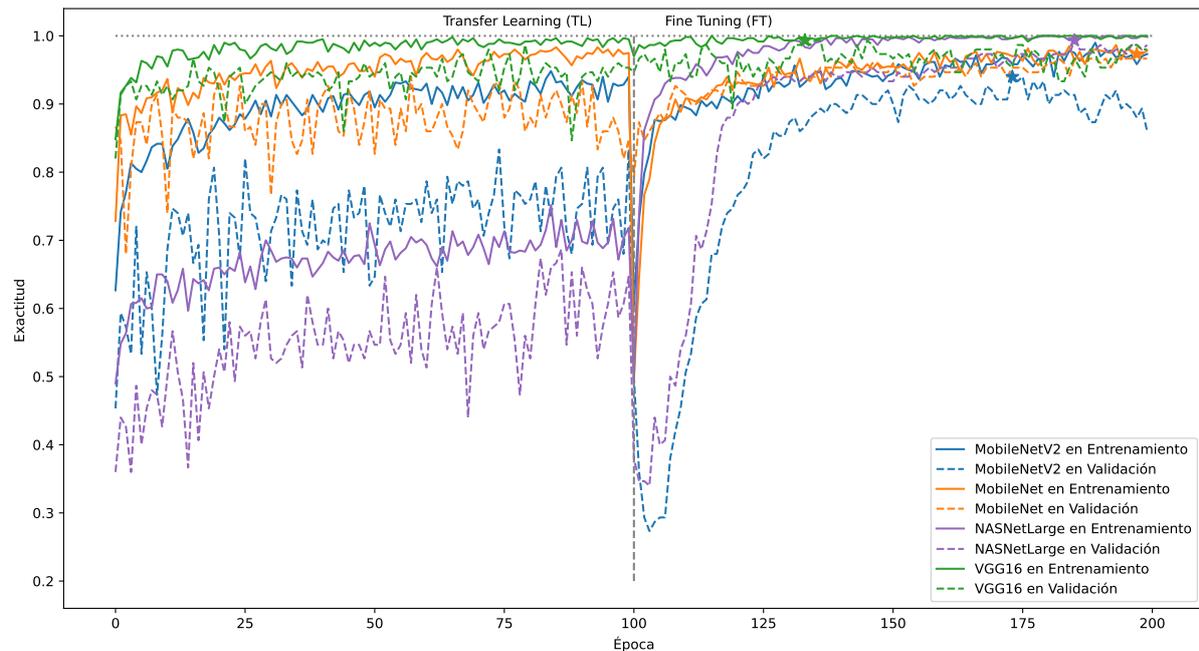
Además, se ha demostrado que la aplicación del ajuste fino ayuda a mejorar la exactitud del conjunto de validación y los valores de pérdida de nuestros modelos. Aplicando solo Aprendizaje por Transferencia, los modelos mostraron una exactitud en validación de 0,766304999472 en promedio. Después, aplicando Ajuste Fino, los modelos mostraron una exactitud en validación de 0,8977949980175 en promedio representando una mejora de aproximadamente el 17,15% en la exactitud del conjunto de validación cuando se aplica Ajuste Fino.

Por último, se observan rendimientos diferenciados, con VGG16 y NASNetLarge en primer lugar de los mejores modelos, MobileNet en tercer lugar y MobileNetV2 en cuarto y último lugar.

4.3.3 Selección de modelo

Figura 18

Olivo: Exactitud del mejor experimento

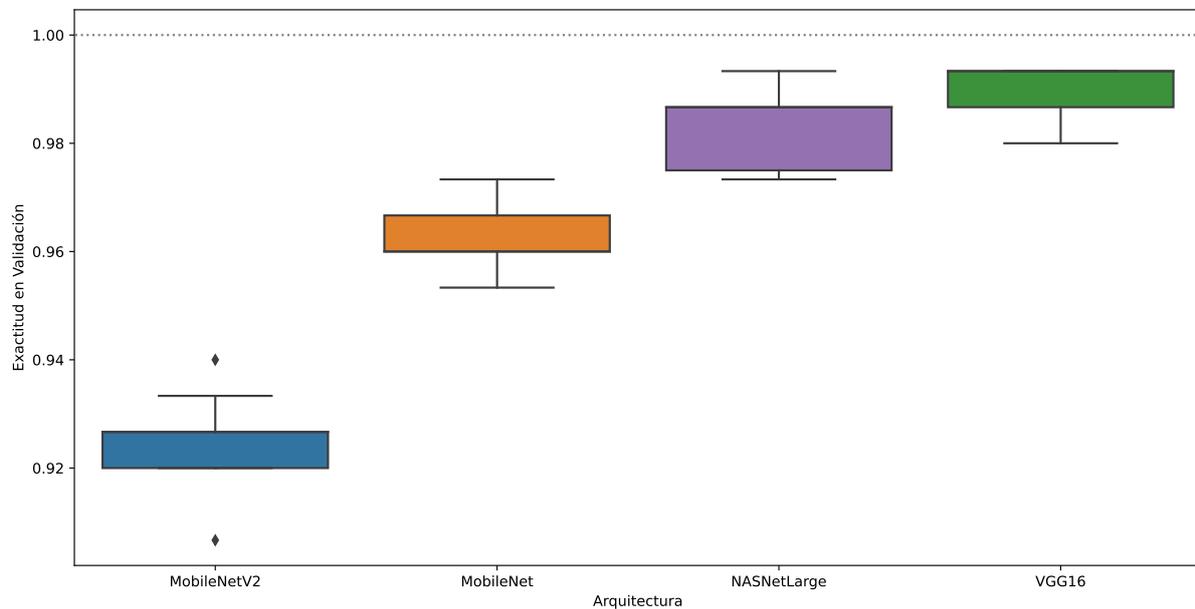


Nota: Elaboración propia

El criterio para seleccionar el modelo fue el modelo de la mejor época basado en la mayor exactitud de validación. La Figura 18 presenta la mejor repetición de todos los experimentos. Dos modelos empataron con la mejor exactitud en validación, procediéndose a elegir al modelo con menor pérdida. Por lo tanto, se seleccionó el modelo VGG16 del experimento 5, época 193, con una pérdida de 0,000029 y exactitud de 1,0 en el conjunto de entrenamiento y una pérdida de 0,011384 y exactitud de 0,993333 en el conjunto de validación, obtenido mediante aprendizaje por transferencia y ajuste fino.

Figura 19

Olivo: Diagrama de caja de la exactitud en validación



Nota: Elaboración propia

Además, la Figura 19 muestra la mayor exactitud de validación de todos los experimentos a lo largo de las diez repeticiones, mostrando la superioridad del modelo VGG16 sobre el resto de modelos, encontrando en segundo lugar a NASNetLarge, cuyo Q1 tiene aproximadamente los mismos valores que el Q1+Q2+Q3 de VGG16. Por debajo, se encuentra el modelo MobileNet cuyo mejor resultado es pertenece al Q4 del modelo NASNetLarge. Finalmente, muy por debajo se encuentra el modelo MobileNetV2.

Aplicando la prueba estadística de U de Mann-Whitney que se presenta en la Tabla 10, encontramos que hay evidencia de que la distribución de la exactitud obtenida en el conjunto de datos de validación de olivo utilizando el modelo VGG16 es significativamente mayor que la del modelo NASNetLarge con un p-valor de 0,074.

Tabla 10

Olivo: Prueba de U de Mann-Whitney para mejor rendimiento

Grupo	N	Rango Promedio	Suma de Rangos	U de Mann-Whitney	P valor
VGG16	10	13,5	135,0	80,0	0,009
NASNetLarge	10	7,5	75,0		

La Tabla 11 muestra los valores: mínimo, máximo, mediana, media e RIQ (Rango InterCuartílico) de exactitud y pérdida en el conjunto de entrenamiento y validación sobre todos los experimentos tras la etapa de Ajuste Fino.

Los mejores valores de exactitud en el conjunto de entrenamiento corresponden al modelo VGG16 y a NASNetLarge con 1,0. Sin embargo, VGG16 obtiene los mejores valores de pérdida en el conjunto de entrenamiento. VGG16 obtiene los mejores valores de exactitud en el conjunto de validación. Comparando todos los modelos, VGG16 muestra resultados superiores en casi todas las mediciones excepto en la pérdida en validación para el máximo, mediana, media y RIQ, sin embargo, el mejor valor de pérdida es aquel que sea menor, en donde VGG16 tiene el mejor valor.

Tabla 11

Olivo: Métricas sobre todos los experimentos

Datos	Métrica	Valor	MobileNetV2	MobileNet	VGG16	NASNetLarge
Entrenamiento	Pérdida	min ↓	0,041403	0,022346	0,000000	0,000754
		max ↓	1,180585	4,582973	0,128752	1,944656
		median ↓	0,135602	0,115776	0,008175	0,656865
		mean ↓	0,174147	0,202873	0,014702	0,417371
		iqr ↓	0,104332	0,101347	0,021269	0,722727
	Exactitud	min ↑	0,605000	0,450000	0,965000	0,493333
		max ↑	0,993333	0,993333	1,000000	1,000000
		median ↑	0,950000	0,958333	0,998333	0,993333
		mean ↑	0,936038	0,940955	0,996075	0,975072
		iqr ↓	0,040000	0,036667	0,005000	0,018333
Validación	Pérdida	min ↓	0,174358	0,104122	0,011384	0,021931
		max ↓	4,642312	1,070804	1,088125	1,771492
		median ↓	0,340608	0,189414	0,204248	0,246572
		mean ↓	0,643429	0,219111	0,233563	0,395289
		iqr ↓	0,174720	0,078970	0,171318	0,320634
	Exactitud	min ↑	0,273333	0,780000	0,840000	0,313333
		max ↑	0,940000	0,973333	0,993333	0,993333
		median ↑	0,886667	0,940000	0,966667	0,946667
		mean ↑	0,824287	0,932013	0,962087	0,872793
		iqr ↓	0,046667	0,026667	0,020000	0,073333

A continuación, el modelo seleccionado se utilizó para generar la Matriz de Confusión que puede verse en la Tabla 12. La Matriz de Confusión nos mostró una clasificación perfecta en el conjunto de entrenamiento, y un pequeño número de errores en el conjunto de validación: una imagen que pertenecía a la clase Oidium fue clasificada como Mosaico. Además, dos imágenes pertenecientes a la clase Oidio se clasificaron como Alternaria.

Tabla 12*Olivo: Matriz de confusión*

		Conjunto de entrenamiento		
		Clase	Deficiencias	Fumagina
Clase verdadera	Deficiencias	200	0	0
	Fumagina	0	200	0
	Virosis	0	0	200
	Conjunto de Validación			
	Class	Deficiencias	Fumagina	Virosis
	Deficiencias	50	0	0
Fumagina	0	50	0	
Virosis	0	1	50	
		Clase predicha		

Las métricas de clasificación típicas, como la precisión, la sensibilidad y la puntuación F1, se tuvieron en cuenta para la evaluación posterior del modelo seleccionado y se muestran en la Tabla 13. La puntuación F1 media de todas las enfermedades fue de 0,9933 en el conjunto de validación. Este valor coincide con la exactitud del modelo debido al balance de clases.

Tabla 13*Olivo: Métricas de clasificación por enfermedad*

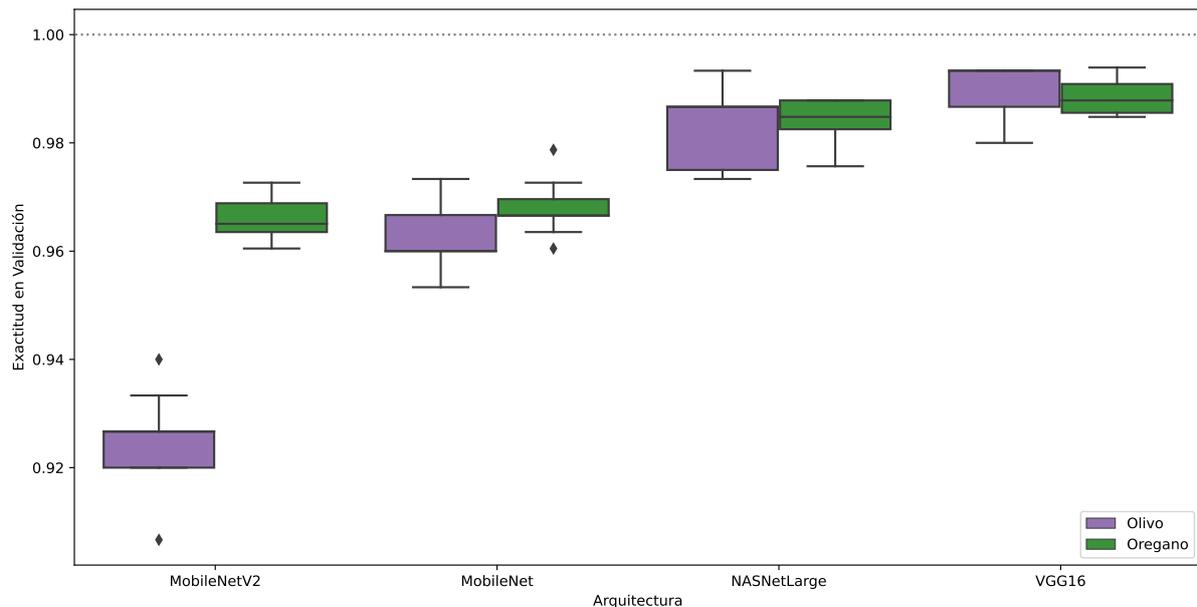
Conjunto de entrenamiento			
Clase	Precisión	Sensibilidad	F1-Score
Deficiencias	1,0000	1,0000	1,0000
Fumagina	1,0000	1,0000	1,0000
Virosis	1,0000	1,0000	1,0000
Conjunto de validación			
Clase	Precisión	Sensibilidad	F1-Score
Deficiencias	1,0000	1,0000	1,0000
Fumagina	0,9804	1,0000	0,9901
Virosis	1,0000	0,9800	0,9899

Por último, el modelo seleccionado está disponible en nuestro [GitHub](#) para su uso y evaluación posteriores.

4.4 COMPARACIÓN MODELOS POR CONJUNTO DE DATOS

Figura 20

Comparación modelos por conjunto de datos



Nota: Elaboración propia

Se han trabajado con 02 conjuntos de datos (olivo y orégano) cuyo protocolo de toma de muestras ha sido anterior a la investigación. La diferencia en el protocolo de toma de muestras se encuentra en que las imágenes de olivo se encuentran en un entorno más controlado, en el cual cada hoja es colocada sobre un papel en blanco sin interferencia de ningún otro objeto. Por otro lado, las imágenes de orégano fueron tomadas directamente del cultivo (entorno no controlado) para luego ser separadas en hojas en una etapa de pre-procesamiento, encontrándonos como resultado final con hojas enfocadas y centradas en la imagen pero que tienen objetos (otras hojas) en el fondo. Algunas muestras de estos 02 conjuntos de datos pueden ser vistas en las Figuras 9 y 10. Es entonces importante responder comparativamente cuál es el mejor modelo y si el tener las muestras en un entorno controlado (sin objetos en el fondo de

la hoja) ejerce algún impacto en el rendimiento de los modelos.

En la Figura 20 se muestra la exactitud en validación a lo largo de los experimentos y para cada uno de los modelos. Se puede ver, una similitud para los modelos VGG16, NASNetLarge y MobileNet. Para el modelo MobileNetV2 vemos un rendimiento menor cuando se utiliza para la clasificación de enfermedades de orégano que cuando se utiliza para enfermedades de olivo. Adicionalmente, es visible que el rango intercuartílico de la exactitud de los modelos entrenados para clasificar las enfermedades de orégano es menor respecto a las enfermedades de olivo.

Podemos establecer la siguiente prueba de hipótesis:

- H_a : La exactitud en validación en los conjuntos de datos de olivo y orégano es mayor para el modelo VGG16 que para el modelo NASNetLarge.
- H_0 : La exactitud en validación en los conjuntos de datos de olivo y orégano no es mayor para el modelo VGG16 que para el modelo NASNetLarge.
- Se asume el nivel de significancia con el valor $\alpha = 0,05$.
- Se utilizará la prueba de U de Mann-Whitney como estadístico de prueba, ya que la distribución de los datos no sigue la distribución normal.
- Se determinó el valor estadístico, los resultados de la aplicación de la prueba se muestran en la Tabla 14.
- El valor obtenido de significación fue de 0,999 (menor a 0,05), por lo tanto, se acepta la hipótesis nula concluyendo que la exactitud en validación en los conjuntos de datos de olivo y orégano no es mayor para el modelo VGG16 que para el modelo NASNetLarge.

Tabla 14

Prueba de U de Mann-Whitney VGG16 y NASNetLarge

Grupo	N	Rango Promedio	Suma de Rangos	U de Mann-Whitney	P valor
VGG16	20	14,875	297,5	87,5	0,999
NASNetLarge	20	26,125	522,5		

Además, podemos establecer la siguiente prueba de hipótesis:

- H_a : La exactitud en validación para el modelo VGG16 con el conjunto de datos de olivo es mayor que en el conjunto de datos de orégano.
- H_0 : La exactitud en validación para el modelo VGG16 con el conjunto de datos de olivo no es mayor que en el conjunto de datos de orégano.
- Se asume el nivel de significancia con el valor $\alpha = 0,05$.
- Se utilizará la prueba de U de Mann-Whitney como estadístico de prueba, ya que la distribución de los datos no sigue la distribución normal.
- Se determinó el valor estadístico, los resultados de la aplicación de la prueba se muestran en la Tabla 15.
- El valor obtenido de significación fue de 0,168 (menor a 0,05), por lo tanto, se acepta la hipótesis nula concluyendo que la exactitud en validación para el modelo VGG16 con el conjunto de datos de olivo no es mayor que en el conjunto de datos de orégano.

Tabla 15

Prueba de U de Mann-Whitney Olivo vs Orégano con VGG16

Grupo	N	Rango Promedio	Suma de Rangos	U de Mann-Whitney	P valor
Olivo	10	11,8	118,0	63,0	0,168
Orégano	10	9,2	92,0		

Finalmente, podemos establecer la siguiente prueba de hipótesis:

- H_a : La exactitud en validación para el modelo NASNetLarge con el conjunto de datos de olivo es mayor que en el conjunto de datos de orégano.
- H_0 : La exactitud en validación para el modelo NASNetLarge con el conjunto de datos de olivo no es mayor que en el conjunto de datos de orégano.
- Se asume el nivel de significancia con el valor $\alpha = 0,05$.
- Se utilizará la prueba de U de Mann-Whitney como estadístico de prueba, ya que la distribución de los datos no sigue la distribución normal.
- Se determinó el valor estadístico, los resultados de la aplicación de la prueba se muestran en la Tabla 15.
- El valor obtenido de significación fue de 0,168 (menor a 0,05), por lo tanto, se acepta la hipótesis nula concluyendo que la exactitud en validación para el modelo NASNetLarge con el conjunto de datos de olivo no es mayor que en el conjunto de datos de orégano.

Tabla 16

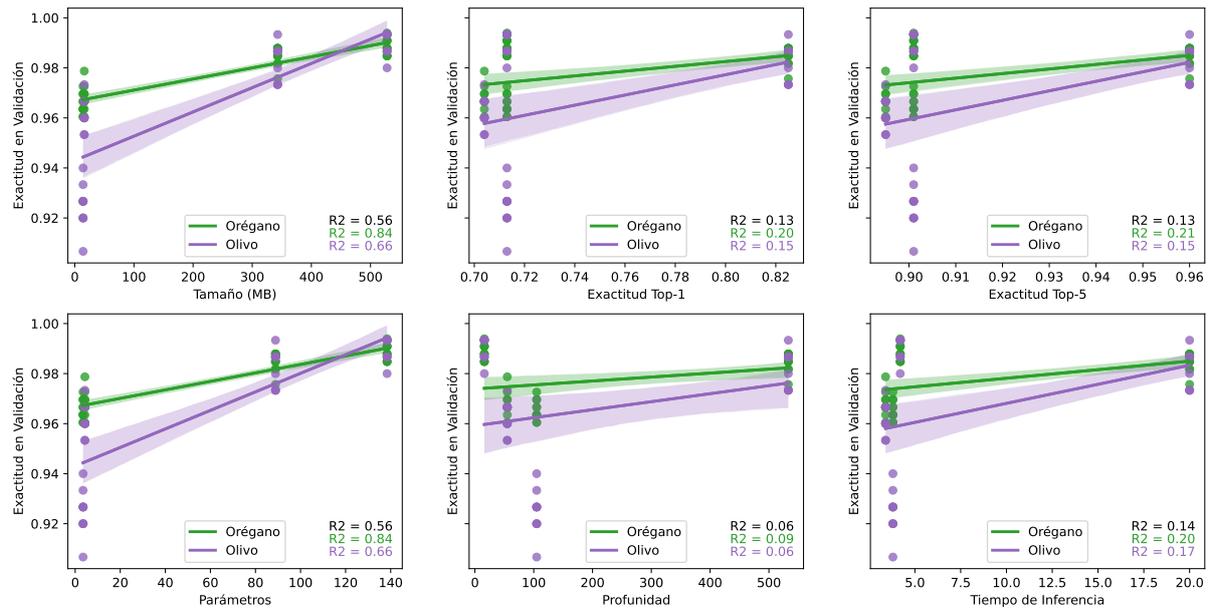
Prueba de U de Mann-Whitney con NASNetLarge Olivo vs Orégano

Grupo	N	Rango Promedio	Suma de Rangos	U de Mann-Whitney	P valor
Olivo	10	9,6	96,0	41,0	0,767
Orégano	10	11,4	114,0		

4.5 CORRELACIÓN ENTRE MODELOS Y SU RENDIMIENTO

Figura 21

Criterios de selección vs rendimiento del modelo



Nota: Elaboración propia

La Figura 21 muestra la relación entre los criterios de selección y el rendimiento (exactitud) de los modelos para cada conjunto de datos. En la Figura se ha trazado una regresión lineal y se muestra el valor R2 para cada conjunto de datos y en promedio. Para el tamaño (MB) y los parámetros, el valor R2 fue de 0,84 para el conjunto de datos de orégano y 0,66 para el conjunto de datos de olivo, lo que indica una fuerte correlación. Para los demás criterios de selección, el valor R2 es demasiado pequeño para indicar correlación alguna.

DISCUSIONES

En cuanto a otros estudios relacionados con la clasificación de las enfermedades de orégano (Othman et al., 2022; Prajapati et al., 2023) y olivo (Lachgar et al., 2022; Ksibi et al., 2022; Alshammari et al., 2023; Alshammari et al., 2022), es imposible hacer una comparación directa porque sus conjuntos de datos no son públicos y las enfermedades son diferentes. Sin embargo, estamos de acuerdo con ellos en que es posible clasificar la enfermedad de la hoja del orégano utilizando redes neuronales convolucionales.

Los resultados experimentales obtenidos reafirman los hallazgos y son consistentes con (Yandouzi et al., 2022; Osco-Mamani and Chaparro-Cruz, 2023; Abubakar et al., 2022; Uğuz and Uysal, 2021), donde VGG16 muestra resultados superiores en comparación con Inceptionv3, ResNet50V2, InceptionResNetV2, Xception, DenseNet, MobileNet, MobileNetV2, y NASNetMobile. Sin embargo, no es viable comparar nuestros resultados con los estudios (Prajapati et al., 2023; Teo et al., 2022; Jaiswal et al., 2022; Alam et al., 2022; Kumar et al., 2022; Alruwaili et al., 2019) en los que la arquitectura VGG16 no se ha tenido en cuenta en sus análisis.

Además, los resultados obtenidos reafirman los hallazgos en relación con el uso de Aumento de Datos (Teo et al., 2022; Al-Dhabyani et al., 2019; Govathoti et al., 2022), Aprendizaje por Transferencia Learning (Jaiswal et al., 2022; Alam et al., 2022; Kumar et al., 2022), y Ajuste Fino (Abubakar et al., 2022; Alharbi et al., 2020).

Finalmente, nuestro exhaustivo estudio experimental comparativo introduce nuevos hallazgos relacionados con los criterios de selección utilizados para comparar los modelos y los conjuntos de datos.

CONCLUSIONES

Se ha logrado el objetivo general de la investigación que es clasificar enfermedades de olivo (*Olea europea*) y orégano (*Origanum vulgare*) mediante modelos de redes neuronales convolucionales en Tacna - Perú a través de cumplir los objetivos específicos establecidos:

1. Se diseñaron arquitecturas de redes neuronales convolucionales utilizando el modelo base de VGG16, MobileNet, MobileNetV2 y NASNetLarge unido a una capa densa de 256 neuronas, función de activación ReLU y una capa densa de 3 neuronas para mapear todo el modelo a las 3 enfermedades para clasificar enfermedades de olivo y orégano.
2. Se entrenaron modelos de redes neuronales convolucionales para clasificar enfermedades de olivo y orégano utilizando aumento de datos, aprendizaje por transferencia y ajuste fino, encontrándose que: (i) Cuando se aplica Ajuste Fino se obtiene una mejora promedio del 19,315% sobre aprendizaje por transferencia en la exactitud del conjunto de validación, (ii) El modelo NASNetLarge tiene un bajo rendimiento cuando sólo se aplica aprendizaje por transferencia.
3. Se compararon los modelos de redes neuronales convolucionales para clasificar enfermedades de olivo y orégano a través de un amplio estudio experimental comparativo, encontrándose que: (i) Los modelos MobileNet y MobileNetV2 tienen un rendimiento similar (excepto para MobileNetV2 en olivo donde la exactitud fue visiblemente menor), (ii) No existen diferencias significativas (p -valor=0,168) en la exactitud en validación entre VGG16 y NASNetLarge para los conjuntos de

datos de olivo y orégano (siendo estos modelos los que mayor exactitud presentaron), (*iii*) No existen diferencias significativas para VGG16 (p -valor=0,168) y NASNetLarge (p -valor=0,767) en la exactitud en validación al comparar los conjuntos de datos de olivo (fondo ausente de objetos - entorno controlado) vs orégano (fondo de hoja con otras hojas - entorno no controlado).

4. Se correlacionaron las características de las arquitecturas y su rendimiento, encontrándose una fuerte relación significativa para orégano ($R^2 = 0,84$) y olivo ($R^2 = 0,66$) entre el número de parámetros y tamaño (en MB) con la exactitud en validación del modelo.

Adicionalmente, se publicaron nuevos conjuntos de datos destinado a clasificar las siguientes enfermedades foliares del *i*) Orégano: Alternaria, Mosaico y Oidio que afectan a los cultivos de Susapaya-Tarata en Tacna, Perú. *ii*) Olivo: Deficiencias Nutricionales, Fumagina y Virosis que afectan a los cultivos de La Yarada-Los Palos - Tacna, Perú.

Finalmente, se obtuvieron y publicaron los modelos de red neuronal convolucional seleccionado con una exactitud del 100% y 99,39%, y 100% y 99,33% en los conjuntos de entrenamiento y validación de orégano y olivo respectivamente, basado en nuestro modelo de arquitectura VGG16 modificado, entrenado mediante Aumento de Datos, Aprendizaje por Transferencia y Ajuste Fino, que resuelve la tarea de clasificación de enfermedades de las hojas de orégano. La puntuación F1 fue de 1 para las enfermedades del conjunto de entrenamiento y de 0,9725, 0,9973 y 0,9886 en las enfermedades Alternaria, Mosaico y Oidium en orégano y 1,00, 0,9901 y 0,9899 en las enfermedades de Deficiencias Nutricionales, Fumagina y Virosis en olivo.

RECOMENDACIONES

Los trabajos futuros que originan esta investigación podrían orientarse a:

1. Desplegar los modelos publicados, que resuelven la tarea de clasificación de enfermedades de orégano y olivo, en dispositivos móviles y de borde para su validación y uso agrícola.
2. Experimentar con diferentes configuraciones de aumento de datos.
3. Comparar los modelos con mejor rendimiento (VGG16 y NASNetLarge) con otros modelos.
4. Someter a prueba la correlación fuerte entre número de parámetros y tamaño (en MB) con otros modelos adicionales a los ya utilizados en la presente investigación.

REFERENCIAS BIBLIOGRÁFICAS

- Abubakar, U. B., Boukar, M. M., and Adeshina, S. (2022). Evaluation of parameter fine-tuning with transfer learning for osteoporosis classification in knee radiograph. *International Journal of Advanced Computer Science and Applications*, 13(8).
- Al-Dhabyani, W., Gomaa, M., Khaled, H., and Aly, F. (2019). Deep learning approaches for data augmentation and classification of breast masses using ultrasound images. *Int. J. Adv. Comput. Sci. Appl*, 10(5):1–11.
- Alam, F., Tisha, F. C., Rahman, S. A., Sultana, S., Chowdhury, M. A. M., Ahmed, W. R., and Arefin, M. S. (2022). Automated brain disease classification using transfer learning based deep learning models. *International Journal of Advanced Computer Science and Applications*, 13(9).
- Alfaro, R. C., Garcia, E. C., Salazar, L., Sepulveda, G., Huarachi, A., and Bartolini, I. (2003). Avances en la investigación colaborativa y control integrado de la "hoja de hoz" en los cultivos de olivo de tacna y arica. *Ciencia & Desarrollo*, (7):7–12.
- Alharbi, F., Elbashir, M. K., Mohammed, M., and Mustafa, M. E. (2020). Fine-tuning pre-trained convolutional neural networks for women common cancer classification using rna-seq gene expression. *International Journal of Advanced Computer Science and Applications*, 11(11).
- Alruwaili, M., Alanazi, S., Abd El-Ghany, S., and Shehab, A. (2019). An efficient deep learning model for olive diseases detection. *International Journal of Advanced Computer Science and Applications*, 10(8).

- Alshammari, H., Gasmi, K., Ben Ltaifa, I., Krichen, M., Ben Ammar, L., and Mahmood, M. A. (2022). Olive disease classification based on vision transformer and cnn models. *Computational Intelligence and Neuroscience*, 2022.
- Alshammari, H. H., Taloba, A. I., and Shahin, O. R. (2023). Identification of olive leaf disease through optimized deep learning approach. *Alexandria Engineering Journal*, 72:213–224.
- Arias, F. (2012). El proyecto de investigación. introducción a la metodología científica.. editorial episteme. *Caraca-República Bolivariana de Venezuela*.
- Baumann, H. (2022). Agraria.pe. exportaciones de la aceituna Perú. <https://agraria.pe/buscar?q=exportaciones+de+aceituna+peru>. Accessed: 2022-09-30.
- Casilla García, M. E. (2011). Análisis de los factores que influyen en la vecería del olivo (olea europea l.) en la región tacna.
- Chagua, T. and Amador, E. (2016). *Comercialización de orégano orgánico como especialidad en la región Tacna-Perú*. PhD thesis, Universidad de Buenos Aires. Facultad de Agronomía. Escuela para Graduados.
- Condori Tintaya, F. and Fernández Cutire, O. (2022). Resultados económicos de producción de orégano (origanum vulgare l.) en cuencas interandinas de la región tacna, Perú. *Idesia (Arica)*, 40(2):17–26.
- Contreras, R., Calle, I., Osses, R., Aguayo, F., Porcile, V., and Arias, M. (2021). Identificación genética del orégano de putre (origanum vulgare l.) mediante its y microsatélites, una especie reconocida en Chile con sello de origen. *Boletín Latinoamericano y del Caribe de Plantas Medicinales y Aromáticas*, 20(2):177–194.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

- Ekman, M. (2021). *Learning Deep Learning: Theory and Practice of Neural Networks, Computer Vision, NLP, and Transformers Using TensorFlow*. Addison-Wesley Professional.
- FAO (2019). El trabajo de la fao sobre el cambio climático conferencia de las naciones unidas sobre el cambio climático 2019. <http://www.fao.org/3/a-i8037s.pdf>.
- Fernández-Escobar, R., Guerreiro, M., Benlloch, M., and Benlloch-González, M. (2016). Symptoms of nutrient deficiencies in young olive trees and leaf nutrient concentration at which such symptoms appear. *Scientia Horticulturae*, 209:279–285.
- Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.
- Govathoti, S., Reddy, A. M., Kamidi, D., BalaKrishna, G., Padmanabhuni, S. S., and Gera, P. (2022). Data augmentation techniques on chilly plants to classify healthy and bacterial blight disease leaves. *International Journal of Advanced Computer Science and Applications*, 13(6).
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- Hernández Sampieri, R., Fernández Collado, C., Baptista Lucio, P., et al. (2018). *Metodología de la investigación*, volume 4. McGraw-Hill Interamericana México.
- Hinton, G., Srivastava, N., and Swersky, K. (2012). Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8):2.
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

- Huacollo Alvarez, M. A. (2014). Manejo ecológico de plagas y enfermedades del orégano.
- Hubel, D. H. and Wiesel, T. N. (1959). Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148(3):574.
- Iparraguirre-Villanueva, O., Guevara-Ponce, V., Roque Paredes, O., Sierra-Liñan, F., Zapata-Paulini, J., and Cabanillas-Carbonell, M. (2022). Convolutional neural networks with transfer learning for pneumonia detection.
- Jaiswal, P., Katkar, V., and Bhirud, S. (2022). Multi oral disease classification from panoramic radiograph using transfer learning and xgboost. *International Journal of Advanced Computer Science and Applications*, 13(12).
- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.
- Ksibi, A., Ayadi, M., Soufiene, B. O., Jamjoom, M. M., and Ullah, Z. (2022). Mobires-net: A hybrid deep learning model for detecting and classifying olive leaf diseases. *Applied Sciences*, 12(20):10278.
- Kumar, J. S., Anuar, S., and Hassan, N. H. (2022). Transfer learning based performance comparison of the pre-trained deep neural networks. *International Journal of Advanced Computer Science and Applications*, 13(1).
- Lachgar, M., Hrimech, H., Kartit, A., et al. (2022). Optimization techniques in deep convolutional neuronal networks applied to olive diseases classification. *Artificial Intelligence in Agriculture*, 6:77–89.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

- Lopez, E. (2022). Exportaciones de orégano y cúrcuma crecieron 50
- Min, S., Lee, B., and Yoon, S. (2017). Deep learning in bioinformatics. *Briefings in bioinformatics*, 18(5):851–869.
- MINCETUR (2022). Perfil de mercado y competitividad exportadora de la aceituna. https://www.mincetur.gob.pe/wp-content/uploads/documentos/comercio_exterior/plan_exportador/publicaciones/Aceituna.pdf.
- Molina de la Rosa, J., Jiménez Herrera, B., Ruiz Coletto, F., García Zamorano, F., Cano Rodríguez, J., and Pérez García, J. (2017). Técnicas de cultivo: Plagas y enfermedades del olivo. *Consejería de Agricultura, Pesca y Desarrollo Rural: Instituto de Investigación y Formación Agraria y Pesquera*. Recuperado de https://www.juntadeandalucia.es/export/drupaljda/publicacion/17/07/1.%20Plagas%20y%20enferm_olivo_2017%20BAJA.pdf.
- Murphy, F., Fauquet, C., Bishop, D., Ghabrial, S., Jarvis, A., Martelli, G., Mayo, M., and Summers, M. (1995). Sixth report of the international committee on taxonomy of viruses.
- Oscó-Mamani, E. F. and Chaparro-Cruz, I. N. (2023). Highly accurate deep learning model for olive leaf disease classification: A study in tacna-perú. *International Journal of Advanced Computer Science and Applications*, 14(4).
- Othman, N. A., Damanhuri, N. S., Ali, N. M., Meng, B. C. C., and Abd Samat, A. A. (2022). Plant leaf classification using convolutional neural network. In *2022 8th International Conference on Control, Decision and Information Technologies (CoDIT)*, volume 1, pages 1043–1048. IEEE.
- Prajapati, S., Qureshi, S., Rao, Y., Nadkarni, S., Retharekar, M., and Avhad, A. (2023). Plant disease identification using deep learning. In *2023 4th International Conference for Emerging Technology (INCET)*, pages 1–5. IEEE.

- Rengifo, B. J. A. A. (2007). Proyecto de cooperación ue-perú en materia de asistencia técnica relativa al comercio apoyo al programa estrategico nacional de exportaciones asistencia técnica para los planes operativos de.
- Salamanca García, M. A. and Sánchez Bermúdez, M. Y. (2009). Extracción y caracterización de la oleorresina del orégano (*origanum vulgare*).
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520.
- SENASA (2021). Tacna promueve control de plagas. <https://www.senasa.gob.pe/senasacontigo/tacna-senasa-promueve-control-de-plagas-del-olivo/-con-metodos-no-contaminantes/>.
- Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Team, K. (2023). Keras documentation: Keras applications.
- Teo, H. C., Hashim, U. R., Ahmad, S., Salahuddin, L., Ngo, H. C., and Kanchymalay, K. (2022). Efficacy of the image augmentation method using cnn transfer learning in identification of timber defect. *International Journal of Advanced Computer Science and Applications*, 13(5).
- Thomkaew, J. and Intakosum, S. (2022). Improvement classification approach in tomato leaf disease using modified visual geometry group (vgg)-inceptionv3. *International Journal of Advanced Computer Science and Applications*, 13(12).
- Uğuz, S. and Uysal, N. (2021). Classification of olive leaf diseases using deep convolutional neural networks. *Neural computing and applications*, 33(9):4133–4149.
- Woudenberg, J., Groenewald, J., Binder, M., and Crous, P. (2013). *Alternaria* redefined. *Studies in mycology*, 75(1):171–212.

- Yandouzi, M., Grari, M., Idrissi, I., Boukabous, M., Moussaoui, O., Azizi, M., Ghoumid, K., and Elmiad, A. K. (2022). Forest fires detection using deep transfer learning. *International Journal of Advanced Computer Science and Applications*, 13(8).
- Zhang, S., Huang, W., and Zhang, C. (2019). Three-channel convolutional neural networks for vegetable leaf disease recognition. *Cognitive Systems Research*, 53:31–41.
- Zoph, B. and Le, Q. V. (2016). Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*.
- Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. (2018). Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710.

ANEXOS

Anexo 1: Código principal de experimentación

```
#importamos librerías
import matplotlib.pyplot as plt
import numpy as np
from tensorflow import keras

#solicitamos el dataset con el cual se va a trabajar
dataset = input("Ingrese el dataset Olivo/Orégano: ")

#ubicación de los conjunto de datos
train_dir = '/home/ichaparroc/datasets/'+dataset+'/train'
valid_dir = '/home/ichaparroc/datasets/'+dataset+'/valid'

#pre-procesamiento de datos de entrenamiento (con aumento de datos)
datagen_train = keras.preprocessing.image.ImageDataGenerator(
    samplewise_center=True,
    rotation_range=45,
    zoom_range=0.2,
    width_shift_range=0.2,
    height_shift_range=0.2,
    horizontal_flip=True,
    vertical_flip=True,
```

```
)

#pre-procesamiento de datos de validación (SIN aumento de datos)
datagen_valid = keras.preprocessing.image.ImageDataGenerator(samplewise_center=True)

#cargamos los datos de entrenamiento y validación de un directorio
train_it = datagen_train.flow_from_directory(
    train_dir,
    target_size=(224, 224),
    color_mode='rgb',
    class_mode='categorical',
    batch_size=8,
)

valid_it = datagen_valid.flow_from_directory(
    valid_dir,
    target_size=(224, 224),
    color_mode='rgb',
    class_mode='categorical',
    batch_size=8,
)

#hiperparámetros
epocas = 100
inicio = 0
fin = 9

for i in range(inicio, fin + 1):
    #construimos la arquitectura
```

```
##base_model = keras.applications.NASNetLarge(  
##base_model = keras.applications.MobileNet(  
##base_model = keras.applications.MobileNetV2(  
base_model = keras.applications.VGG16(  
    weights='imagenet', # Load weights pre-trained on ImageNet.  
    input_shape=(224, 224, 3),  
    include_top=False)  
  
#congelamos las capas  
for layer in base_model.layers:  
    layer.trainable = False  
  
#añadimos capas personalizadas de clasificación  
x = base_model.output  
x = GlobalAveragePooling2D()(x)  
x = Dense(256, activation='relu')(x)  
predictions = Dense(3, activation='softmax')(x)  
model = Model(inputs=base_model.input, outputs=predictions)  
model.compile(optimizer='adam', loss='categorical_crossentropy',  
metrics=['accuracy'])  
  
#establecemos que se guarde el mejor modelo  
model_checkpoint_callback = keras.callbacks.ModelCheckpoint(  
    filepath= './t1_'+str(i)+'.h5',  
    #save_weights_only=True,  
    monitor='val_accuracy',  
    mode='max',  
    save_best_only=True)
```

```
#ajustamos (entrenamos) usando aprendizaje por transferencia
history = model.fit(train_it,
validation_data=valid_it,
steps_per_epoch=train_it.samples/train_it.batch_size,
validation_steps=valid_it.samples/valid_it.batch_size,
epochs=epocas,
callbacks=[model_checkpoint_callback])

#guardamos las estadísticas
hist_df = pd.DataFrame(history.history)
hist_csv_file = './history_tl_'+str(i)+'.csv'
with open(hist_csv_file, mode='w') as f:
    hist_df.to_csv(f)

#cargamos el modelo
model.load_weights('./tl_'+str(i)+'.h5')

#descongelamos las capas de extracción de características
for layer in model.layers:
    layer.trainable = True

#ajustamos el ratio de aprendizaje para ajuste fino
model.compile(optimizer=keras.optimizers.RMSprop(learning_rate = .00001),
loss='categorical_crossentropy', metrics=['accuracy'])

#establecemos que se guarde el mejor modelo
model_checkpoint_callback = keras.callbacks.ModelCheckpoint(
filepath=str(nombre_carpeta)+'/ConvNeXtXLarge_ft_'+str(i)+'.h5',
```

```
#save_weights_only=True,  
monitor='val_accuracy',  
mode='max',  
save_best_only=True)  
  
#ajustamos (entrenamos) usando ajuste fino  
history = model.fit(train_it,  
validation_data=valid_it,  
steps_per_epoch=train_it.samples/train_it.batch_size,  
validation_steps=valid_it.samples/valid_it.batch_size,  
epochs=epocas,  
callbacks=[model_checkpoint_callback])  
  
#guardamos las estadísticas  
hist_df = pd.DataFrame(history.history)  
hist_csv_file = str(nombre_carpeta)+'/'+'history_ft_'+str(i)+''.csv'  
with open(hist_csv_file, mode='w') as f:  
    hist_df.to_csv(f)
```