

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE



“DETECCIÓN DE ENFERMEDADES EN PLANTAS UTILIZANDO PROCESAMIENTO DE IMÁGENES”

NICOLE AILÍN GALLARDO CASTILLO

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: Hubert Hoffmann
Profesor Correferente: Claudio Lobos

Agosto - 2023

DEDICATORIA

Dedicado a mi madre, padre y perro, quienes me han acompañado durante todo el proceso.

AGRADECIMIENTOS

Agradezco a mis padres que siempre me han brindado su apoyo incondicional para poder cumplir todos mis objetivos personales y académicos. Ellos son los que con su cariño me han impulsado siempre a perseguir mis metas y nunca abandonarlas frente a las adversidades. También agradezco a mi perro, que muchas veces visitó mi área de trabajo sin que lo llamara para acompañarme.

Le agradezco muy profundamente a mi Profesor guía, por su dedicación y paciencia, y por siempre responder mis consultas.

RESUMEN

Resumen— La detección de enfermedades y plagas en plantas es un proceso complejo que requiere la participación de especialistas y monitoreo constante. El diagnóstico tardío de estas amenazas puede resultar en severos daños a los cultivos y pérdidas económicas significativas. Para abordar este problema se propone el uso de *Deep Learning* para el procesamiento de imágenes, así disminuyendo el nivel de capacitación y tiempo necesarios para la detección de enfermedades. Los modelos de redes neuronales convolucionales (CNNs) propuestos alcanzan más del 99 % de exactitud en el reconocimiento de tres tipos de enfermedades en vides. Se demuestra que estos modelos son una excelente opción para desarrollar sistemas de detección de enfermedades en cultivos, como aplicaciones móviles, que puedan permitir a agricultores pequeños simplificar el diagnóstico e incorporar los beneficios de emplear nuevas tecnologías en sus tareas a un bajo costo.

Palabras Clave— Detección de enfermedades en vides; procesamiento de imágenes; deep learning; redes neuronales convolucionales.

ABSTRACT

Abstract— The detection of diseases and pests in plants is a complex process that requires the involvement of specialists and constant monitoring. Late diagnosis of these threats can result in severe damage to crops and significant monetary losses. To solve this problem, the use of Deep Learning for image processing is proposed, so that the amount of training and time needed for detecting diseases are lower. The proposed convolutional neural network models achieve over 99 % accuracy recognizing three different types of diseases present in grapes. It is demonstrated that these models are an excellent option for developing crop disease detection systems, like mobile apps, that can allow small farmers to simplify diagnosis and incorporate the benefits of using new technologies in their work at a low cost.

Keywords— Grape disease detection; image processing; deep learning; convolutional neural networks.

GLOSARIO

ANN: *Artificial Neural Network*, Red Neuronal Artificial.

CNN: *Convolutional Neural Network*, Red Neuronal Convolutacional.

Dataset: Conjunto de datos.

DL: *Deep Learning*, Aprendizaje Profundo.

GA: *Genetic Algorithm*, Algoritmo Genético.

Grad-CAM: *Gradient-weighted Class Activation Mapping*, Mapa de Activación de Clase Ponderada por el Gradiente.

HOG: *Histogram of Oriented Gradients*, Histogramas de Gradientes Orientados.

HSI: *Hue, saturation, intensity*; Matiz, saturación, intensidad.

Input: Entrada.

Kernel: Núcleo.

KNN: *k-Nearest Neighbor*

LBP: *Local Binary Pattern*, Patrón Binario Local.

ML: *Machine Learning*, Aprendizaje de Máquina.

Output: Salida.

ReLu: *Rectified Linear Unit*, Rectificador.

RF: *Random Forest*

RGB: *Red, green, blue*; Rojo, verde, azul.

SVM: *Support Vector Machine*

ÍNDICE DE CONTENIDOS

RESUMEN	IV
ABSTRACT	IV
GLOSARIO	V
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS	IX
INTRODUCCIÓN	1
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	3
1.1 Contexto y situación actual	3
1.2 Objetivos	5
1.2.1 Objetivo general:	5
1.2.2 Objetivos específicos:	5
CAPÍTULO 2: MARCO CONCEPTUAL	7
2.1 Análisis de técnicas para el reconocimiento de enfermedades en plantas	7
2.1.1 Pasos del procesamiento de imágenes	7
2.1.2 Revisión de la literatura	9
2.2 <i>Deep Learning</i>	12
2.2.1 ImageNet Large Scale Visual Recognition Challenge	14
2.2.2 Redes neuronales convolucionales	15
2.2.3 Redes neuronales residuales (ResNet)	19
2.2.4 EfficientNet	20
2.2.5 Métricas de clasificación	20
2.3 Análisis de cultivos de Chile, sus enfermedades y características	24
2.3.1 Cultivos Chilenos	25
2.3.2 Enfermedades relevantes y cultivos afectados	25
2.3.3 Prevención y tratamiento de enfermedades y plagas	26
CAPÍTULO 3: PROPUESTA DE SOLUCIÓN	28
3.1 Entorno de desarrollo	28
3.2 Obtención de datos para el entrenamiento	28
3.3 Redes neuronales convolucionales	31
3.4 Explicación visual de las predicciones	32
CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN	33
4.1 Resultados Resnet50	33
4.2 Resultados Resnet101	35

4.3 Resultados EfficientNet-B0	37
4.4 Resultados EfficientNet-B1	39
4.5 Representaciones visuales	42
4.6 Análisis del desempeño	45
CAPÍTULO 5: CONCLUSIONES	47
5.1 Trabajo futuro	48
ANEXOS	50
6 Resultados EfficientNet-B1 previo al cambio de configuración	50
REFERENCIAS BIBLIOGRÁFICAS	52

ÍNDICE DE FIGURAS

1	Diagrama representativo de las exportaciones de Chile en el año 2019	3
2	Árbol del problema	6
3	Pasos para el procesamiento de imágenes (inglés).	9
4	Una red neuronal profunda sencilla con dos capas ocultas y una capa de <i>output</i>	13
5	Diagrama de flujo para la implementación de DL aplicado a la detección y clasificación de enfermedades en plantas (inglés)	14
6	Arquitectura simplificada de una CNN con 5 capas para la clasificación del <i>dataset</i> MNIST	16
7	Activaciones de la primera capa convolucional de una CNN simple tras entrenarla en el <i>dataset</i> MNIST de dígitos manuscritos. Es posible observar que la red ha logrado extraer características únicas a dígitos específicos.	17
8	Procedimiento de la capa convolucional de una CNN bidimensional. Inicialmente se introduce <i>zero-padding</i> , luego los elementos del <i>kernel</i> se multiplican con un segmento de la matriz original, produciendo una nueva matriz. Finalmente se aplica <i>max-pooling</i> para obtener un <i>feature map</i>	17
9	Superposición de los mapas de características de las primeras capas convolucionales de una CNN sobre las imágenes de entrada. Se puede apreciar que cada mapa resalta aspectos distintos de la imagen.	18
10	Bloque residual de una ResNet. La conexión atajo en la derecha salta dos capas.	19
11	Arquitectura general de EfficientNet-B0.	20
12	Matriz de confusión binaria.	22
13	Matriz de confusión para tres clases.	23
14	Cantidad de muestras en los conjuntos de entrenamiento y prueba.	29
15	Porcentaje del total de muestras para cada clase en los conjuntos de entrenamiento y prueba.	29
16	Muestra de 12 imágenes pertenecientes a las 4 clases presentes en el <i>dataset</i> de entrenamiento.	30

17 Gráfico de exactitud de entrenamiento y validación según época del modelo Resnet50. Notar que 0 en el eje x corresponde a la época 1. 33

18 Matriz de confusión para modelo Resnet50 al evaluarlo en el conjunto de prueba de Grape Disease. 34

19 Gráfico de exactitud de entrenamiento y validación según época del modelo Resnet101. 0 en el eje x corresponde a la época 1. 35

20 Matriz de confusión para modelo Resnet101 al evaluarlo en el conjunto de prueba de Grape Disease. 36

21 Gráfico de exactitud de validación y entrenamiento según época de modelo EfficientNet-B0. 37

22 Matriz de confusión para el modelo EfficientNet-B0 al evaluarlo en el conjunto de prueba de Grape Disease. 38

23 Gráfico de exactitud de validación y entrenamiento según época de modelo EfficientNet-B1 previo al cambio de configuración. 39

24 Gráfico de exactitud de validación y entrenamiento según época de modelo EfficientNet-B1. 40

25 Matriz de confusión para el modelo EfficientNet-B1 al evaluarlo en el conjunto de prueba de Grape Disease. 41

26 Grad-CAMs del modelo Resnet50. 43

27 Grad-CAMs del modelo Resnet101. 43

28 Grad-CAMs del modelo EfficientNet-B0. 44

29 Grad-CAMs del modelo EfficientNet-B1 44

30 Matriz de confusión para el modelo EfficientNet-B1 previo al cambio de configuración al evaluarlo en el conjunto de prueba de Grape Disease. 50

ÍNDICE DE TABLAS

1 Resultados del modelo Resnet50. 35

2 Resultados del modelo Resnet101. 37

3 Resultados de la evaluación del modelo EfficientNet-B0. 39

4 Resultados de la evaluación del modelo EfficientNet-B1. 42

5 Resultados de la evaluación de todos los modelos. Los mejores valores se destacan con negrita si es que no existen empates. Exactitud se abrevia como Acc. y precisión como Pre. 46

6 Resultados de la evaluación del modelo EfficientNet-B1 previo al cambio de configuración. 51

INTRODUCCIÓN

La presencia de enfermedades y plagas en los cultivos son una grave amenaza para la agricultura, pues estas pueden dañar a las plantas en su completitud y disminuir severamente los niveles de producción, resultando en pérdidas monetarias significativas. Por ello, es esencial que los agricultores se enfrenten a dichas enfermedades con eficacia y las controlen de forma preventiva con una detección temprana. Dependiendo del tamaño del área cultivada, esta tarea puede ser difícil, ya que el proceso de diagnóstico es complejo, lento y costoso.¹

En la actualidad, a pesar del desarrollo de nuevas soluciones tecnológicas, muchos productores todavía confían en la inspección visual y el juicio experto para la detección de enfermedades, pues las alternativas disponibles en el mercado suelen ser costosas y poco fiables. Esto introduce subjetividad al proceso, lo que ocasionalmente puede resultar en diagnósticos errados, y además reduce la agilidad del diagnóstico, pues un solo especialista puede demorar horas en inspeccionar una sola hectárea.²

El campo de *Machine Learning* ha progresado significativamente en la última década, lo que ha resultado en el desarrollo de sistemas clasificadores de imágenes que pueden discriminar entre más de mil categorías con un alto grado de exactitud. Como consecuencia, existen numerosos precedentes de la aplicación de dichas tecnologías en el área del reconocimiento de enfermedades en plantas [Saleem *et al.*, 2019].

En este trabajo se propone la utilización de inteligencia artificial, específicamente *Deep Learning* aplicado al procesamiento de imágenes, para abordar el problema de la detección de enfermedades en los cultivos. Se proponen cuatro modelos basados en redes neuronales convolucionales para el reconocimiento y clasificación de tres tipos de enfermedades presentes en vides, cuyos desempeños son comparados de acuerdo a las métricas de clasificación multi-clase. El objetivo es obtener modelos clasificadores de alta precisión que permitan desarrollar sistemas de detección de enfermedades confiables y accesibles para así brindar a agricultores los beneficios de aplicar nuevas tecnologías en sus quehaceres, especialmente a los productores más pequeños.

Los modelos clasificadores se elaboraron haciendo uso de la transferencia del aprendizaje. Las arquitecturas escogidas fueron Resnet50 y 101, y EfficientNet-B0 y B1, las cuales fueron entrenadas con imágenes de hojas de vides sanas y enfermas provenientes del conjunto de datos Grape Disease. Se elaboran Grad-CAMs con cada uno de los modelos entrenados para generar visualizaciones del proceso de clasificación, así generando explicaciones visuales de las características de las imágenes que cada modelo utiliza al momento de realizar una predicción.

¹Enfermedades De Las Plantas: Tipos Y Prevención | EOS Data Analytics

²Inteligencia Artificial aplicada a la agricultura: detección de tipología y grado de avance de enfermedades en viñedos. | Coreal

La estructura de esta memoria de titulación se describe a continuación:

- **Capítulo 1 - Definición del problema:** Se detallan los problemas causados por la alta complejidad del proceso de detección de enfermedades en plantas y se plantean los objetivos del trabajo.
- **Capítulo 2 - Marco conceptual:** Se analiza y describe material relacionado con el problema planteado. Se realiza un análisis de la literatura con tres enfoques: técnicas de procesamiento de imágenes para la detección de enfermedades en plantas, *Deep Learning* aplicado a la clasificación de imágenes e información acerca de los mayores cultivos de Chile y sus principales amenazas.
- **Capítulo 3 - Propuesta de solución:** Se detallan los modelos propuestos y el proceso de entrenamiento a los que fueron sometidos.
- **Capítulo 4 - Validación de la solución:** Se describen y analizan los resultados obtenidos de acuerdo a las métricas propuestas junto con exponer representaciones visuales del proceso de detección y clasificación de enfermedades.
- **Capítulo 5 - Conclusiones:** Se expone acerca de lo que se aprendió al realizar el estudio, se valida el cumplimiento de los objetivos propuestos, y se proponen trabajos futuros en forma de potenciales mejoras y aplicaciones prácticas para los modelos desarrollados.

CAPÍTULO 1

DEFINICIÓN DEL PROBLEMA

1.1. Contexto y situación actual

En la agricultura, la detección temprana y el manejo de enfermedades y plagas presentes en los cultivos es esencial para alcanzar un rendimiento económico aceptable, independiente del agro-ecosistema en cuestión. No identificar estas amenazas oportunamente puede resultar en daños severos en las plantas, incurriendo en significativas pérdidas económicas si la propagación no se logra contener tempranamente. A nivel mundial, las plagas y enfermedades provocan pérdidas de hasta el **40 % de los cultivos alimentarios**, causando pérdidas monetarias de más de **220.000 millones de dólares al año**, de acuerdo al informe final del Año Internacional de la Sanidad Vegetal [FAO, 2021]. En la actualidad, la agricultura no constituye un porcentaje mayoritario del producto interno bruto de Chile como solía serlo en el pasado, alcanzando cerca del 11 % de las exportaciones totales en términos monetarios, como se puede ver en la Figura 1. Sin embargo, el país sigue estando entre los 10 mayores productores de numerosas frutas, como kiwi, cerezas, y manzanas, entre otras³. La producción de uvas, en particular, no es sólo relevante en términos de la cantidad de fruta exportada, sino también por la producción y exportación de vino, pues Chile es uno de los mayores productores de vino del mundo.

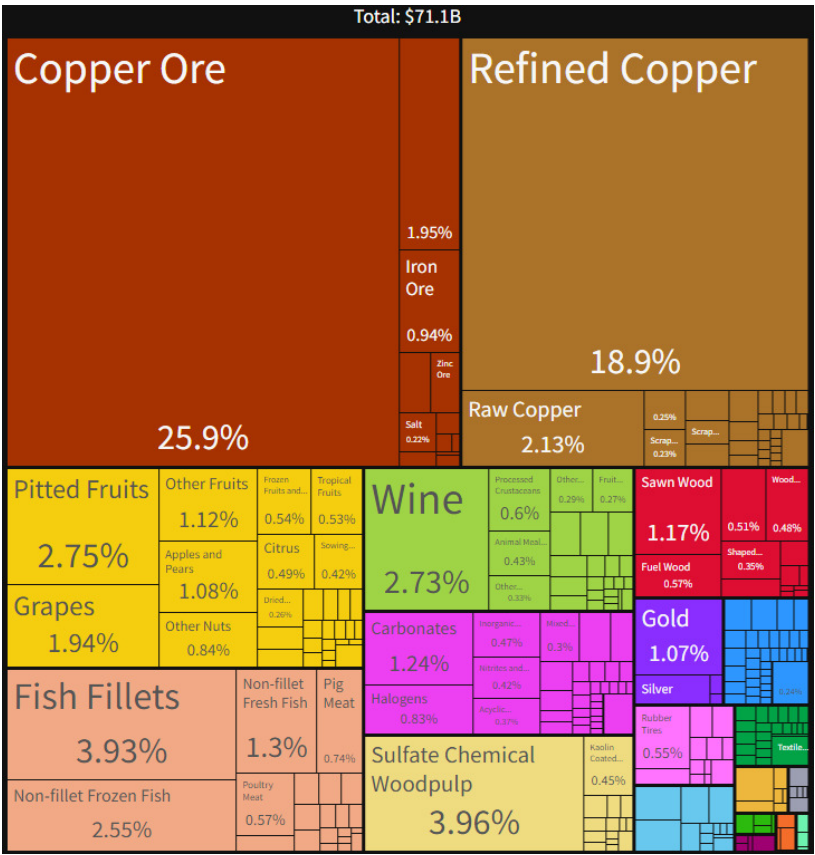


Figura 1: Diagrama representativo de las exportaciones de Chile en el año 2019
Fuente: Chile (CHL) Exports, Imports and Trade Partners | OEC

³Crops and Livestock Products | FAOStat

Monitorear el estado de las plantas manualmente e identificar correctamente las enfermedades o plagas que pueden surgir en éstas es un proceso **difícil** que requiere de capacitación o la asistencia de un especialista, equipos especializados, tiempo y una cantidad considerable de trabajo, lo que significa que detectar enfermedades no sólo es un proceso lento que muchas veces depende de la intervención de terceros, sino que también es **económicamente costoso**. Está también sujeto a prejuicios debido a la dependencia del juicio humano en el proceso; existen distintas enfermedades en cultivos que presentan síntomas similares, lo que conlleva a diagnósticos incorrectos en caso de que un especialista las confunda.

La detección tardía o errónea de enfermedades y plagas en cultivos tiene consecuencias severas, las cuales pueden afectar no solo el bienestar de los cultivos de un agricultor, sino a comunidades completas. Un ineficiente control de plagas peligrosas puede causar pérdidas económicas significantes en toda una región, e incluso al medio ambiente en caso de especies invasivas. En el caso de productos de exportación, una plaga o enfermedad no detectada podría establecerse en otros países si llega a exportarse, mientras que cargamentos que presenten enfermedades serán incapaces de ser exportados y es posible que sean destruidos, causando grandes pérdidas monetarias y de alimento [Boza *et al.*, 2018]. Un diagnóstico incorrecto conlleva a que se utilicen químicos y pesticidas no apropiados para el tratamiento, lo que puede afectar negativamente al cultivo y en el peor de los casos introducir compuestos tóxicos para el consumo en los alimentos sin ningún beneficio.

En Chile, existen iniciativas para capacitar a los agricultores con el fin de que puedan detectar y manejar adecuadamente enfermedades en sus cultivos. Una de éstas, ofrecida por el Ministerio de Agricultura del país, es la iniciativa Chile Agrícola, que ofrece capacitación en línea gratuita para el control y detección de plagas y enfermedades a los agricultores del país mediante cursos, vídeos, fichas informativas y manuales⁴. Aunque este tipo de iniciativas ayuda a que los agricultores puedan detectar enfermedades en sus cultivos por sus propios medios, no es una solución definitiva al problema, pues la capacitación toma tiempo, el proceso de detección sigue siendo igual de lento y el monitoreo manual del estado de los cultivos continúa requiriendo gran cantidad de trabajo.

El problema de la detección de enfermedades y plagas en plantas se abordará en esta memoria de titulación utilizando **Deep Learning** para el procesamiento de imágenes, así reduciendo la cantidad de tiempo y nivel de capacitación necesarios para monitorear el estado de los cultivos. Esto también permitiría a los agricultores aprovechar los beneficios de emplear la computación y tecnología en sus labores.

⁴Control de plagas y enfermedades | Chile Agrícola

1.2. Objetivos

1.2.1. Objetivo general:

- Desarrollar un modelo de clasificación que permita detectar enfermedades en plantas mediante el uso de *Deep Learning*.

1.2.2. Objetivos específicos:

- Desarrollar modelos clasificadores de alta precisión que permitan detectar enfermedades en plantas.
- Detectar tempranamente enfermedades en plantas sin depender de un especialista.
- Llevar la computación y sus beneficios a agricultores pequeños con bajo costo.

El árbol del problema resumiendo esta situación se presenta a continuación en la Figura 2.

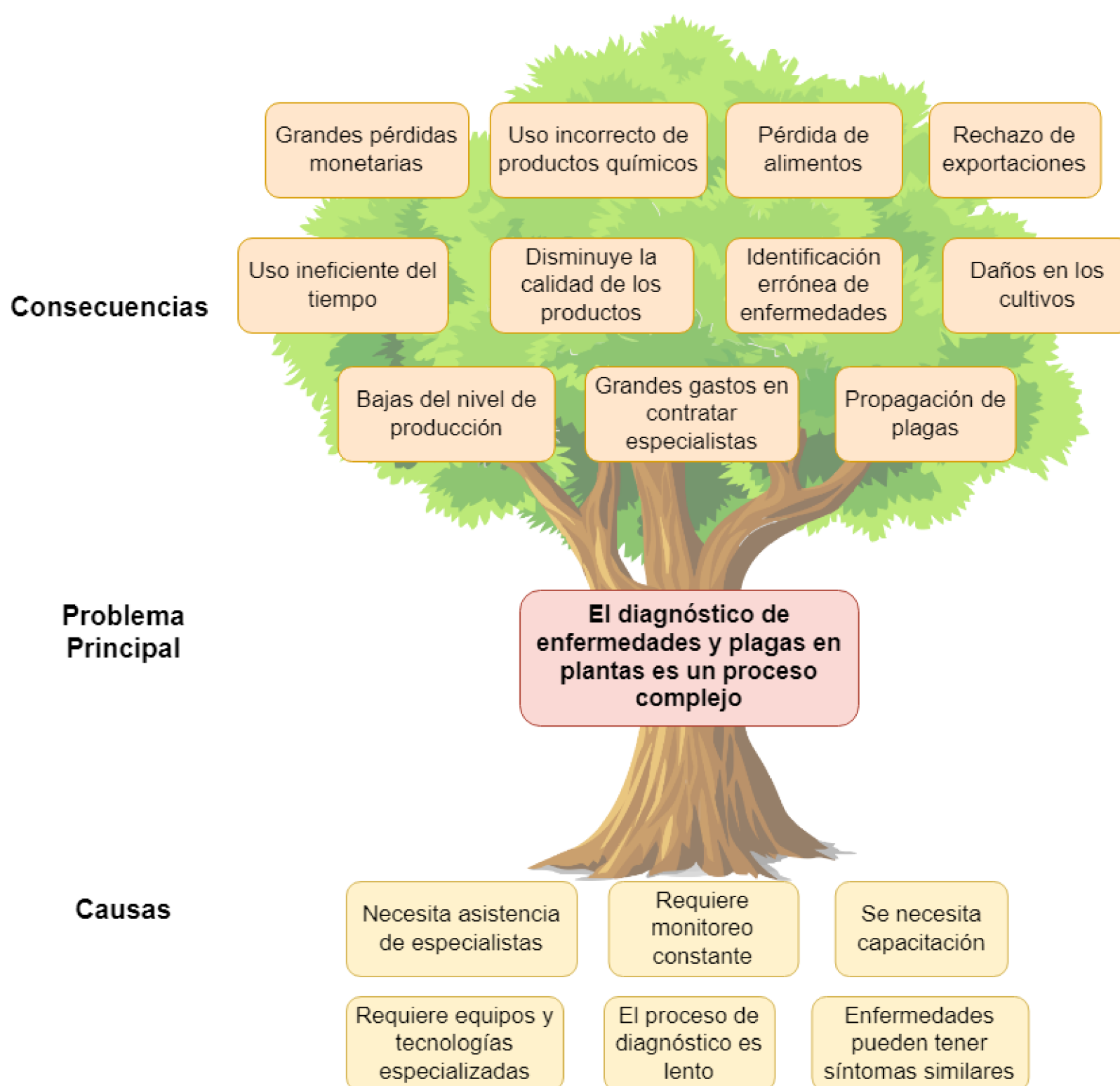


Figura 2: Árbol del problema
Fuente: Elaboración propia.

CAPÍTULO 2

MARCO CONCEPTUAL

La bibliografía que será analizada para el desarrollo de esta memoria se clasifica en tres grupos: Técnicas de procesamiento de imágenes utilizadas en la literatura para reconocer la presencia de enfermedades en plantas, *Deep Learning* y sus modelos aplicados a la clasificación de imágenes, e información acerca de cultivos comunes en Chile y las principales enfermedades y plagas que pueden atacarlos.

2.1. Análisis de técnicas para el reconocimiento de enfermedades en plantas

El uso de procesamiento de imágenes para detectar enfermedades tiene numerosos precedentes en la literatura académica dentro de la última década, y se han utilizado una variedad de técnicas distintas para detectar enfermedades de diversas especies con resultados satisfactorios, pero que en muchos casos tienen oportunidad de mejora.

2.1.1. Pasos del procesamiento de imágenes

[Khirade y Patil, 2015] discuten en su *paper* métodos para detectar enfermedades en plantas procesando imágenes de sus hojas, describiendo los pasos del proceso uno por uno. Estos pasos consisten en:

- **Adquisición de la imagen:** Se capturan imágenes de la hoja de la planta a través de una cámara, generando una imagen de color en formato RGB (rojo, verde y azul). Se crea una estructura para la transformación del color, y luego se aplica una transformación del espacio de colores independiente del dispositivo a la estructura.
- **Pre-procesamiento de la imagen:** Para remover ruidos u otros objetos no deseables de la imagen se ejecutan distintas técnicas de pre-procesamiento. Entre ellas están el recorte de la imagen para obtener la región de interés, suavizado de la imagen mediante el uso de filtros, incremento del contraste, ecualización del histograma, entre otras. También es posible que se alteren las dimensiones de la imagen para asegurar que todas las imágenes usadas tengan las mismas dimensiones.
- **Segmentación de la imagen:** La segmentación de la imagen se realiza para separar la imagen en distintas secciones que posean las mismas características o algún grado de similitud. Existen variados métodos para realizar este proceso, entre los cuales se encuentran los siguientes:

- **Algoritmo de detección de bordes:** La imagen en formato RGB se convierte al modelo HSI (matiz, saturación, intensidad) previamente. Este algoritmo ayuda a encontrar la parte infectada de la hoja al delimitar los bordes.
 - **K-means clustering:** Este algoritmo se utiliza para la clasificación de objetos basándose en un conjunto de características que pertenecen a un número K de clases. La categorización de un objeto se realiza al minimizar la suma de los cuadrados de la distancia entre el objeto y el *cluster* correspondiente.
 - **Método de Otsu:** Crea imágenes binarias a partir de imágenes en escala de grises en donde todos los píxeles bajo cierto límite corresponden a 0 y todos los píxeles mayores a este a 1.
- **Extracción de características de la imagen:** La extracción de características es un proceso esencial para identificar y clasificar un objeto. Algunas de las características que pueden ser consideradas para la detección de enfermedades en plantas son el color, la textura, la morfología y los bordes. Métodos usados para el proceso son:
- **Método de co-ocurrencia del color:** La imagen en formato RGB es convertida a HSI, y se hace uso del color y textura para obtener las características de la imagen.
 - **Extracción del color de la hoja usando componentes H y B:** Se procesa la imagen usando técnica de difusión anisotrópica para preservar la información de los píxeles afectados. Se utilizan los componentes H (*Hue*, matiz) y B (azul a amarillo del espacio de color LAB) para distinguir la hoja de otras partes.
- **Detección y clasificación de la enfermedad de la planta:** Es el proceso en donde se determina si una planta está enferma en base a la imagen de su hoja, y qué enfermedad padece. Existen múltiples técnicas para la clasificación, de las cuales dos fueron descritas en el trabajo anteriormente mencionado:
- **Redes neuronales artificiales:** Tras la extracción de características, las imágenes se clasifican según el estado (sana o enferma) y/o la enfermedad presente usando redes neuronales.
 - **Back Propagation:** Algoritmo de redes neuronales en el que, una vez entrenado, los pesos de la red neuronal son fijos y pueden usarse para clasificar imágenes no presentes en el conjunto de aprendizaje.

El proceso de detección de enfermedades en plantas utilizando procesamiento de imágenes en grandes rasgos se resume en la Figura 3.

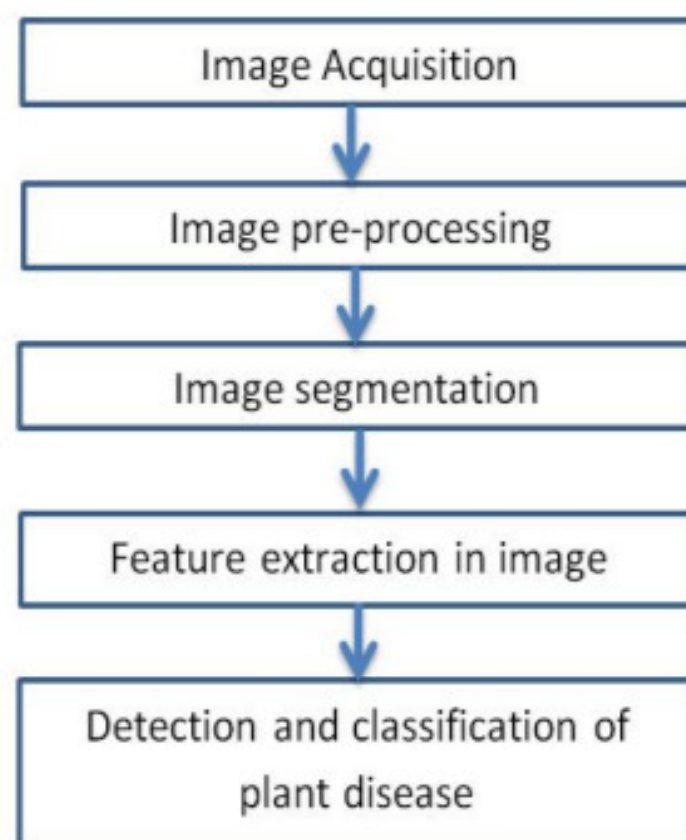


Figura 3: Pasos para el procesamiento de imágenes (inglés).
Fuente: [Khirade y Patil, 2015]

2.1.2. Revisión de la literatura

En *Plant Disease Detection Techniques: A Review*, [Sandhu y Kaur, 2019] hacen una revisión de la literatura, identificando distintos métodos y algoritmos utilizados para el reconocimiento de enfermedades en plantas, listando la efectividad de cada uno y comparando resultados. Los trabajos analizados comprenden distintos tipos de cultivos y enfermedades. Las técnicas analizadas y comparadas en su revisión de la literatura se resumirán a continuación:

- [Al Bashish *et al.*, 2010]: La partición de la imagen de una hoja se hace en 4 *clusters* utilizando *k-means clustering* bajo el supuesto de que uno de estos *clusters* contiene píxeles asociados a la enfermedad. Se usa el método *Color Co-occurrence* para la extracción de características de color y textura. La clasificación se realiza con redes neuronales usando metodologías basadas en *Back Propagation*. El programa resultante es probado para detectar cinco tipos diferentes de enfermedades, y la exactitud promedio del sistema es de 93 %.
- [Bhange y Hingoliwala, 2015]: Extracción de características en base al color, morfología y usando *Color Coherence Vector*. Partición de imagen de la fruta (granada en este caso) utilizando *k-means clustering*. Entrenamiento y clasificación usando algoritmo *Support Vector Machine* (SVM). La exactitud del sistema usando SVM para la clasificación es de 82 % en promedio.
- [Pujari *et al.*, 2015]: Se detectan enfermedades causadas por hongos y las técnicas

usadas varían según el tipo de cultivo a ser analizado, los cuales son frutales, vegetales, comerciales y cereales.

- **Cultivos frutales:** Se utiliza *k-means clustering* en la partición, y el algoritmo *Nearest Neighbor* y una *Artificial Neural Network* para la extracción de características y clasificación de la enfermedad presente, alcanzando una exactitud promedio sobre 90 %.
 - **Cultivos vegetales:** La partición se realizó con segmentación Chan-Vese, la extracción de características con *Local Binary Patterns* (LBP) y la clasificación con SVM y *k-nearest neighbor* (KNN), alcanzando una exactitud promedio de casi 88 %.
 - **Cultivos comerciales:** La partición es realizada con segmentación *grab-cut*, la extracción de características se basa en una solución *Wavelet* y la clasificación se realiza vía distancia de Mahalanobis *Probabilistic Neural Network*, resultando en una exactitud promedio de 85 %.
 - **Cultivos cereales:** Se realiza la partición usando *k-means clustering* y *Canny Edge Detector*, la extracción de características utilizando el color, textura, forma, textura del color y la transformada de Radón, y la clasificación con SVM y *Nearest Neighbor Classifier*, alcanzando una exactitud general del 83 %.
- [Singh y Misra, 2017]: Se utilizan hojas de cuatro plantas para las pruebas, las cuales son plátano, frijol, limón y rosas. Se propone el Algoritmo Genético (GA) como técnica para realizar el proceso de segmentación, separando puntos sin etiquetar de N dimensiones en K clusters. Para el proceso de extracción de características se hace uso del método de co-ocurrencia del color considerando el color y la textura. La clasificación se realiza usando *Minimum Distance Criterion* junto con partición *k-means clustering* y con SVM, alcanzando una exactitud del 86,64 % y 95,71 % respectivamente. El uso de *Minimum Distance Criterion* con partición vía GA en vez de *k-means clustering* aumenta la exactitud al 93,63 %.
 - [Kiani y Mamedov, 2017]: Se hace uso de un *Fuzzy Decision Maker* para clasificar las enfermedades de una hoja enferma, recibiendo cinco parámetros de entrada y dos de salida. Los parámetros de entrada son dos asociados a deficiencias de hierro y tres a infecciones fungales, mientras que los parámetros de salida corresponden a las dos enfermedades que pueden ser detectadas por el programa. Se logra un 97 % de exactitud para la detección y segmentación de enfermedades, y el tiempo empleado en la detección promedia 1,2 segundos.
 - [Ali et al., 2017]: La segmentación se realiza usando un algoritmo basado en ΔE Color Difference. La extracción de características es hecha utilizando LBP, histogramas RGB e histogramas HSV. Para la clasificación se utilizan KNN, *Cubic SVM*, *Boosted Tree* y *Bagged Tree*, de los cuales el último alcanza los mejores resultados, logrando exactitudes de 99,5 % con histogramas RGB, 100 % con histogramas HSV y 100 % con LBP. Las exactitudes promedio obtenidas para KNN, *Cubic SVM* y *Boosted Tree* son 88,9 %, 90,1 % y 50,9 % respectivamente.

- [Saradhambal *et al.*, 2018]: La segmentación de la imagen se realiza usando *k-means clustering* y el método de Otsu. Para la extracción de características se utilizan textura, que incluye el área, largo de los ejes de color, excentricidad, solidez y perímetro, y forma, que comprende el contraste, correlación, energía, homogeneidad y media. La clasificación se realiza usando un clasificador basado en redes neuronales.

[Bankar *et al.*, 2014] proponen un sistema sencillo para el proceso de la detección de enfermedades en plantas. Los métodos a utilizar se dividen en dos fases:

- **Fase I:** La primera fase consiste en elegir imágenes RGB de hojas de plantas, tanto sanas como enfermas. Luego estas imágenes son procesadas para convertirlas a escala de grises y obtener las imágenes con los canales de color rojo, azul y verde de la imagen original. Se utiliza la técnica de Canny para la detección de bordes en la imagen en escala de grises, pues no se puede aplicar a imágenes RGB. El proceso de entrenamiento inicia en esta fase.
- **Fase II:** Se elige una imagen de hoja para la prueba. La imagen es procesada de igual manera que en la primera fase. Una vez que el proceso de entrenamiento de la primera fase concluye se generan histogramas para hojas sanas y para hojas enfermas, los cuales serán comparados con el histograma de la imagen de prueba para determinar si es que esta corresponde a una hoja sana o enferma.

Además de describir los procesos de un sistema para la detección de enfermedades de plantas en imágenes de hojas, los autores también detallan los algoritmos de los pasos a seguir para la extracción de características, los cuales son la obtención del área de la hoja, conversión de la imagen a escala de grises, la detección de bordes usando la técnica de Canny, obtención del histograma de bordes y obtención del histograma de colores. Finalmente los autores definen el algoritmo final para el funcionamiento del sistema propuesto.

En su estudio [Ramesh *et al.*, 2018] utiliza *Machine Learning* con *Random Forest* (RF) para la clasificación. *Random Forest* construye un "bosque" de árboles de decisión durante el periodo de entrenamiento que pueden ser usados para tareas de clasificación y regresión. *Histogram of Oriented Gradients* (HOG) se emplea para la extracción de características de los objetos, y en este caso se hace uso de tres descriptores de componentes: *Hu Moments* para extraer la forma de la hoja, textura de *Haralick* para obtener la textura y el color de la hoja, y finalmente el histograma de color para representar la distribución de colores en la imagen. El entrenamiento se realiza extrayendo las características con HOG, añadiendo el resultado al conjunto de entrenamiento, y tras ejecutar el entrenamiento se almacena el modelo RF resultante. Para la clasificación se extraen las características con HOG, se carga el modelo RF, se clasifica de acuerdo al modelo y se muestra el resultado en pantalla. Con los conjuntos de entrenamiento y prueba usados, los cuales constaban de imágenes de hojas de papayos, la exactitud alcanzada por el modelo RF es del 70,14 %.

[Liu *et al.*, 2020] utilizan un modelo basado en redes neuronales convolucionales profundas para identificar 6 enfermedades y plagas diferentes en vides. Para aliviar el problema de que el conjunto de datos de entrenamiento no es suficientemente grande se aplicaron técnicas de aumentación de datos para derivar 13 variantes por imagen, generando un *dataset* de 107.366 imágenes de hojas de vides a partir de las 4.023 imágenes recolectadas en terreno y 3.646 de *datasets* públicos, la exactitud en un 14,42 % comparado al resultado obtenido al entrenar el modelo usando solo los datos originales. El modelo propuesto corresponde a *Dense Inception Convolutional Neural Network* (DICNN), que usa convoluciones separables profundas para armar las dos primeras capas convolucionales con el fin de reducir el número de parámetros y evitar *overfitting*⁵. Luego, se hizo uso de la estructura del modelo Inception para mejorar el rendimiento de la extracción de características, y finalmente se aplica la estrategia de conexiones densas a las 4 estructuras cascada de Inception para aliviar el problema del desvanecimiento y explosión del gradiente, promoviendo la propagación de características y el reuso. El modelo resultante logra una exactitud de 97,22 % en las pruebas, mejorando los resultados de GoogLeNet y ResNet-34, modelos altamente competitivos, en un 2,97 % y 2,55 % respectivamente.

2.2. Deep Learning

Deep Learning (DL) es una subcategoría de *Machine Learning*. DL permite a modelos computacionales compuestos por al menos tres capas de procesamiento aprender representaciones de datos con diferentes niveles de abstracción. El aprendizaje en DL permite descubrir patrones complejos en conjuntos de datos de gran tamaño utilizando el algoritmo *Backpropagation*, refinando los parámetros internos que controlan cómo se computan las representaciones en cada capa utilizando la representación en la capa previa, ver Figura 4 [LeCun *et al.*, 2015].

En el proceso de la creación de un modelo de *Deep Learning*, y por extensión, de *Machine Learning*, se necesita una gran cantidad de experiencia pasada en forma de datos recolectados etiquetados para el aprendizaje supervisado; en caso de ser no supervisado los datos no poseen etiquetas. En el caso de clasificación de objetos usando imágenes, los datos corresponden a imágenes etiquetadas según la clase a la que correspondan.

La mayor parte de los datos recopilados se utilizan como **conjunto de entrenamiento** y **conjunto de validación**. En el entrenamiento, el modelo es alimentado con una imagen del conjunto de entrenamiento y produce un vector de puntajes de clase. La clase a la que corresponde el objeto debería tener el puntaje más alto, pero esto no suele ocurrir inicialmente. *Backpropagation* es utilizado para ajustar los parámetros de la clasificación y reducir el error en las clasificaciones. Este proceso de entrenamiento ocurre en varias iteraciones o *epochs* (épocas). El **conjunto de validación** contiene datos que el modelo no ha utilizado en el en-

⁵El *overfitting* ocurre cuando un modelo es sobre-entrenado, lo que causa que éste prediga observaciones correctamente durante el entrenamiento pero falle al evaluar datos nuevos.

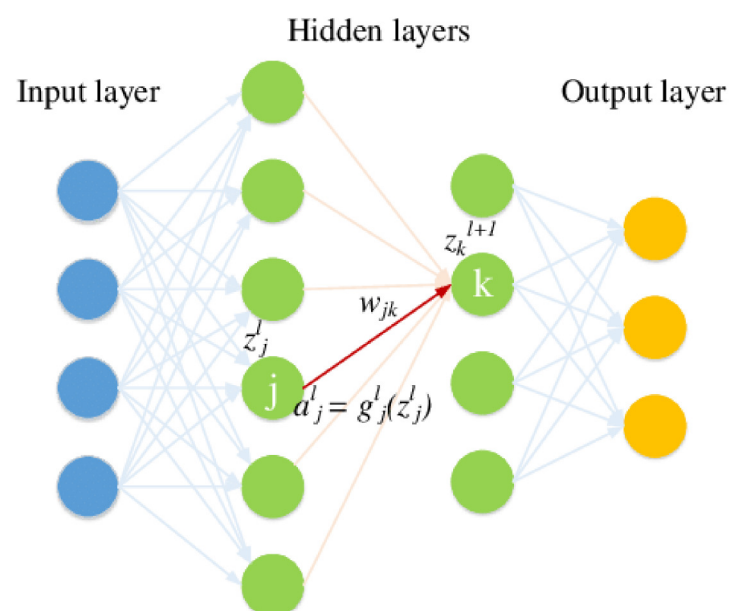


Figura 4: Una red neuronal profunda sencilla con dos capas ocultas y una capa de *output*.
Fuente: [Oroojlooyjadid *et al.*, 2020]

trenamiento y se utiliza para evaluar la habilidad de generalización del modelo; la iteración que presente el menor error de validación es considerada la mejor versión del modelo entrenado.

Una vez finalizado el proceso de entrenamiento se utiliza el **conjunto de prueba** para evaluar la habilidad de generalización, es decir, la capacidad de clasificar correctamente instancias totalmente nuevas, del mejor modelo de acuerdo a la validación. El conjunto de prueba consiste de datos del conjunto original que no han sido usados en el entrenamiento o la validación. El propósito de las pruebas es dar una idea de cuál sería el error esperado del modelo en situaciones reales [Alpaydin, 2020].

Deep Learning ha mejorado dramáticamente el estado del arte en áreas como reconocimiento de voz, clasificación y detección de objetos, entre otras, por lo que ha sido muy utilizado en el último tiempo para la clasificación de enfermedades en plantas en la literatura científica, pues resulta ser un área que muestra gran potencial para alcanzar más altos grados de exactitud en las clasificaciones en comparación a los métodos tradicionales [Saleem *et al.*, 2019].

Los pasos de una implementación típica de DL aplicada a la clasificación de enfermedades en plantas se describen a continuación. En resumen, el primer paso consiste en **establecer el dataset**, luego separándose éste en **conjuntos de entrenamiento y de validación**, que suelen distribuirse el 80 % y 20 % del total respectivamente. Posteriormente se escogen las **arquitecturas de Deep Learning** a utilizar, se **entrenan** los modelos anteriormente definidos y se evalúa su desempeño, graficando los resultados para visualizar la significancia de los modelos. Finalmente se usan **métricas del desempeño** de los modelos para la clasificación de las imágenes, que en este caso corresponderían al tipo de enfermedad presente en la planta, y técnicas de visualización son utilizadas para detectar, localizar o clasificar las imágenes. Este proceso se puede visualizar en la Figura 5.

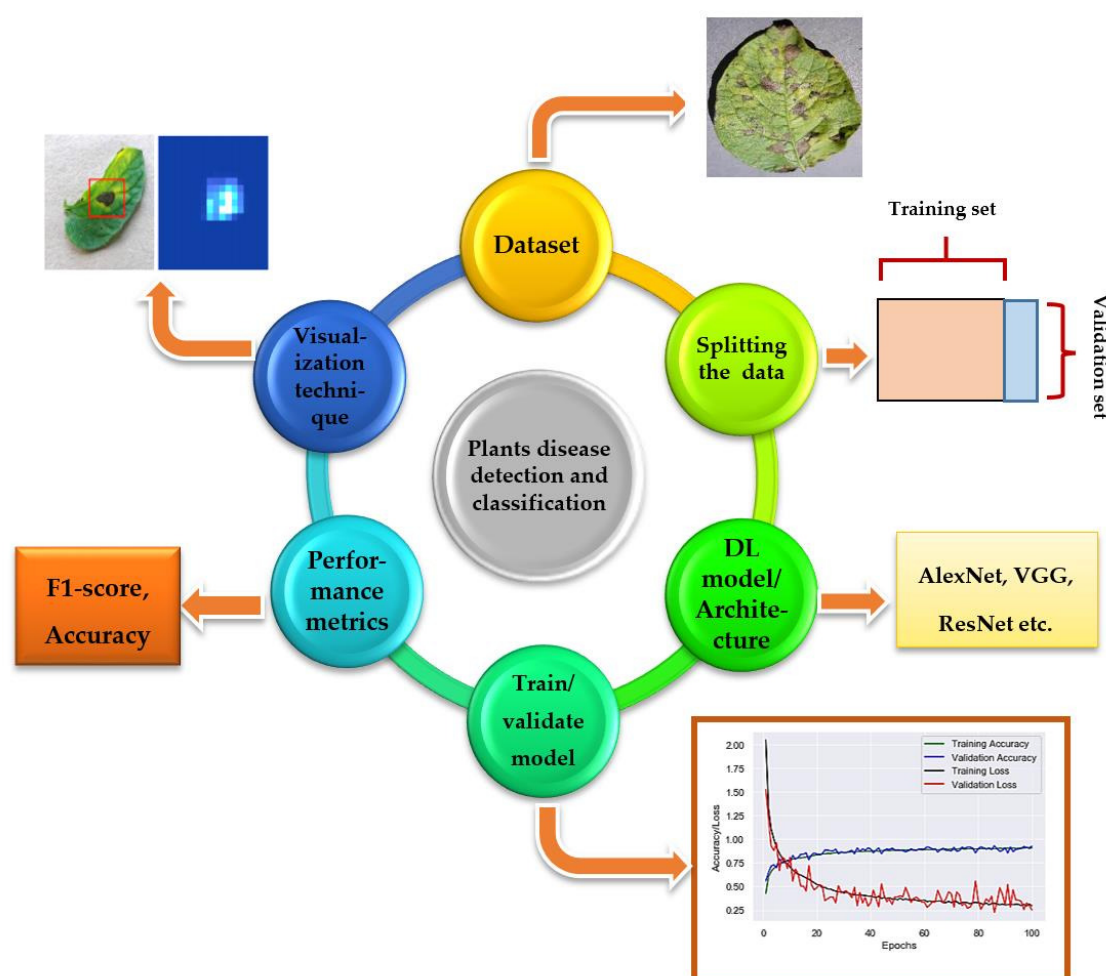


Figura 5: Diagrama de flujo para la implementación de DL aplicado a la detección y clasificación de enfermedades en plantas (inglés)

Fuente: [Saleem *et al.*, 2019]

2.2.1. ImageNet Large Scale Visual Recognition Challenge

ImageNet Large Scale Visual Recognition Challenge (ILSVRC)⁶ es una competencia que evalúa algoritmos para detección de objetos y clasificación de imágenes a larga escala. Tiene los objetivos de permitir a los investigadores comparar el progreso en los algoritmos de detección y clasificación y fomentar el desarrollo de nuevas y mejoradas técnicas para la visión computacional. El *dataset* utilizado en la competencia es un subconjunto del *dataset* ImageNet [Deng *et al.*, 2009] que abarca 1000 clases de objetos y contiene 1.281.167 imágenes de entrenamiento, 50.000 de validación y 100.000 de prueba.

En el año 2012 [Krizhevsky *et al.*, 2012] proponen el modelo **AlexNet**, una CNN profunda de gran tamaño que tiene 60 millones de parámetros y 650.000 neuronas. Una variante de este modelo fue registrada en la competencia ILSVRC de 2012, alcanzando un error *top-5*⁷ de prueba de 15,3 %, muy por delante del 26,2 % alcanzado por el segundo mejor modelo participante. Este resultado evidenció el potencial de las CNNs para resolver problemas de clasificación de imágenes, causando que las CNN se volvieran la técnica principal para

⁶ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

⁷El ratio de error *top-5* corresponde a la fracción de imágenes de prueba en las que la etiqueta correcta no está presente en las cinco etiquetas consideradas más probables por el modelo. En el caso de la exactitud *top-5*, la etiqueta correcta está presente dentro de las cinco etiquetas consideradas más probables.

la resolución de estos e iniciando un auge no sólo de las CNNs, sino del *Deep Learning* en general.

2.2.2. Redes neuronales convolucionales

Las **redes neuronales convolucionales**, *Convolutional Neural Networks* (CNN) en inglés, son redes neuronales de tipo *feedforward* (es decir, la información se mueve en solo una dirección) inspiradas en la percepción visual de los organismos vivos, representando las neuronas biológicas como neuronas artificiales. Las neuronas que componen una CNN tienen la capacidad de optimizarse a sí mismas a través del aprendizaje utilizando *back-propagation* en el entrenamiento, al igual que en las redes neuronales artificiales (ANNs). Las CNNs tienen la capacidad de **extraer características** relevantes de los datos utilizando estructuras de convolución, lo que significa que a diferencia de los métodos tradicionales, no se necesita extraer las características manualmente [Li et al., 2021].

Las CNNs son unas de las redes más importantes y estudiadas en el campo de DL, logrando resultados notables en áreas como la visión por computador y procesamiento del lenguaje. Éstas surgen como una simplificación de las ANNs con tal de adaptarlas a tareas relacionadas a imágenes, pues la complejidad computacional aumenta significativamente al incrementar las dimensiones de las imágenes de entrada, y simplificar el modelo ayuda a evitar el *overfitting*. Las ANN logran resolver apropiadamente problemas de clasificación de datasets con imágenes de dimensiones pequeñas, como por ejemplo MNIST, el cual tiene imágenes de 28x28 píxeles normalizados a valores blancos y negros. En este caso, una neurona de la primera capa oculta tendrá 784 pesos, pero si se necesita utilizar un dataset con imágenes RGB de tamaño 64x64, el número de pesos de esta neurona aumenta a 12.288, lo que evidencia que la red necesitará ser mucho más grande para manejar el problema y por tanto no es una opción adecuada.

Una CNN está compuesta por tres tipos de capas: capas **convolucionales**, capas de **pooling**, y capas **fully-connected o densas**. Cada capa contiene neuronas organizadas en tres dimensiones: alto, ancho y profundidad. Esta última no corresponde al número total de capas, sino a la dimensión de un volumen de activación. A diferencia de las ANN, las neuronas presentes en una capa de una CNN solo se conectan a una región pequeña de la capa que les precede. Esta arquitectura puede visualizarse en la Figura 6.

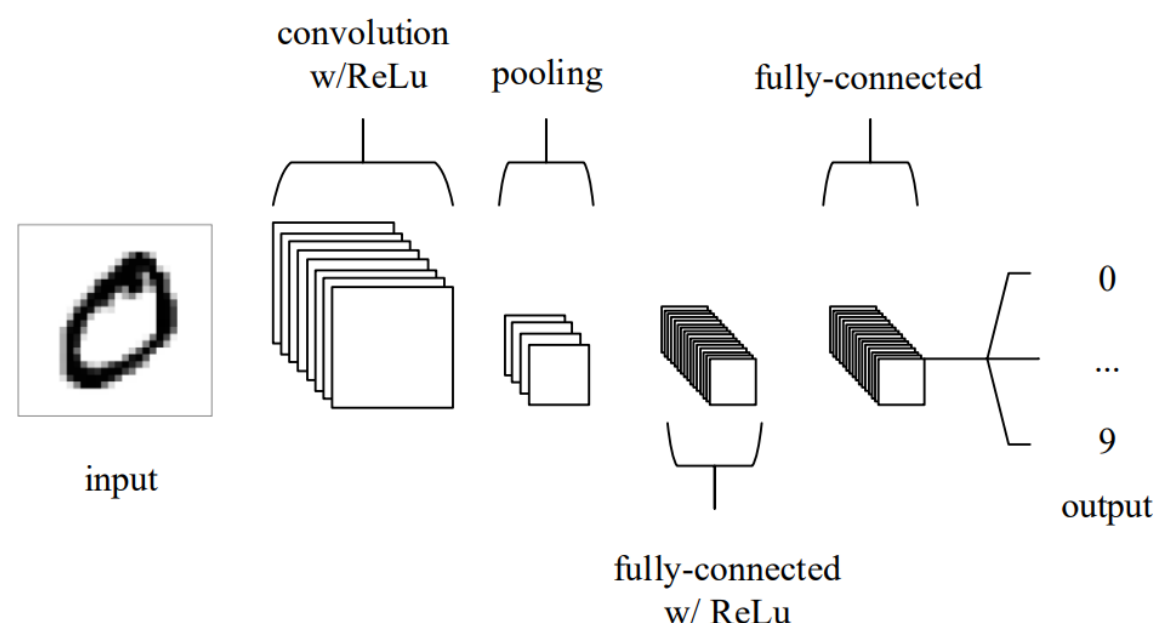


Figura 6: Arquitectura simplificada de una CNN con 5 capas para la clasificación del *dataset* MNIST

Fuente: [O'Shea y Nash, 2015]

Las **capas convolucionales** se encargan de determinar el *output* de las neuronas conectadas a regiones locales del *input* (extracción de características) y de reducir la complejidad del modelo, las **capas de pooling** buscan reducir el número de parámetros de una activación y las **capas densas**, que están conectadas directamente a las neuronas de sus capas adyacentes, producen un puntaje por cada clase con el fin de que estos se usen para la clasificación [O'Shea y Nash, 2015].

La **capa convolucional** juega un rol vital en la extracción de características. Esta capa hace uso de **kernels**, receptores de las neuronas que responden a diferentes características del *input* a medida que operan sobre éste, los cuales usualmente tienen dimensiones espaciales pequeñas pero cubren la profundidad de la entrada completamente en su recorrido.

La **convolución** es un tipo especializado de operación lineal usada para la extracción de características en donde un *kernel*, arreglo pequeño de números, se aplica a lo largo de una entrada, que es un arreglo de números conocido como *tensor* [Yamashita *et al.*, 2018]. Cuando los datos llegan a una capa convolucional, se convoluciona a lo largo de la dimensionalidad espacial de la entrada, filtrando la imagen para luego detectar una característica particular usando la función de activación ReLu (*rectified linear unit*), finalmente utilizando *maximum pooling* para condensar y enfatizar las características. El resultado es un mapa de activaciones 2D llamado **feature map**; estos mapas pueden ser visualizados como se aprecia en la Figura 7. A medida que se recorre la entrada, se calcula el producto escalar para cada valor en el *kernel* y el *input*. A partir de esto los *kernels* de la red aprenderán a activarse cuando encuentran características específicas en una posición dada del *tensor* de entrada.

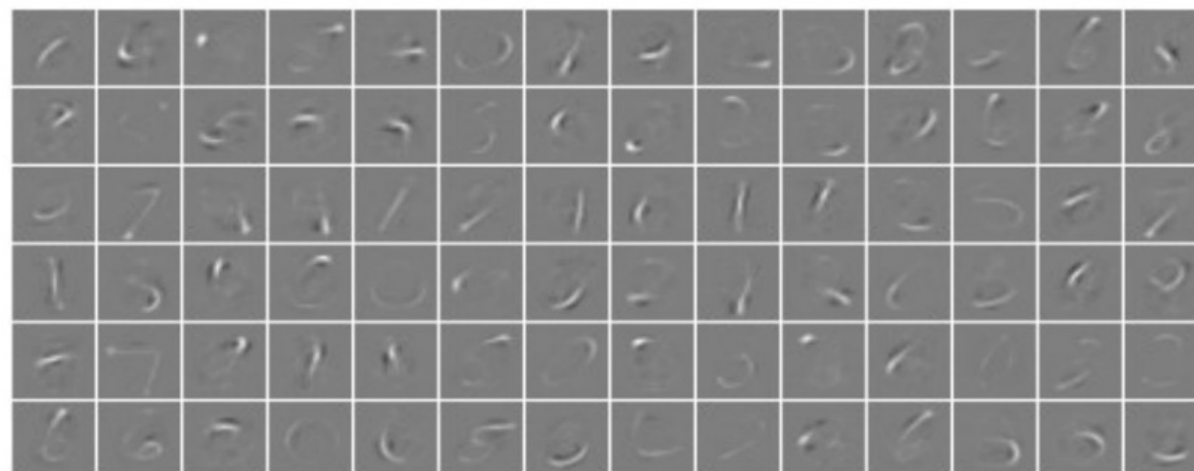


Figura 7: Activaciones de la primera capa convolucional de una CNN simple tras entrenarla en el *dataset* MNIST de dígitos manuscritos. Es posible observar que la red ha logrado extraer características únicas a dígitos específicos.

Fuente: [O'Shea y Nash, 2015]

Al asignar a un *kernel* de convolución ciertos tamaños, la información en el borde del vector de entrada puede quedar subrepresentada, por lo que se agrega *padding* con valores cero, así ajustando la dimensionalidad del *output*. Para controlar la densidad de las convoluciones se emplea el *stride*, el cual reduce la densidad a medida que se incrementa. El *feature map* resultante de la convolución es propenso a causar *overfitting* si consiste en un alto número de características, por lo que emplea *pooling* para aliviar el problema. El proceso de convolución se visualiza en la figura 8.

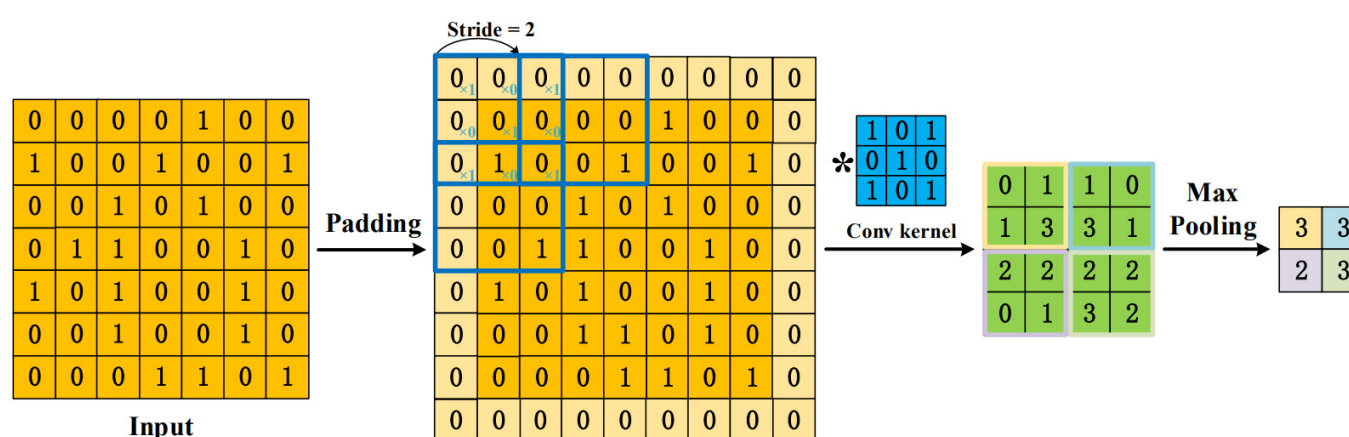


Figura 8: Procedimiento de la capa convolucional de una CNN bidimensional. Inicialmente se introduce *zero-padding*, luego los elementos del *kernel* se multiplican con un segmento de la matriz original, produciendo una nueva matriz. Finalmente se aplica *max-pooling* para obtener un *feature map*.

Fuente: [Li et al., 2021]

Cada **kernel** produce su propio *feature map*, el cual representa por si mismo alguna característica de la entrada. Estos *feature maps* son apilados sobre los mapas de los otros *kernels* a lo largo de la dimensión de profundidad para formar el *output* completo de una capa convolucional, como se puede ver en la Figura 9.

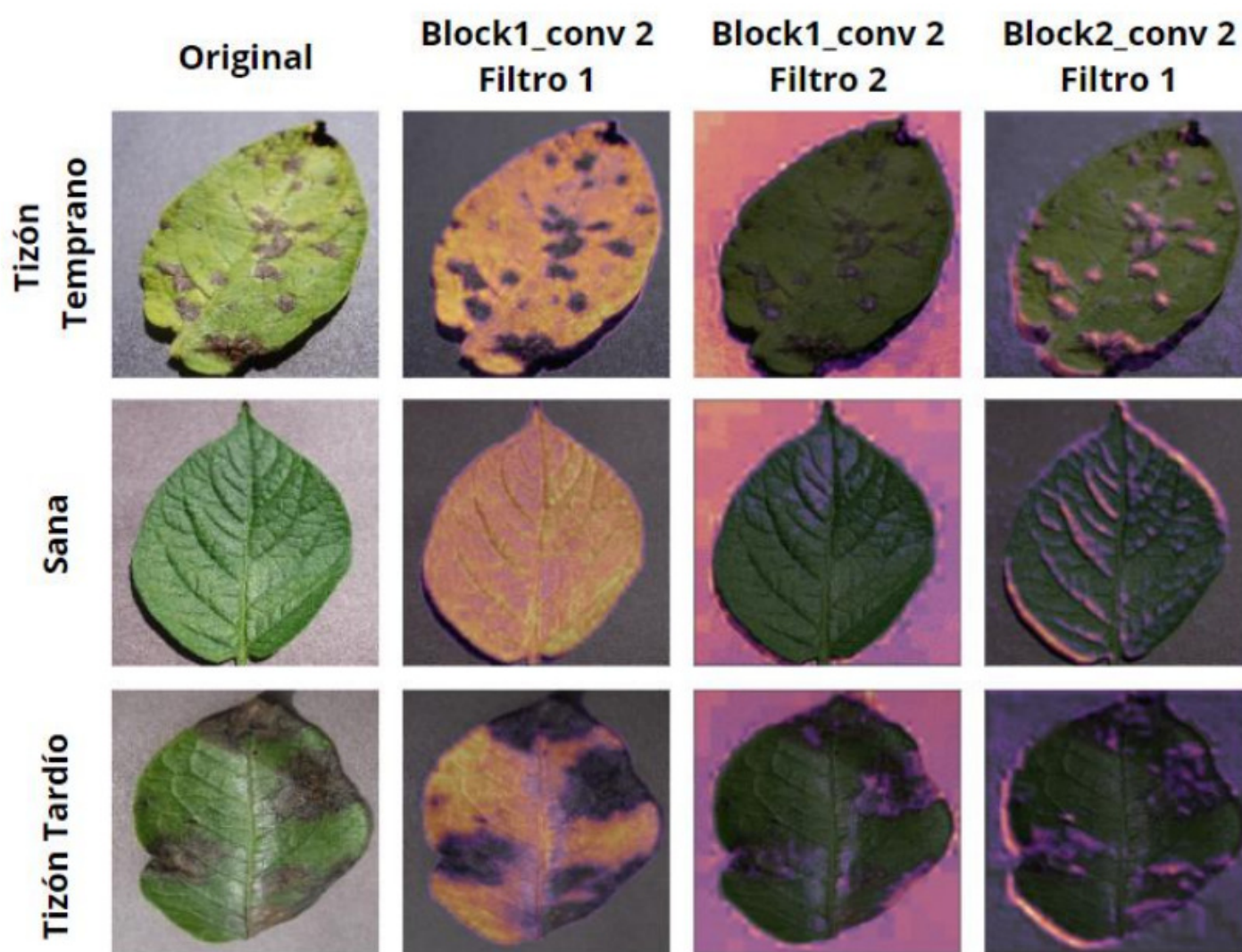


Figura 9: Superposición de los mapas de características de las primeras capas convolucionales de una CNN sobre las imágenes de entrada. Se puede apreciar que cada mapa resalta aspectos distintos de la imagen.

Fuente: [Fuentes, 2021]

A pesar de que la capa convolucional aporta significativamente a reducir la complejidad en comparación a una ANN, aún es necesario crear modelos enormes si se usan imágenes que no sean de dimensiones muy pequeñas como entrada. Para reducir aún más la complejidad se introduce el *parameter sharing*, el cual corresponde a que todas las neuronas de un *feature map* compartan sus pesos. Esto se implementa bajo el supuesto de que si existen características que son útiles en una región, es bastante probable de que también lo sean en otra zona. Esto conlleva a una reducción significativa en la cantidad de parámetros y por tanto una disminución de la complejidad computacional al entrenar el modelo.

2.2.3. Redes neuronales residuales (ResNet)

Las **redes residuales**, más conocidas como **ResNet**, son CNNs propuestas por [He *et al.*, 2016] que incorporan capas residuales para facilitar el entrenamiento de redes neuronales con un alto número de capas. ResNets de profundidades de hasta 152 capas fueron evaluadas en el *dataset* ImageNet, y un conjunto de estas redes logran alcanzar un error *top-5* de 3,57 % en el conjunto de prueba, resultado que ganó el primer lugar en el *ImageNet Large Scale Visual Recognition Challenge* de 2015.

En teoría, aumentar la profundidad de una CNN debería mejorar la capacidad de ésta para detectar características, pero incrementar el número de capas hace surgir el problema del desvanecimiento/explosión del gradiente y aumento del error de entrenamiento. Para solucionar este problema se introduce un *framework* de aprendizaje residual profundo inspirado por la estructura de redes VGG⁸ que hace uso de conexiones atajo (*shortcut connections* en inglés) cada dos capas, como se aprecia en la Figura 10.

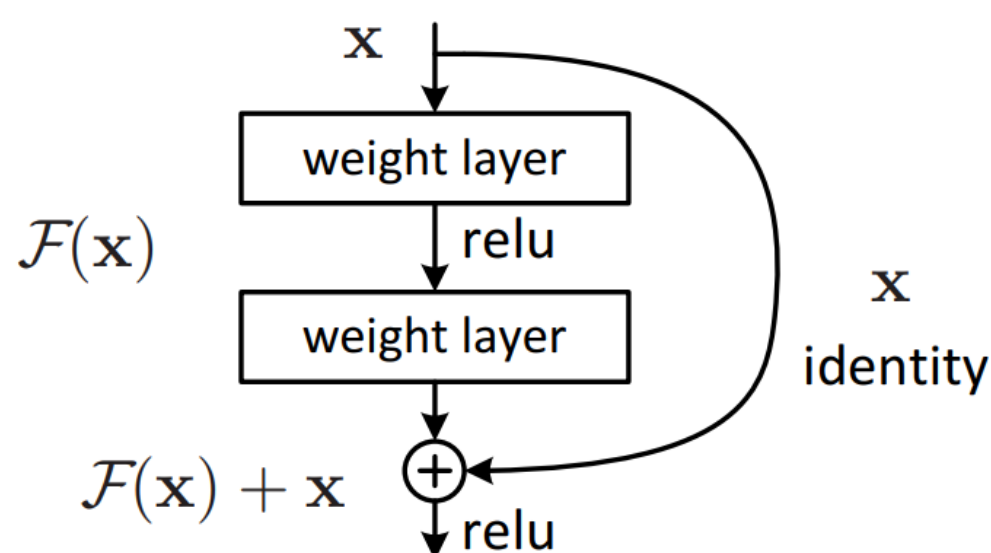


Figura 10: Bloque residual de una ResNet. La conexión atajo en la derecha salta dos capas.
Fuente: [He *et al.*, 2016]

El resultado es un modelo denominado ResNet, el cual presenta menores errores de entrenamiento y prueba al aumentar el número de capas a diferencia de los modelos tradicionales, produce resultados competitivos, es más fácil de entrenar, e incluso demostró que es posible entrenar modelos de más de 1000 capas exitosamente, lo que se anteriormente se consideraba imposible [Li *et al.*, 2016].

⁸VGG (*Visual Geometry Group*) es una arquitectura de CNN propuesta por [Simonyan y Zisserman, 2014], basada en un análisis para incrementar la profundidad de dichas redes. La red utiliza filtros de convolución muy pequeños (3x3) y logra aumentar la profundidad a 16 (VGG-16) y 19 (VGG-19) capas.

2.2.4. EfficientNet

EfficientNet es una CNN y método para escalar uniformemente las dimensiones de profundidad, ancho y resolución de CNNs utilizando conjuntos de coeficientes, propuestos por [Tan y Le, 2019]. El concepto se origina al intentar resolver el problema del escalado de CNNs, proponiendo un método de escalado basado en principios que resulte en mejor exactitud y eficiencia. El resultado corresponde a escalar la profundidad, ancho y resolución de la imagen usando un conjunto de coeficientes de escalado fijos, en comparación al escalado tradicional que usualmente solo escala una de estas tres dimensiones.

El método de escalado de EfficientNet logra funcionar correctamente en modelos existentes, mejorando la exactitud de estos comparados a métodos de escalado unidimensionales. EfficientNet-B0 nace al crear una nueva red base eficiente de bajo peso para la evaluación, escalándola utilizando las técnicas propuestas para obtener una nueva familia de modelos. EfficientNet-B7 logró un 84,3 % de exactitud *top-1* y 97.1 % exactitud *top-5* en la clasificación del conjunto *ImageNet*, el mejor resultado de la competencia, siendo al mismo tiempo 8,4 veces más pequeña y 6,1 veces más rápida en la inferencia que la mejor red convolucional hasta ese momento, GPipe.

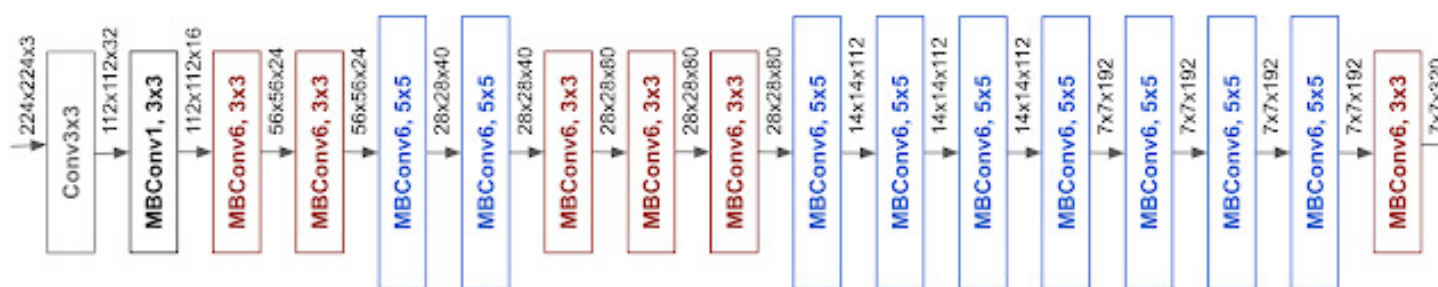


Figura 11: Arquitectura general de EfficientNet-B0.

Fuente: [Mazzeo et al., 2020]

EfficientNet consiste en una familia de ocho modelos que van desde B0 (Figura 11), la red base, hasta B7, la red más grande, de acuerdo al nivel de escalado. Al incrementar el número del modelo, la cantidad de parámetros calculados no crece de manera significativa, pero la exactitud aumenta notablemente. A diferencia de otros modelos de redes neuronales convolucionales, EfficientNet hace uso de una nueva función de activación llamada Swish en vez de la función de activación ReLu.

2.2.5. Métricas de clasificación

Para evaluar el desempeño de los modelos en el proceso de clasificación es necesario establecer **métricas de clasificación**. Las métricas de clasificación pueden ser divididas en tres grupos principales: métricas para clasificadores binarios, métricas para clasificadores multi-clase y métricas para clasificadores multi-etiqueta. Un clasificador binario predice si una muestra pertenece a una de dos posibles clases, como por ejemplo si es que una perso-

na está enferma o no. Un clasificador multi-clase escoge solo una clase de dos o más clases predefinidas; un ejemplo sería si una imagen de entrada es una manzana, naranja o pera. Los clasificadores multi-etiqueta asignan a una entrada una o más clases, por ejemplo, si una imagen contiene un perro y un gato, a esta se le asignan las etiquetas perro y gato simultáneamente [Erickson y Kitamura, 2021].

En un problema de clasificación binaria existen cuatro casos posibles para las predicciones, los cuales se detallan a continuación:

- **True Positives (TP):** Los verdaderos positivos son el número de casos positivos que fueron correctamente clasificados como positivos.
- **True Negatives (TN):** Los verdaderos negativos corresponden al número de casos negativos que fueron correctamente clasificados como negativos.
- **False Positives (FP):** Los falsos positivos se definen como el número de casos negativos que fueron incorrectamente clasificados como positivos. Es también conocido como error tipo 1.
- **False Negatives (FN):** Los falsos negativos corresponden al número de casos positivos que fueron incorrectamente clasificados como negativos. Es también conocido como error tipo 2.

Una **matriz de confusión** es una tabla utilizada para visualizar el rendimiento de un algoritmo de aprendizaje supervisado. En la Figura 12, la cual muestra cómo se estructura la matriz de confusión para un problema de clasificación binaria, las columnas de la matriz representan las predicciones hechas por clase real, mientras que las filas representan la cantidad de predicciones para una clase. En el caso ideal, o sea una clasificación inequívoca, sólo la diagonal contiene valores no cero.

En la Figura 13 se ilustra la matriz de confusión para tres clases. En este caso, las clasificaciones correctas corresponden a los verdaderos positivos TP_S , en donde S es la clase correctamente predicha, y las clasificaciones incorrectas corresponden al error representado como E_{XY} , en donde X es la clase real e Y es la clase erróneamente predicha.

		Valores reales	
		Positivo	Negativo
Valores predichos	Positivo	Verdaderos Positivos/True Positives (TP)	Falsos Positivos/False Positives (FP)
	Negativo	Falsos Negativos/False Negatives (FN)	Verdaderos Negativos/True Negatives (TN)

Figura 12: Matriz de confusión binaria.
Fuente: Elaboración propia.

A continuación se listan las métricas utilizadas para la evaluación de modelos de clasificación binarios junto con sus ecuaciones [Vujović y other, 2021].

- **Exactitud:** Conocida como *accuracy* en inglés, es la fracción de casos que fueron clasificados correctamente. Se calcula como la suma de las predicciones correctas divididas en el total de predicciones. Es una métrica apropiada para evaluar el desempeño si las clases están balanceadas, pues no considera la distribución de las clases.

$$\text{Exactitud} = \frac{TP + FN}{TP + TN + FP + FN} \quad (1)$$

- **Precisión:** Corresponde a la fracción de casos verdaderamente positivos de todos los casos que el modelo clasificó como positivos.

$$\text{Precisión} = \frac{TP}{TP + FP} \quad (2)$$

- **Recall:** También conocido como sensibilidad o *True Positive Rate*, corresponde a la fracción de casos positivos que fueron clasificados como positivos. Se calcula como el número de predicciones positivas correctas dividido por el número total de positivos reales.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

- **F1-score:** Es una medida de la verdadera exactitud del clasificador. Se calcula como la media armónica entre la precisión y *recall*, por lo que representa ambas métricas equitativamente.

$$\text{F1-Score} = \frac{2TP}{2TP + FP + FN} = 2 * \frac{\text{Precisión} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

		Valores reales		
		Clase A	Clase B	Clase C
Valores predichos	Clase A	Verdaderos Positivos A (TP _A)	Error B predicho como A (E _{BA})	Error C predicho como A (E _{CA})
	Clase B	Error A predicho como B (E _{AB})	Verdaderos Positivos B (TP _B)	Error C predicho como B (E _{CB})
	Clase C	Error A predicho como C (E _{AC})	Error B predicho como C (E _{BC})	Verdaderos Positivos C (TP _C)

Figura 13: Matriz de confusión para tres clases.
Fuente: Elaboración propia.

Las fórmulas de las métricas para problemas de clasificación binaria, con excepción de la exactitud, no pueden ser utilizadas para la clasificación multi-clase. Para solucionar este problema se emplean los enfoques macro *average* (promedio en inglés) y micro *average* [Grandini *et al.*, 2020].

El enfoque **macro average** considera todas las **clases** en el cálculo, y cada clase tiene el mismo peso en el promedio, por lo que no hay distinciones entre clases con alta y baja población. Esto significa que las métricas macro pierden representatividad si existe un desbalance notable en la cantidad de observaciones de cada clase. Sin embargo, las métricas macro son apropiadas si es que se quiere tratar a todas las clases equitativamente a pesar del desbalance. En los cálculos de las métricas multi-clase k corresponde a una clase singular, mientras que K es el número total de clases.

- **Precisión Macro:** Corresponde al promedio de la precisión para cada una de las clases.

$$\text{Precisión Macro} = \frac{\sum_{k=1}^K \text{Precisión}_k}{K} \quad (5)$$

- **Recall Macro:** Se define como el *recall* promedio de todas las clases.

$$\text{Recall Macro} = \frac{\sum_{k=1}^K \text{Recall}_k}{K} \quad (6)$$

- **F1-score Macro:** Es el promedio del F1-score de todas las clases. También se puede calcular como la media armónica de la precisión macro y el *recall* macro.

$$\text{F1-score Macro} = \frac{\sum_{k=1}^K \text{F1-score}_k}{K} = 2 * \frac{\text{Precisión Macro} * \text{Recall Macro}}{\text{Precisión Macro} + \text{Recall Macro}} \quad (7)$$

El enfoque **micro average** muestra el desempeño promedio de todas las **observaciones** y cada una de estas tiene el mismo peso. Como consecuencia, las clases con poblaciones más grandes tienen mayor influencia en el resultado. En caso de que el conjunto de datos sea balanceado, los enfoques macro y micro producen resultados similares.

- **Precisión Micro:** Corresponde a la suma de verdaderos positivos para cada clase dividida por la suma de predicciones positivas para todas las clases.

$$\text{Precisión Micro} = \frac{\sum_{k=1}^K TP_k}{\sum_{k=1}^K TP_k + \sum_{k=1}^K FP_k} \quad (8)$$

- **Recall Micro:** Se calcula como la suma de los verdaderos positivos de cada clase dividida por la suma de los positivos para todas las clases.

$$\text{Recall Micro} = \frac{\sum_{k=1}^K TP_k}{\sum_{k=1}^K TP_k + \sum_{k=1}^K FN_k} \quad (9)$$

- **F1-score Micro:** Es la media armónica entre la precisión micro y el *recall* micro.

$$\text{F1-score Macro} = \frac{2 \sum_{k=1}^K TP}{2 \sum_{k=1}^K TP_k + \sum_{k=1}^K FP_k + \sum_{k=1}^K FN_k} \quad (10)$$

$$\text{F1-score Macro} = 2 * \frac{\text{Precisión Micro} * \text{Recall Micro}}{\text{Precision Micro} + \text{Recall Micro}} \quad (11)$$

En la clasificación multi-clase, los resultados al calcular la precisión micro, *recall* micro, F1-score micro y la exactitud son **iguales**. Esta igualdad no se cumple para la clasificación multi-etiqueta.

2.3. Análisis de cultivos de Chile, sus enfermedades y características

Para determinar las plantas y enfermedades que serán sujeto de análisis, y así diseñar una solución que se acomode a sus características se analizarán documentos relacionados a la situación de la agricultura en Chile y sus cultivos más relevantes. También se estudiarán las enfermedades más prevalentes en estas plantaciones y las medidas de prevención y mitigación que toman los agricultores para combatir estos males. El Servicio Agrícola y Ganadero

del Gobierno de Chile (SAG) se encarga de “apoyar el desarrollo de la agricultura, los bosques y la ganadería, a través de la protección y mejoramiento de la salud de los animales y vegetales”. En su sitio web provee información acerca de las plagas y enfermedades presentes, como la identificación, métodos de propagación, prevención y folletos informativos.⁹ Chile Agrícola provee en su sitio web capacitación gratuita para la detección de plagas y enfermedades presentes en los cultivos del país. Entre los recursos que ofrece relacionados al tema se encuentran cursos, vídeos, fichas informativas y manuales.¹⁰

2.3.1. Cultivos Chilenos

Los cultivos no frutales de mayor importancia en Chile son el trigo, avena, maíz, raps y arroz, los cuales representan el 75 % de la superficie total plantada. Por otra parte, la superficie plantada de cultivos frutales es liderada por la uva de mesa, con 14,9 % del total. Los siguientes frutales más relevantes son los nogales con 11,5 %, manzanos con 10,6 %, cerezos con 9,4 % y paltos con 9 % de la superficie cubierta por frutales. Cabe destacar que la superficie total cubierta por frutales aumentó de casi 250.000 hectáreas en 2008 a más de 321.000 hectáreas en 2018, aumentando en un 29 % durante el período.¹¹

2.3.2. Enfermedades relevantes y cultivos afectados

La **moliniasis**, también conocida como pudrición morena/parda de los carozos o tizón de la flor, es una enfermedad grave causada por el hongo *Monilinia fructicola* que afecta principalmente a frutales de carozos como duraznero, nectarín, ciruelo y cerezo, aunque también puede afectar al damasco, almendro y la guinda. Se le considera una de las enfermedades más graves en los carozos pues causa pérdidas no solo de los frutos, sino también por el atizonamiento (necrosis) de las flores y los brotes. Esta enfermedad se caracteriza por causar atizonamiento (necrosis) de flores y brotes con o sin presencia de gomosis (resina), cancro (lesiones en brotes), la muerte de ramas y pudrición de frutos en pre y post cosecha.¹²

El **cancro bacteriano del kiwi**, causada por la bacteria *Pseudomonas syringae* pv. *actinidiae* (PSA), es una enfermedad que causa exudación de savia color naranja, aparición de canchros en troncos, brazos y/o sarmientos que pueden conllevar a la exudación y necrozamiento de la madera, manchas necróticas en hojas, decaimiento de brotes y daño en flores, aunque no todas las plantas pasan por todos los síntomas.¹³ Esta bacteriosis debe ser detectada lo mas tempranamente posible, pues el control de la bacteria es irreversible una vez que ésta haya generado necrosis vascular, decaimiento y muerte de parte o toda la planta, y muchos

⁹SAG | Servicio Agrícola y Ganadero

¹⁰Control de plagas y enfermedades | Chile Agrícola

¹¹Panorama de la agricultura chilena - 2019

¹²Monilinia fructicola | SAG

¹³Bacteriosis del Kiwi (PSA) | SAG

de los síntomas que causa pueden aparecer tras la erradicación de la bacteria si no se hace oportunamente.

La **enfermedad de Sharka** es una enfermedad causada por el *Plum Pox Virus* (PPV) que provoca serios daños en frutales de carozo, siendo los principales afectados los ciruelos, damascos, durazneros y nectarinos. Esta enfermedad causa deformaciones, daño severo a los frutos del carozo, pérdida de la calidad organoléptica y cosmética de la fruta, y pérdidas de rendimiento al causar caída de la fruta en el periodo de pre cosecha.¹⁴ Se caracteriza por causar la aparición de líneas, bandas o anillos cloróticos difusos de forma y tamaño variables en las hojas al inicio de la primavera, protuberancias, depresiones o manchas en los frutos que causan que se vuelvan incomibles y caída prematura de los frutos. El aspecto de los síntomas varía según la especie afectada, siendo más fácil de identificar la enfermedad en algunos frutales que en otros.

El **oídio** es una enfermedad causada por hongos que afecta tanto a plantas ornamentales como a frutales. EL oídio de la vid, causada por el hongo *Ucinula Necator*, es considerada la enfermedad de más alta importancia del cultivo de parras en el norte y centro de Chile. El principal síntoma es la aparición de manchas de color blanco en las hojas, las cuales corresponden a la presencia del hongo. El hongo y sus esporas pueden visualizarse en cualquier tejido verde de la planta, pudiendo afectar a flores, hojas, raquis y frutos, disminuyendo la productividad de la planta y perjudicando la comercialización de los frutos. Las conidias del hongo son muy sensibles a la luz solar directa, por lo que el hongo suele aparecer en la cara inferior de las hojas expuestas al sol y en ambos lados en caso de que las hojas estén en la sombra [Puelles T. y Sepúlveda R., 2012].

2.3.3. Prevención y tratamiento de enfermedades y plagas

Las consecuencias de la pérdida de cultivos por efecto de plagas y enfermedades no son solo visibles en el ámbito económico, sino también en la alimentación, pues la pérdida de cultivos puede causar escasez de alimentos en zonas rurales y de bajos recursos que dependan de la agricultura para sustentarse. Como resultado, es esencial que los agricultores tomen medidas para minimizar el daño que estas plagas pueden causar a sus localidades, idealmente de naturaleza preventiva para así minimizar potenciales pérdidas a un menor costo.

La Asociación Nacional de Fabricantes e Importadores de Productos Fitosanitarios Agrícolas A.G (AFIPA Chile) sugiere a los agricultores practicar el Manejo Integrado de Plagas (MIP), que consta de tres pasos: prevención, observación e intervención. En la etapa de prevención se sugiere la elección de cultivos que sean lo mejor adaptados posible a la región y tipo de suelo donde serán cultivados, manejar cultivos de manera que disminuya la cantidad de alimento disponible para plagas y aumente la cantidad de enemigos naturales para limitar su aparición, y para estos fines aconseja el uso de herramientas como la rotación de cultivos, plantar variedades resistentes a plagas y manejar correctamente los restos de cosechas. En

¹⁴Enfermedad de Sharka (PPV)

el segundo paso, observación, AFIPA recomienda monitorear regularmente a los cultivos, investigar para tomar decisiones informadas en los momentos correctos y tomar medidas de control junto con los agricultores de la localidad si la plaga presente lo requiere. Finalmente se halla la intervención, que tiene el objetivo de reducir la población presente de plagas a niveles económicamente aceptables mediante control químico, cultural, biológico, físico y genético.¹⁵

En adición a las técnicas de prevención generales, existen además medidas específicas a una especie que los agricultores pueden tomar para proteger a sus cultivos y/o mantenerlos productivos en caso de presentar problemas. En el caso de las uvas, para tratar enfermedades del tronco se puede practicar la cirugía de la vid, que corresponde a la remoción de parras infectadas con una motosierra y aprovechar el sistema de raíces saludables para entrenar una nueva enredadera, así reduciendo el efecto de patógenos presentes en los troncos sobre la productividad de la planta, aunque debe considerarse que esta nueva herida también necesita ser protegida de nuevas infecciones.¹⁶ Para combatir el oídio en vides las medidas de prevención recomendadas son establecer los cultivos en sitios asoleados y abiertos, mantener el follaje poco denso y evitar dosis excesivas de nitrógeno, y para tratarlo una vez presente se utilizan azufre y fungicidas orgánicos, asegurándose de lograr una buena cobertura con ellos pues la mayoría de estos compuestos solo tienen efecto cerca del área donde fueron aplicados [Cruz A., 2004].

Es imperante elegir un tratamiento que sea el adecuado para la enfermedad que se desea controlar y tener cuidado de que las condiciones de aplicación sean favorables, pues no todos los métodos son efectivos para una enfermedad dada y en algunos casos pueden ser totalmente ineficaces y/o dañar a la planta. El momento de la aplicación del tratamiento es también de suma importancia, ya que muchos de los compuestos varían en efectividad según el estado de la planta y también pueden afectar negativamente el valor comercial o calidad de los frutos. Por ejemplo, en la utilización de azufre para el tratamiento del oídio en vides el momento óptimo para la aplicación es al inicio de la temporada, pues el micelio del hongo está más nuevo y la menor densidad del follaje facilita la penetración en las hojas. Algunos de las desventajas de la aplicación del azufre surgen al ser aplicado bajo condiciones no adecuadas; el azufre en polvo puede provocar quemaduras si la temperatura excede los 30°C durante tres días posteriores al tratamiento, y si no se suspende su uso 70 a 80 días antes de la cosecha este podría potencialmente empeorar el sabor de los vinos producidos con los frutos cosechados [Cruz A., 2004].

¹⁵ AFIPA Chile (consultado 11/07/2023)

¹⁶ El manejo de las enfermedades del tronco de vid varía por la región | *Good Fruit Grower* en español (consultado 02/07/2023)

CAPÍTULO 3

PROPUESTA DE SOLUCIÓN

En este capítulo se describe la preparación y el proceso realizado para desarrollar los modelos de *Deep Learning* para la detección de enfermedades en plantas de uvas.

3.1. Entorno de desarrollo

El desarrollo de este trabajo se realizó en la plataforma Google Colaboratory¹⁷. Google Colab es una plataforma de Google que permite escribir y ejecutar código Python en un navegador. El principal beneficio de utilizar Google Colab es el acceso a hardware especializado para el desarrollo y ejecución de algoritmos de *machine learning*. Se hace uso del plan gratuito con GPUs del tipo T4.

El lenguaje de programación utilizado es Python versión 3.10, y las librerías de Python más importantes del desarrollo de la solución fueron Tensorflow version 2.12.0 y Keras versión 2.12.0. Los gráficos se elaboraron utilizando las librerías Matplotlib versión 3.7.1 y Seaborn versión 0.12.2.

3.2. Obtención de datos para el entrenamiento

Para entrenar modelos de *Deep Learning* se hace uso del conjunto de datos público **Grape Disease**¹⁸, el cual tiene origen en el subconjunto de uvas del *dataset Plant Village* [Hughes *et al.*, 2015]. *Plant Village* contiene 54.309 imágenes de hojas provenientes de plantas de manzana, arándano, cereza, maíz, uvas, naranja, durazno, morrón, papa, frambuesa, soja, calabaza, frutilla y tomate. Cada cultivo posee una categoría de plantas sanas y una o más categorías de plantas enfermas. Todas las imágenes del *dataset* fueron capturadas por con una cámara Sony DSC - Rx100/13 de 20.2 megapíxeles en estaciones de investigación experimental asociadas con Universidades *Land Grant* en los Estados Unidos. Para capturar las imágenes se recolectaron hojas removidas de la planta en cuestión, luego ubicando las hojas sobre papel blanco o gris el cual actúa como fondo. Todas las imágenes fueron obtenidas en exterior.

Grape Disease contiene 9030 fotografías de hojas de parras; algunas de éstas presentan aumentación de datos mediante rotaciones con el fin de balancear las clases. El *dataset* se encuentra dividido en un conjunto de entrenamiento de 7022 imágenes y un conjunto de prueba de 1808 imágenes. Las imágenes son de formato jpg, en RGB y de dimensiones 256x256

¹⁷Google Colab | Welcome To Colaboratory

¹⁸Kaggle | Grape Disease

píxeles. En el *dataset* las imágenes están clasificadas en 4 clases: hojas sanas, **Healthy**; la podredumbre negra o pudrición negra de la vid, **Black rot**; la esca de la vid **Esca (Black Measles)** y la Isariopsis clavispora, **Leaf Blight (Isariopsis Leaf Spot)**. La cantidad de muestras disponibles en los conjuntos de entrenamiento y de prueba se pueden apreciar en la Figura 14.

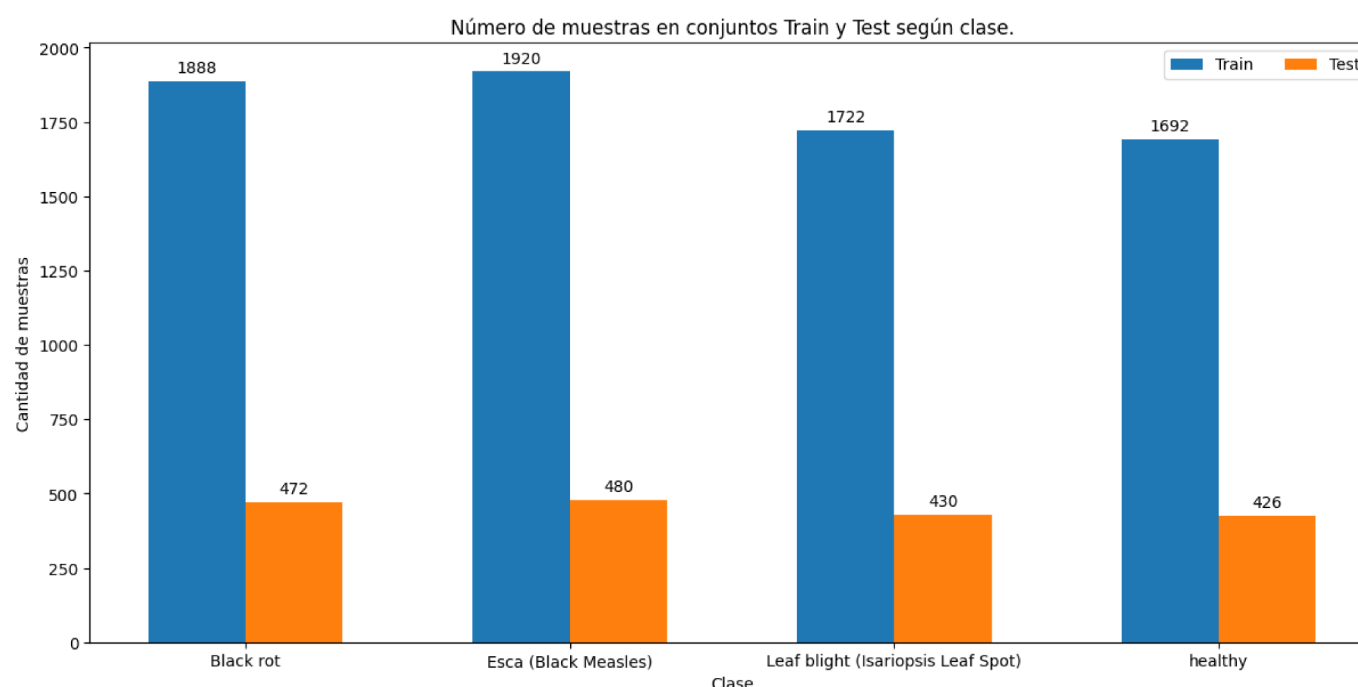


Figura 14: Cantidad de muestras en los conjuntos de entrenamiento y prueba.
Fuente: Elaboración propia.

No hay clases que presenten un número de muestras notablemente inferior o superior a otras; el *dataset* se encuentra balanceado. La proporción de cada clase se puede visualizar mejor en la Figura 15.

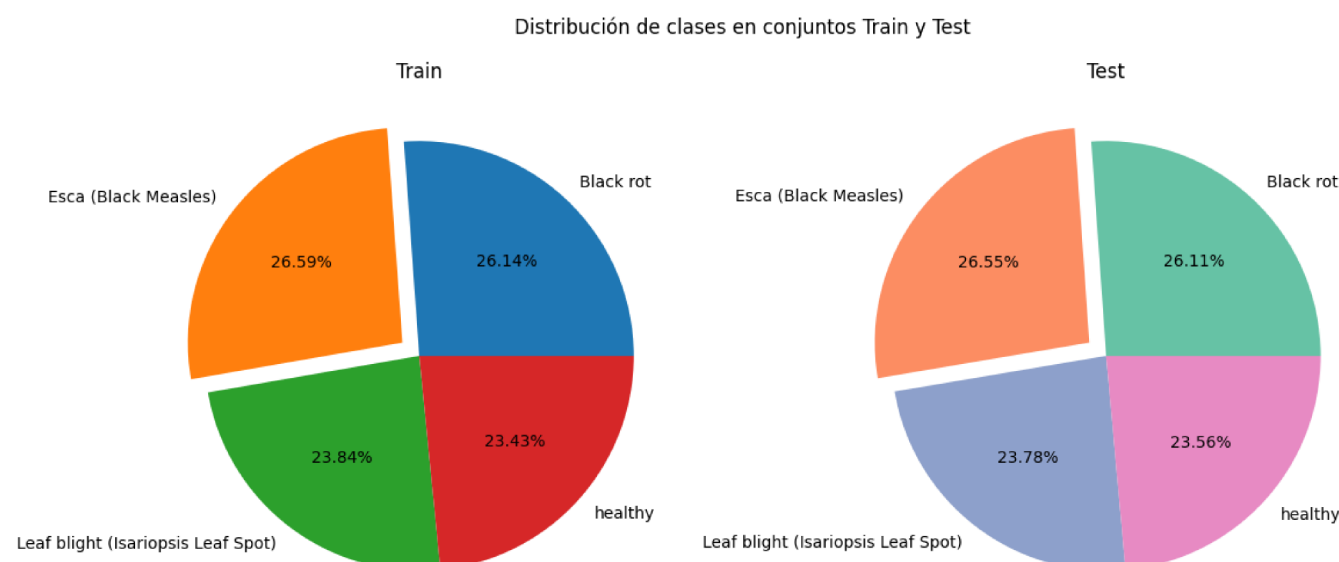


Figura 15: Porcentaje del total de muestras para cada clase en los conjuntos de entrenamiento y prueba.
Fuente: Elaboración propia.

En la figura 16 se ilustra una muestra aleatoria de 12 imágenes del conjunto de entrenamiento, en donde se muestran por lo menos dos imágenes por clase.

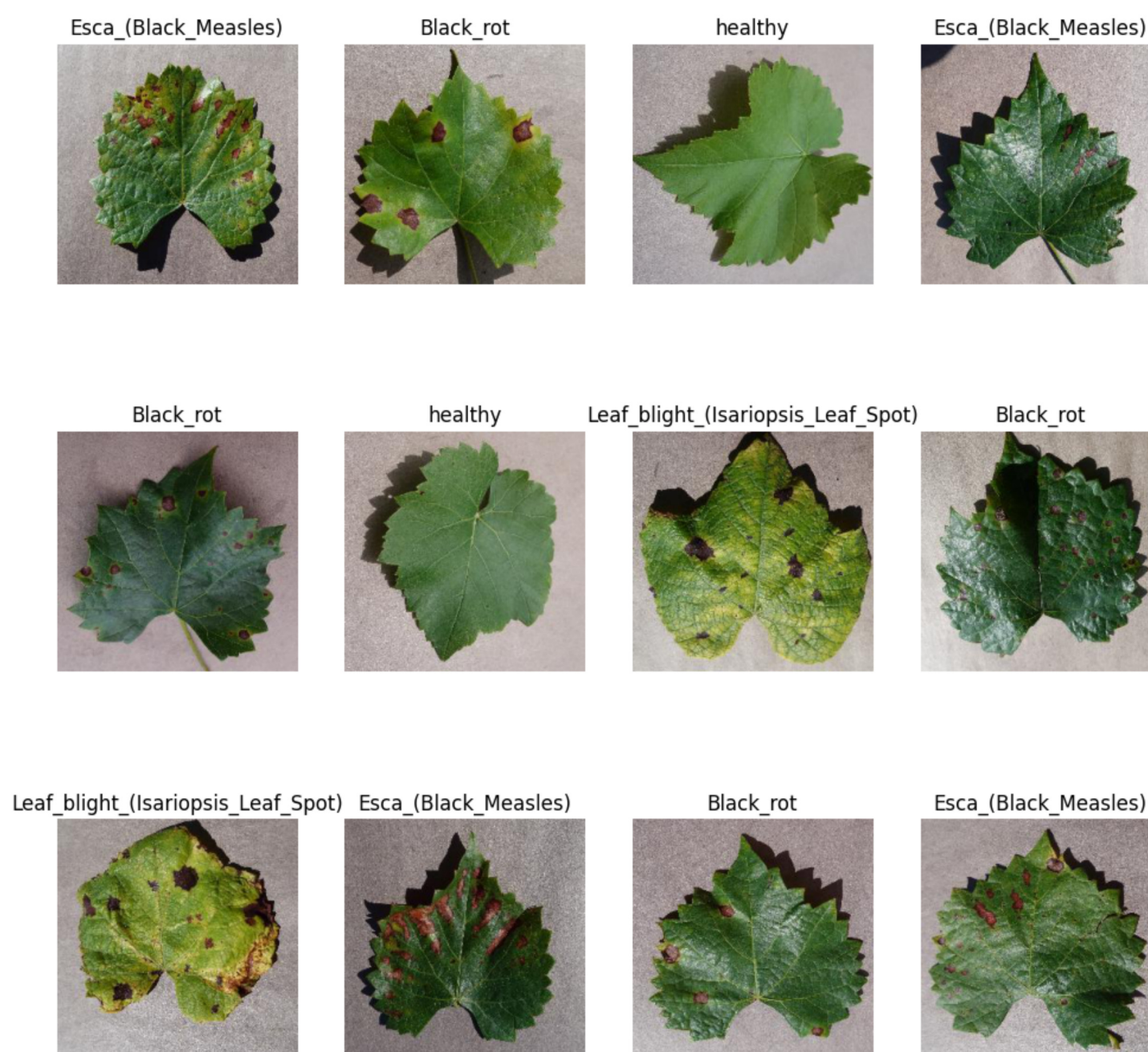


Figura 16: Muestra de 12 imágenes pertenecientes a las 4 clases presentes en el *dataset* de entrenamiento.

Fuente: Elaboración propia.

Para adaptar las imágenes de entrada a los modelos y aumentar la capacidad de generalización de estos se aplican las funciones de pre-procesamiento y aumentación de datos de Keras sobre las imágenes de entrada. Esto se implementa a través de capas de pre-procesamiento que se incorporan dentro del modelo, por lo que el proceso se realiza durante el entrenamiento. Inicialmente se redimensionan las imágenes de entrada a 224x224 (240x240 para EfficientNet-B1), y posteriormente se aplican operaciones para la aumentación de datos. Las operaciones que se realizan en esta etapa son volteo horizontal aleatorio, rotación aleatoria, *zoom* aleatorio y contraste aleatorio.

3.3. Redes neuronales convolucionales

Para solucionar el problema de la clasificación se decide emplear redes neuronales convolucionales pre-entrenadas, lo que se corresponde a **transferencia del aprendizaje** o *transfer learning*. La transferencia del aprendizaje tiene como objetivo mejorar el desempeño de modelos de aprendizaje transfiriendo los conocimientos adquiridos en un campo similar, pero diferente [Zhuang *et al.*, 2020].

Se propone poner en práctica la transferencia del aprendizaje escogiendo dos arquitecturas de CNNs pre-entrenadas en el *dataset ImageNet*, ambas con resultados notables en el área de clasificación de imágenes y alto desempeño en la detección de enfermedades en parras dentro de la literatura científica, además de servir como buenos modelos base en el desarrollo de nuevos modelos para resolver dicho problema; estas son **Resnet50** y **Resnet101** [Xie *et al.*, 2020], y **EfficientNet B0-B1** [Atila *et al.*, 2021].

Para los modelos Resnet50 y Resnet101 se implementaron las arquitecturas originales de cada red, se escogió *average pooling* y se utilizaron los pesos de *ImageNet* en la inicialización. Las entradas son redimensionadas a 224x224 píxeles para ambas redes. Se añaden cinco nuevas capas a los modelos iniciales: una capa densa de 128 unidades con activación ReLu, una capa *Dropout*¹⁹ con probabilidad de 0.4, otra capa densa de 256 unidades con activación ReLu, otra capa *Dropout* con probabilidad de 0.4 y finalmente una capa de clasificación densa para cuatro categorías que usa la función de activación softmax.

En el caso del modelo EfficientNet se implementaron las arquitecturas originales de las redes B0 y B1, utilizando los pesos de *ImageNet* en la inicialización y *max pooling*. Las entradas se redimensionaron a 224x224 y a 240x240 píxeles para EfficientNet-B0 y EfficientNet-B1 respectivamente; esto corresponde a las dimensiones de entrada utilizadas en el *paper* donde fueron propuestos los modelos originalmente [Tan y Le, 2019]. Se añaden cinco nuevas capas a cada uno de los modelos iniciales: una capa densa de 128 unidades con activación ReLu, una capa *Dropout* con probabilidad de 0.4, otra capa densa de 256 unidades con activación ReLu, otra capa *Dropout* con probabilidad de 0.4 y finalmente una capa de clasificación densa para cuatro categorías que usa la función de activación softmax.

Para todos los modelos entrenados se utiliza el *learning rate* por defecto (0.001), el optimizador Adam, función de pérdida del tipo *categorical crossentropy* y la exactitud como métrica en el entrenamiento durante la compilación de estos. El entrenamiento se extiende por 40 épocas, pero se establece *early stopping* para detener el entrenamiento y restaurar los mejores pesos aprendidos si es que la pérdida de validación no mejora después de 5 épocas.

Todos los modelos fueron entrenados en el *dataset Grape Disease*, el cual se encuentra ya dividido en conjuntos de entrenamiento y prueba que contienen un 80 % y 20 % del total de muestras respectivamente. El conjunto de entrenamiento fue separado en un 80 % pa-

¹⁹Las capas *Dropout* son capas que "desechan" parte de la entrada aleatoriamente durante el entrenamiento, así ayudando a prevenir *overfitting*.

ra el entrenamiento y un 20 % para la validación. Como resultado, hay 5779 imágenes de entrenamiento, 1443 de validación y 1808 en el conjunto de prueba.

3.4. Explicación visual de las predicciones

Después de que los modelos anteriormente propuestos finalicen el proceso de entrenamiento, se desea llegar a una manera de representar visualmente la manera en que estos toman decisiones durante la etapa de clasificación. Para solucionar este problema se propone el uso de *Gradient-weighted Class Activation Mapping*.

Gradient-weighted Class Activation Mapping (Grad-CAM) es una técnica de visualización propuesta por [Selvaraju *et al.*, 2017] que tiene el objetivo de producir “explicaciones visuales” de las decisiones que toma cualquier modelo basado en CNNs, con tal de volverlo más transparente e interpretable para un usuario humano. Grad-CAM usa las gradientes de cualquier concepto que se producen al pasar el *input* por la última capa convolucional del modelo, para así elaborar un mapa de localizaciones que destaca las regiones de la imagen que más importancia tienen para predecir dicho concepto.

El objetivo de implementar Grad-CAM es visualizar qué secciones de las imágenes de prueba tienen mayor influencia en el proceso de clasificación. Esto permitirá también reconocer visualmente si es que algún modelo está erróneamente utilizando regiones de las imágenes que un observador humano consideraría irrelevantes en las predicciones. Para elaborar los Grad-CAMs de cada modelo se utilizó una muestra aleatoria de 12 imágenes del conjunto de prueba de *Grape Disease*.

CAPÍTULO 4

VALIDACIÓN DE LA SOLUCIÓN

En este capítulo se describen y analizan los resultados obtenidos al entrenar los modelos de clasificación propuestos para solucionar el problema de la detección de enfermedades en plantas.

4.1. Resultados Resnet50

Bajo la configuración aplicada, el entrenamiento del modelo Resnet50 usando el conjunto de prueba *Grape Disease* finaliza al alcanzar 10 épocas, deteniéndose tempranamente debido al *early stopping*. La mejor exactitud de entrenamiento se obtiene en la época 8, con un valor de 99,78 %. La mejor exactitud de validación es 99,45 %, valor que se alcanza en las épocas 4 y 9. La evolución de las exactitudes durante el proceso de entrenamiento se puede visualizar en la Figura 17.

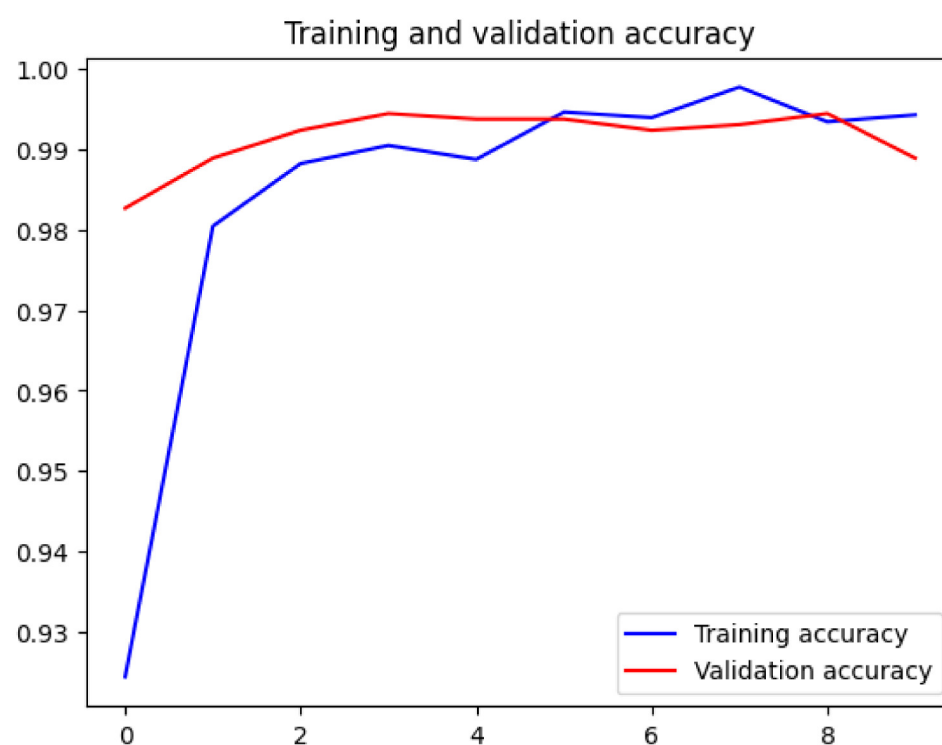


Figura 17: Gráfico de exactitud de entrenamiento y validación según época del modelo Resnet50. Notar que 0 en el eje x corresponde a la época 1.

Fuente: Elaboración propia.

Al evaluar el modelo resultante en el conjunto de prueba se obtiene una exactitud total del **99,61 %** y *loss* de 0,01333, con precisiones del 99,36 %, 99,17 %, 100 % y 100 % para las clases *black rot*, *esca*, *leaf blight* y *healthy* respectivamente. El *recall* obtenido para cada clase es 99,15 %, 99,37 %, 100 % y 100 % para *black rot*, *esca*, *leaf blight* y *healthy* respectivamente. En la Figura 18 se ilustra una matriz de confusión creada a partir de los resultados de la evaluación.

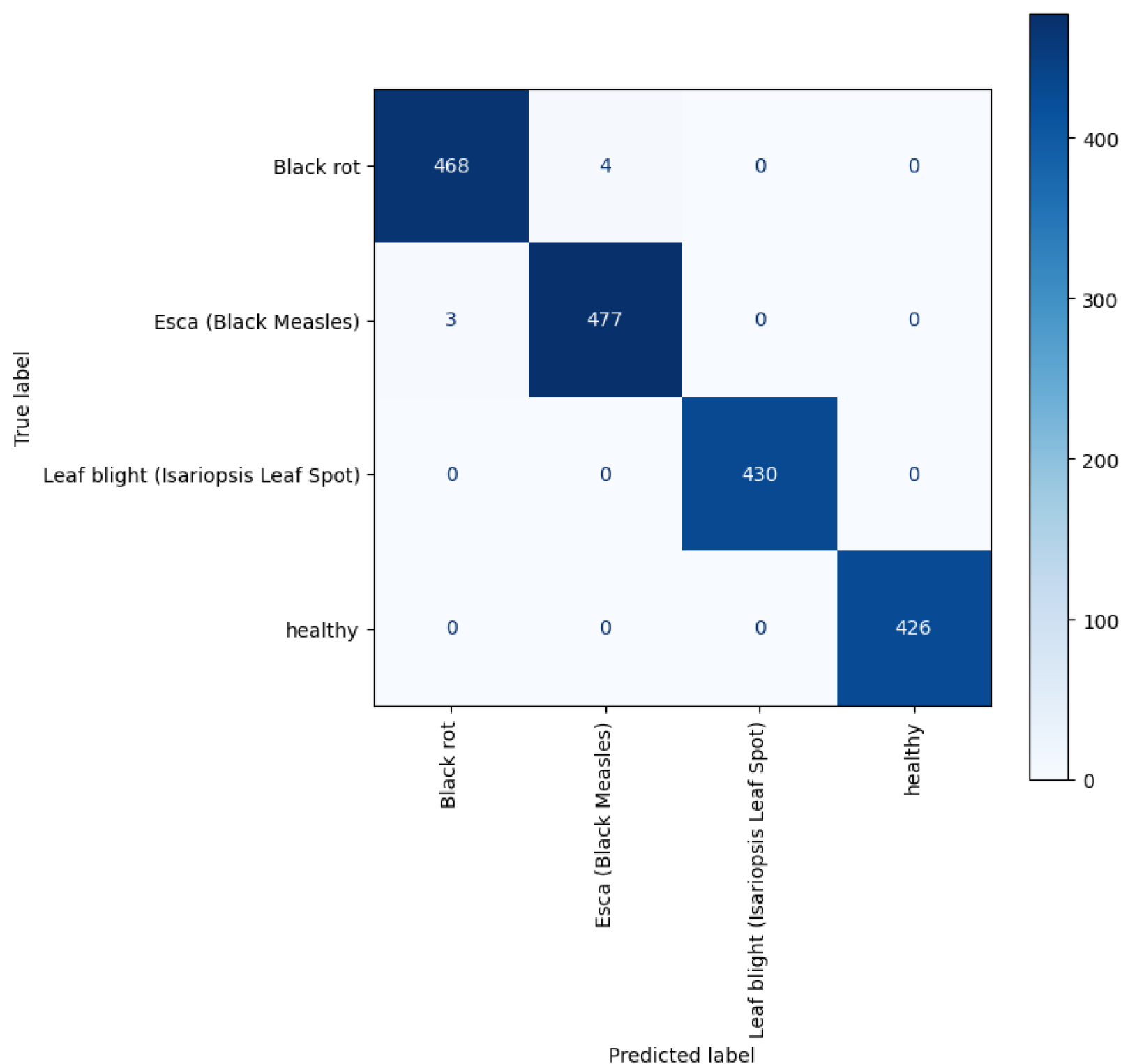


Figura 18: Matriz de confusión para modelo Resnet50 al evaluarlo en el conjunto de prueba de Grape Disease.

Fuente: Elaboración propia.

Los resultados del modelo Resnet50 utilizando las métricas de exactitud, precisión, *recall*, F1-score en sus variantes micro (expresados como *accuracy*), macro y con pesos según la clase (*weighted*) se visualizan en la tabla 1. Notar que *support* se refiere al número de muestras.

Tabla 1: Resultados del modelo Resnet50.

Fuente: Elaboración Propia.

	Precision	Recall	F1-Score	Support
Black rot	0.993631	0.991525	0.992577	472
Esca (Black Measles)	0.991684	0.993750	0.992716	480
Leaf blight (Isariopsis Leaf Spot)	1.0	1.0	1.0	430
Healthy	1.0	1.0	1.0	426
Accuracy	0.996128	0.996128	0.996128	1808
Macro average	0.996329	0.996319	0.996323	1808
Weighted average	0.996129	0.996128	0.996128	1808

4.2. Resultados Resnet101

El objetivo de entrenar un modelo Resnet101 en adición al modelo Resnet50 anteriormente descrito es determinar si entrenar un modelo más grande producirá mejores resultados en la clasificación.

El entrenamiento del modelo Resnet101 es detenido tempranamente por el *early stopping* al alcanzar 17 épocas, 7 épocas más que el modelo Resnet50. La mejor exactitud de entrenamiento se obtiene en la época 14, con un valor de 99,88 %. La mejor exactitud de validación obtenida es 99,65 %, valor que se alcanza tanto en la época 5 como en la 12. La evolución de la exactitud durante el proceso de entrenamiento puede verse en la Figura 19.

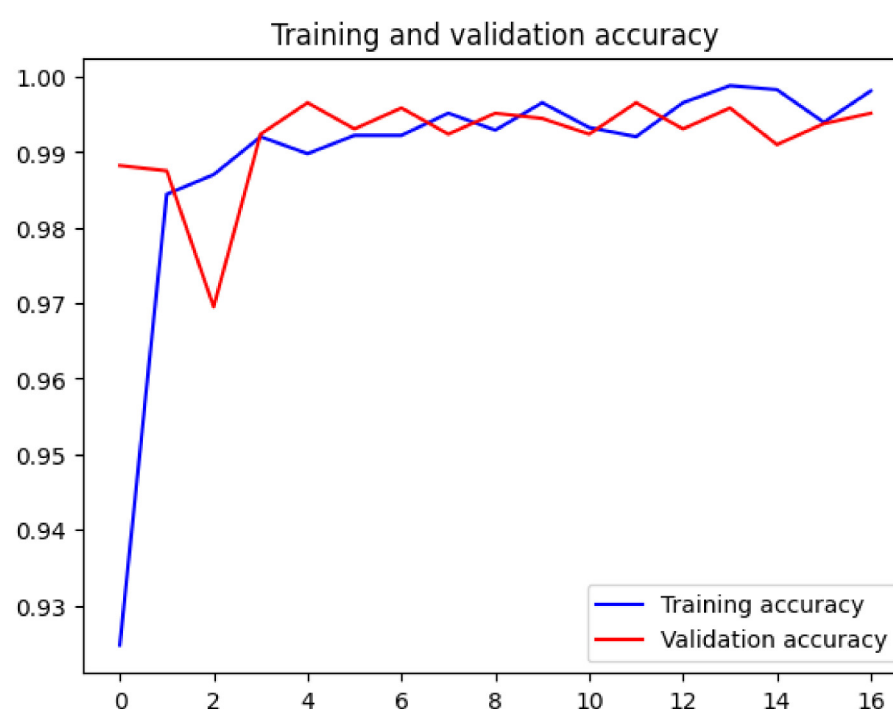


Figura 19: Gráfico de exactitud de entrenamiento y validación según época del modelo Resnet101. 0 en el eje x corresponde a la época 1.

Fuente: Elaboración propia.

Al momento de evaluar el modelo Resnet101 en el conjunto de prueba se obtiene una exactitud total del **99,56 %** y *loss* de 0,0147, con precisiones del 98,54 %, 99,79 %, 100 % y 100 % para las clases *black rot*, *esca*, *leaf blight* y *healthy* respectivamente. El *recall* obtenido para cada clase es 99,79 %, 98,54 %, 100 % y 100 % para *black rot*, *esca*, *leaf blight* y *healthy* respectivamente. En la Figura 20 se ilustra una matriz de confusión creada a partir de los resultados de la evaluación.

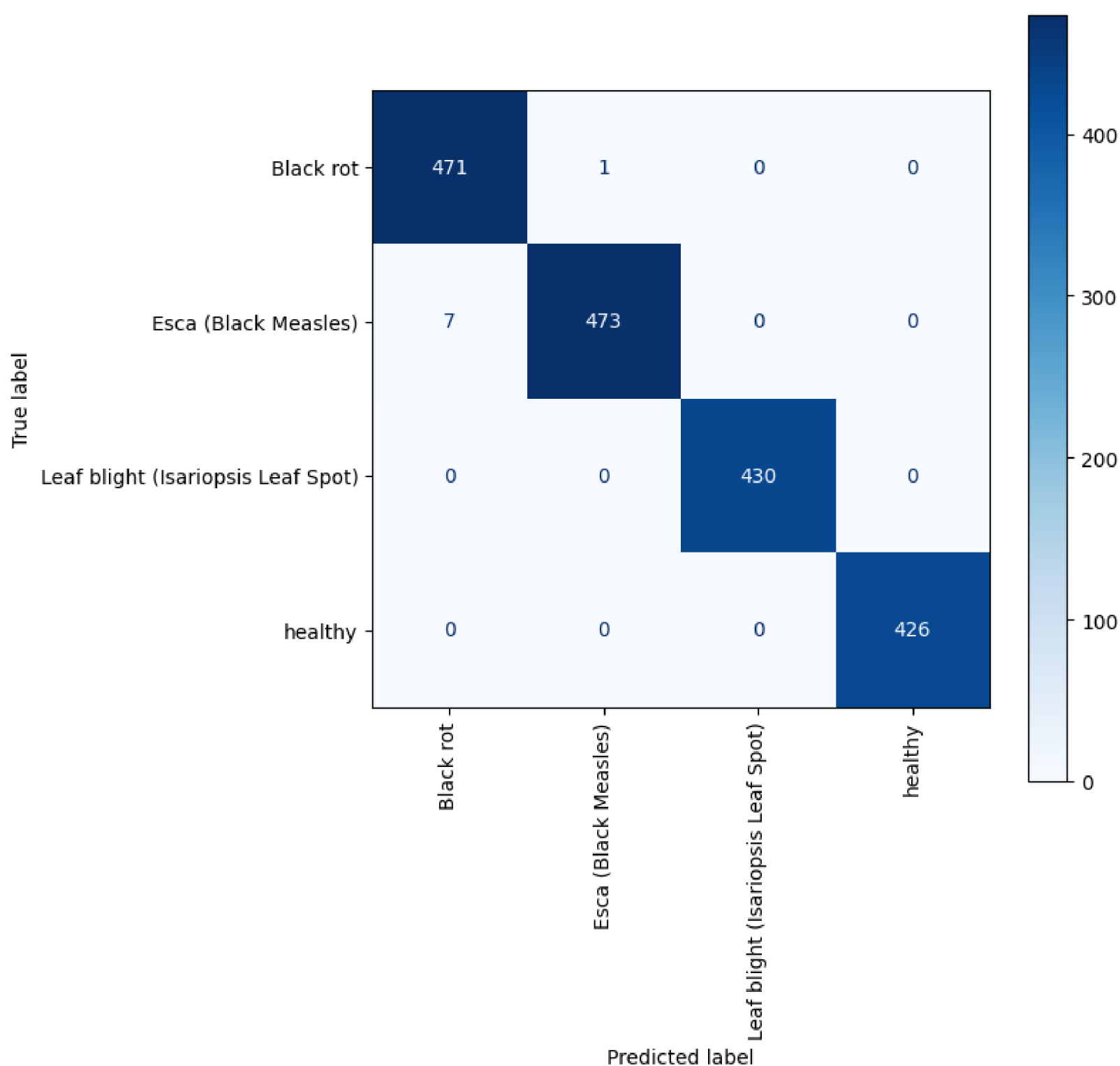


Figura 20: Matriz de confusión para modelo Resnet101 al evaluarlo en el conjunto de prueba de Grape Disease.

Fuente: Elaboración propia.

Los resultados del modelo Resnet101 utilizando las métricas de exactitud, precisión, *recall*, F1-score en sus variantes micro (expresados como *accuracy*), macro y con pesos según clase (*weighted*) se pueden ver en la tabla 2.

Tabla 2: Resultados del modelo Resnet101.

Fuente: Elaboración Propia.

	Precision	Recall	F1-Score	Support
Black rot	0.985356	0.997881	0.991579	472
Esca (Black Measles)	0.997890	0.985417	0.991614	480
Leaf blight (Isariopsis Leaf Spot)	1.0	1.0	1.0	430
Healthy	1.0	1.0	1.0	426
Accuracy	0.995575	0.995575	0.995575	1808
Macro average	0.995811	0.995825	0.995798	1808
Weighted average	0.995617	0.995575	0.995575	1808

4.3. Resultados EfficientNet-B0

El entrenamiento del modelo EfficientNet-B0 finaliza al alcanzar 23 épocas del total de 40, deteniéndose tempranamente debido al *early stopping*. La mejor exactitud de entrenamiento es 98,48 %, valor obtenido en la época 17. La exactitud de validación máxima es 99,31 %, la cual se alcanza en las épocas 18, 20 y 23. La evolución de las exactitudes durante el proceso de entrenamiento se puede visualizar en la Figura 21.

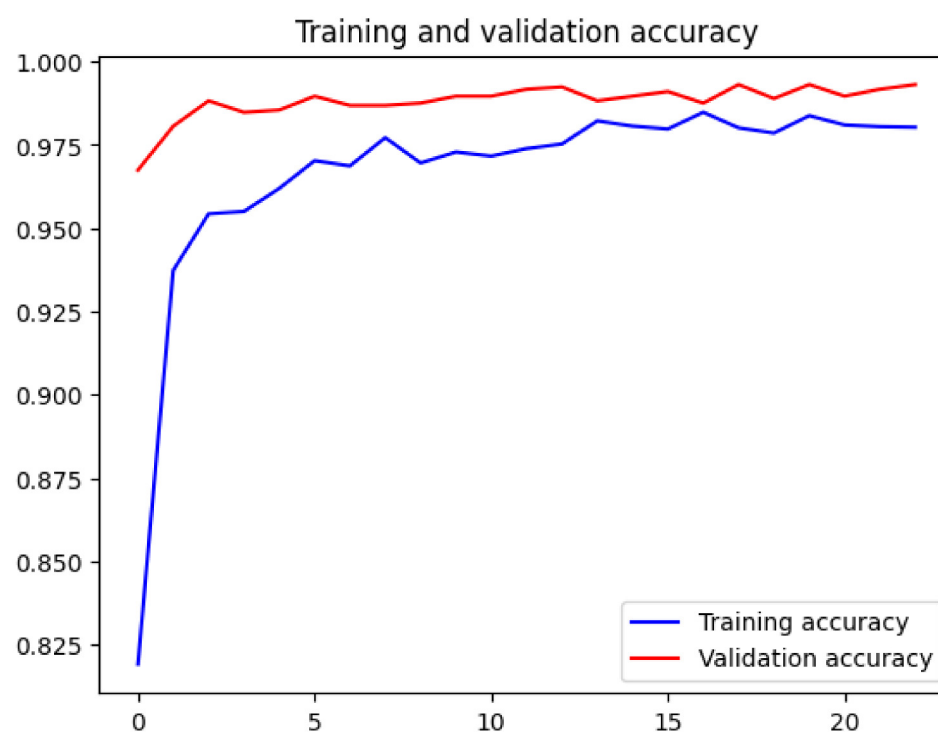


Figura 21: Gráfico de exactitud de validación y entrenamiento según época de modelo EfficientNet-B0.

Fuente: Elaboración propia.

EfficientNet-B0 evaluado en el conjunto de prueba obtiene una exactitud del **99,23 %** y *loss* de 0.0223, con precisiones del 97,71 %, 99,36 %, 100 % y 100 % para las clases *black rot*, *esca*, *leaf blight* y *healthy* respectivamente. El *recall* obtenido para cada clase es 99,36 %, 97,71 %, 100 % y 100 % para *black rot*, *esca*, *leaf blight* y *healthy* respectivamente. En la Figura 22 se ilustra una matriz de confusión creada a partir de los resultados de la evaluación.

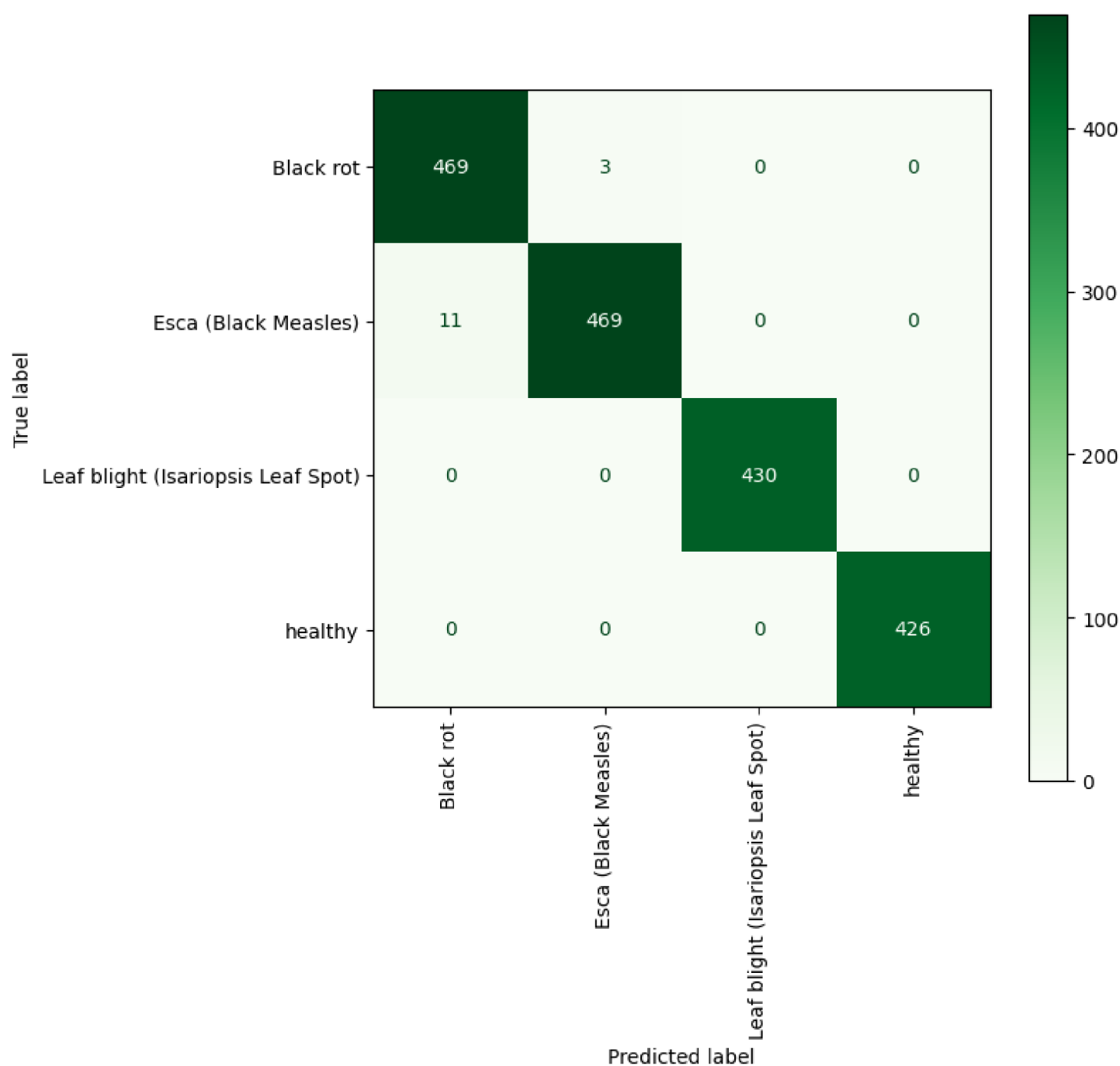


Figura 22: Matriz de confusión para el modelo EfficientNet-B0 al evaluarlo en el conjunto de prueba de Grape Disease.

Fuente: Elaboración propia.

Los resultados del modelo EfficientNet-B0 utilizando las métricas de exactitud, precisión, *recall*, F1-score con los enfoques micro (expresado como *accuracy*), macro y con pesos de acuerdo a la clase (*weighted*) se visualizan en la tabla 3.

Tabla 3: Resultados de la evaluación del modelo EfficientNet-B0.

Fuente: Elaboración Propia.

	Precision	Recall	F1-Score	Support
Black rot	0.977083	0.993644	0.985294	472
Esca (Black Measles)	0.993644	0.977083	0.985294	480
Leaf blight (Isariopsis Leaf Spot)	1.0	1.0	1.0	430
Healthy	1.0	1.0	1.0	426
Accuracy	0.992257	0.992257	0.992257	1808
Macro average	0.992682	0.992682	0.992647	1808
Weighted average	0.992330	0.992257	0.992257	1808

4.4. Resultados EfficientNet-B1

De manera similar a los modelos Resnet50 y Resnet101 descritos anteriormente, se entrenó un modelo EfficientNet-B1 en adición al modelo EfficientNet-B0 para comprobar si un modelo de mayor complejidad entregará mejores resultados en la clasificación.

El *early stopping* utilizado en los otros modelos causó que EfficientNet-B1 entrenara por pocas épocas y que obtuviera una curva de exactitud que evidenciaba que aún se podían obtener beneficios al entrenar al modelo por más tiempo. Esta curva se puede ver en la Figura 23. Se cambió la configuración de *early stopping* para que el entrenamiento termine prematuramente si transcurren 8 épocas sin que mejore el *loss* de validación en lugar de 5. Este cambio conlleva a una ligera mejora en los resultados. La exactitud para la configuración original fue de 98,06 %.

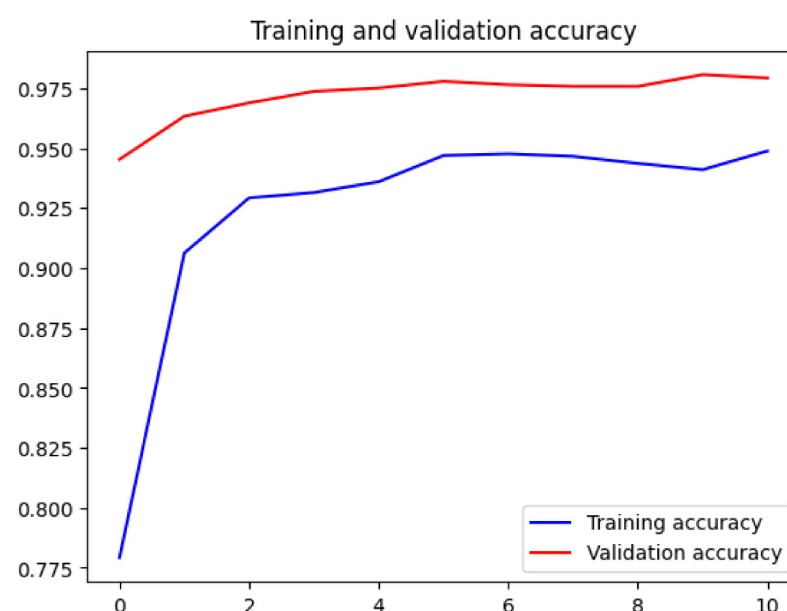


Figura 23: Gráfico de exactitud de validación y entrenamiento según época de modelo EfficientNet-B1 previo al cambio de configuración.

Fuente: Elaboración propia.

EfficientNet-B1 finaliza su proceso de entrenamiento en la época 40 y *early stopping* no se activa. La exactitud de entrenamiento más alta es 97,75 % en la época 38. La mayor exactitud de validación es 98,82 %, valor que se alcanza en la época 40. La evolución de las exactitudes durante el proceso de entrenamiento se puede visualizar en la Figura 24.

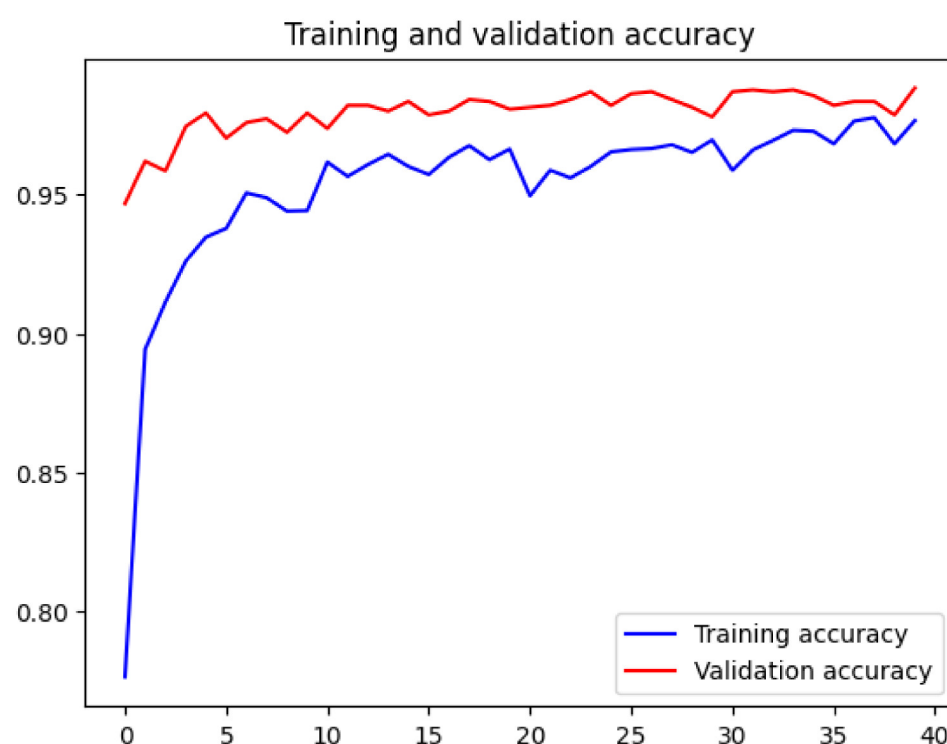


Figura 24: Gráfico de exactitud de validación y entrenamiento según época de modelo EfficientNet-B1.

Fuente: Elaboración propia.

EfficientNet-B1 evaluado en el conjunto de prueba obtiene una exactitud del **99,12 %** y 0,02695 de *loss*, con precisiones del 98,51 %, 98,13 %, 100 % y 100 % para las clases *black rot*, *esca*, *leaf blight* y *healthy* respectivamente. El *recall* obtenido para cada clase es 98,09 %, 98,54 %, 100 % y 100 % para *black rot*, *esca*, *leaf blight* y *healthy* respectivamente. Una matriz de confusión elaborada con estos resultados puede verse en la Figura 25.

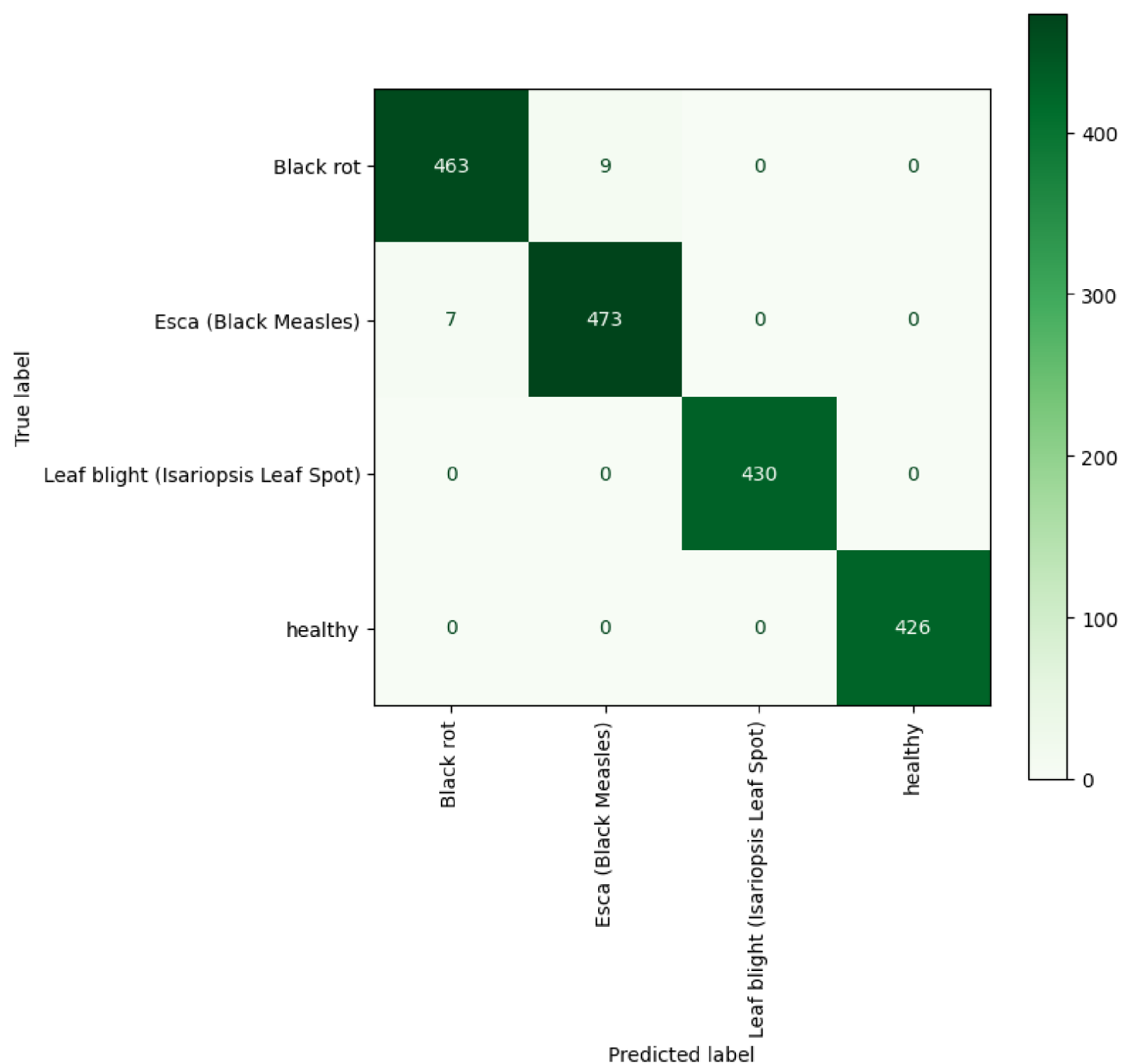


Figura 25: Matriz de confusión para el modelo EfficientNet-B1 al evaluarlo en el conjunto de prueba de Grape Disease.

Fuente: Elaboración propia.

Los resultados del modelo EfficientNet-B1 en las métricas de exactitud, precisión, *recall*, F1-score con los enfoques micro (expresado como *accuracy*), macro y con pesos de acuerdo a la clase (*weighted*) se visualizan en la tabla 4.

Tabla 4: Resultados de la evaluación del modelo EfficientNet-B1.

Fuente: Elaboración Propia.

	Precision	Recall	F1-Score	Support
Black rot	0.985106	0.980932	0.983015	472
Esca (Black Measles)	0.981328	0.985417	0.983368	480
Leaf blight (Isariopsis Leaf Spot)	1.0	1.0	1.0	430
Healthy	1.0	1.0	1.0	426
Accuracy	0.991150	0.991150	0.991150	1808
Macro average	0.991609	0.991587	0.991596	1808
Weighted average	0.991155	0.991150	0.991150	1808

4.5. Representaciones visuales

A pesar de que modelos anteriormente descritos producen buenos resultados en términos de métricas de clasificación, es difícil interpretar y entender el proceso que cada modelo ha realizado durante la clasificación para llegar a dichos resultados, pues estos funcionan prácticamente como cajas negras. Para llegar a una explicación visual fácil de entender para un observador humano se hizo uso de Grad-CAMs.

En las Figuras 26, 27, 28 y 29 se visualizan los Grad-CAM para los modelos Resnet50, Resnet101, EfficientNet-B0 y EfficientNet-B1 respectivamente. Cada visualización utiliza las mismas imágenes y muestra tanto la clase real como la clase predicha por el modelo. Las zonas de mayor importancia en la clasificación se destacan con color rojo, mientras que las áreas menos influyentes aparecen de color azul.

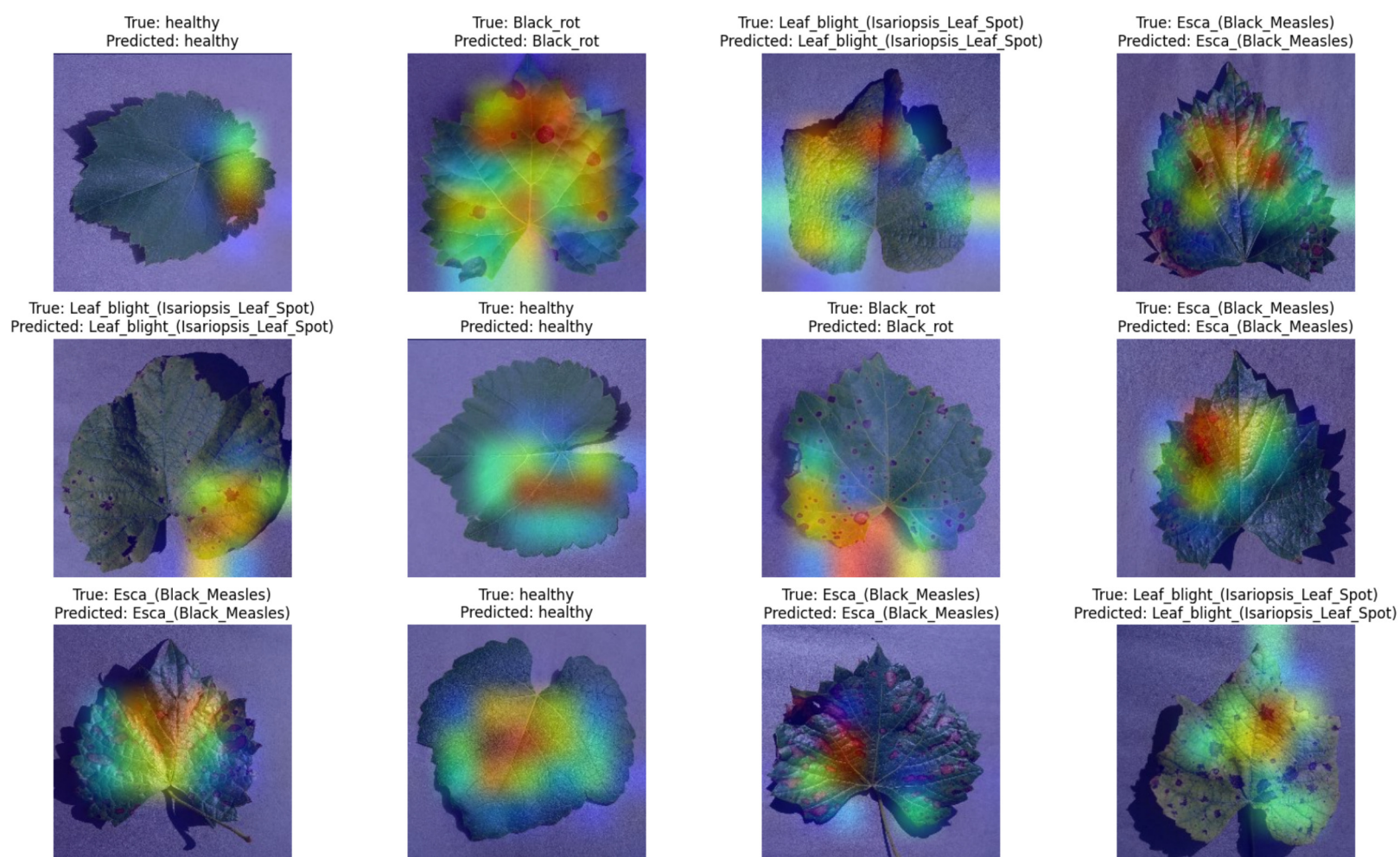


Figura 26: Grad-CAMs del modelo Resnet50.

Fuente: Elaboración propia.

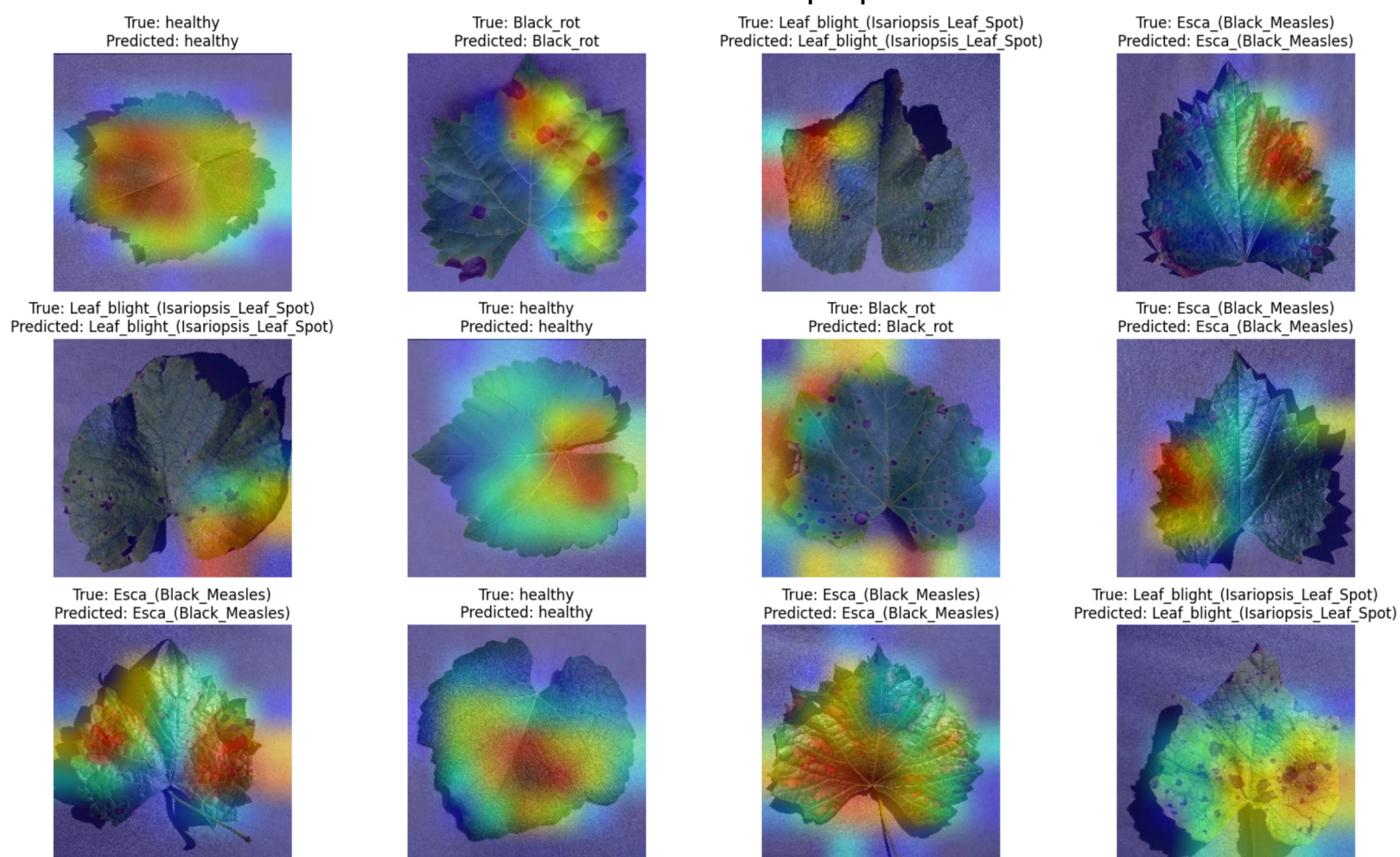


Figura 27: Grad-CAMs del modelo Resnet101.

Fuente: Elaboración propia.

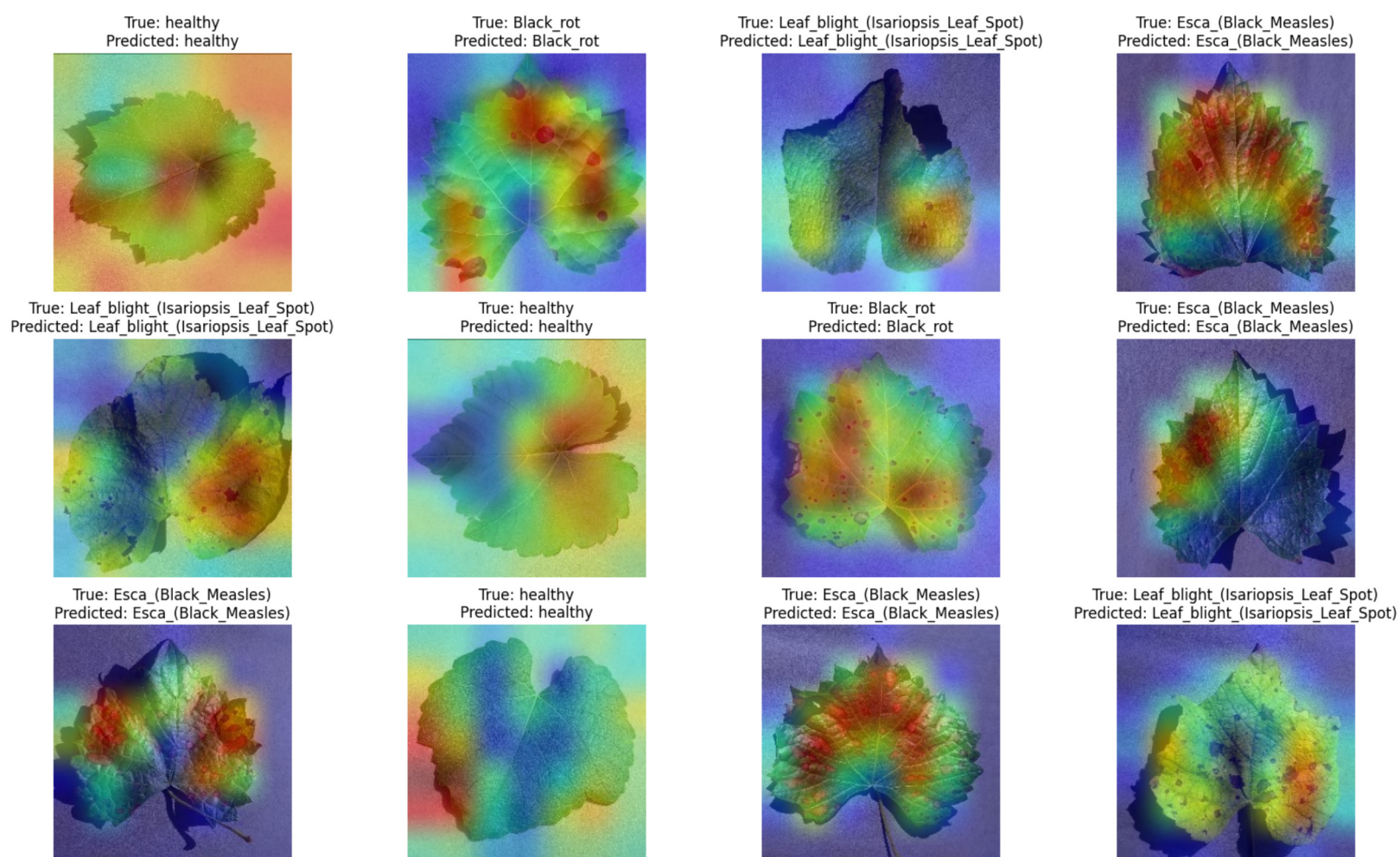


Figura 28: Grad-CAMs del modelo EfficientNet-B0.

Fuente: Elaboración propia.

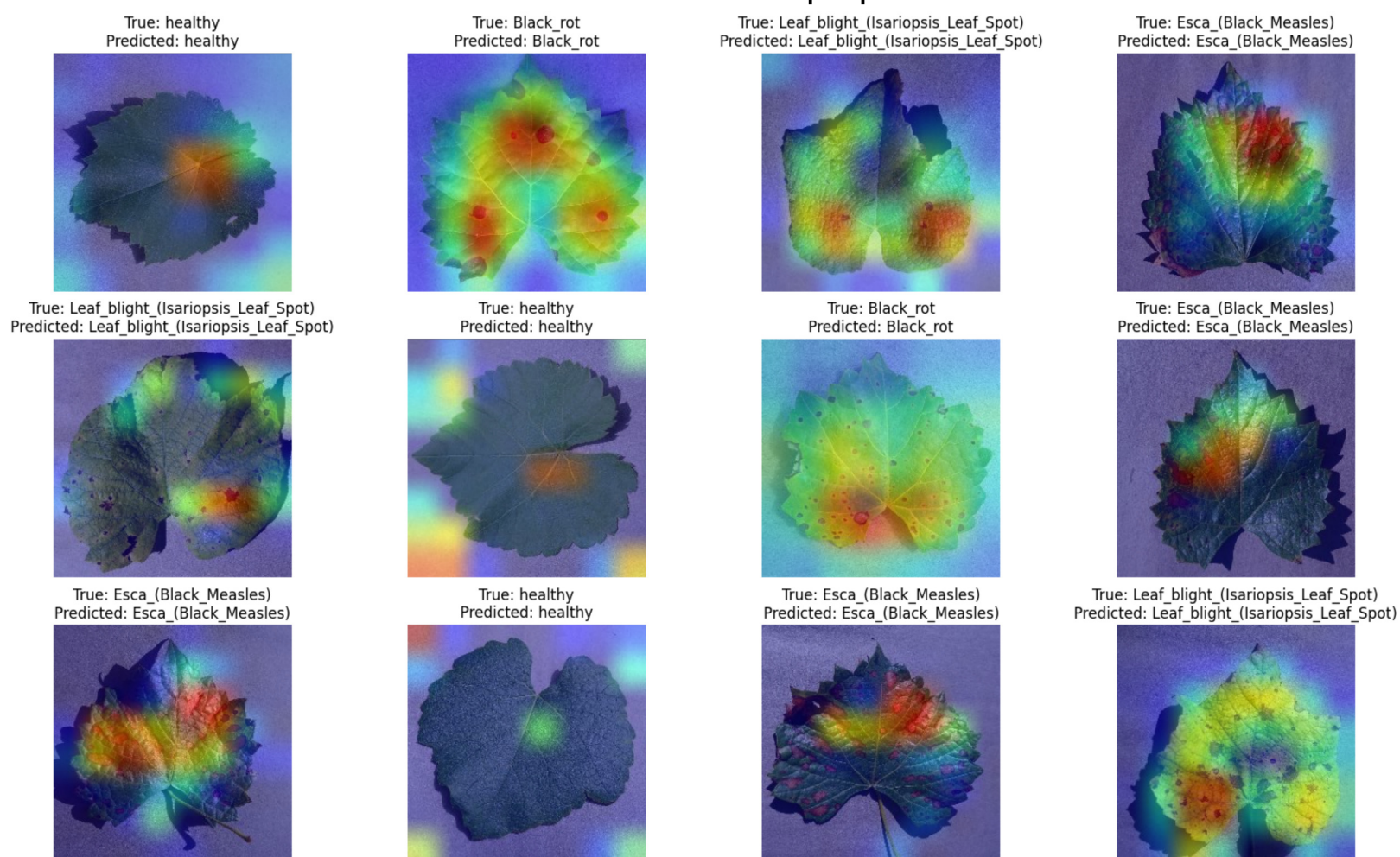


Figura 29: Grad-CAMs del modelo EfficientNet-B1

Fuente: Elaboración propia.

Puede observarse en las Grad-CAMs para la clase *Black Rot* que los modelos entrenados consideran las manchas circulares de las hojas como la región más importante, aunque Resnet101 consideró más importantes los bordes dañados de la hoja en una de las muestras.

En el caso de la clase *Esca (Black Measles)*, las regiones de las hojas con manchas alargadas son consideradas las más importantes en la clasificación por los cuatro modelos entrenados. Las zonas sin manchas o irregularidades suelen ser ignoradas.

Para la clase *Leaf Blight*, se consideraron más importantes las áreas con manchas de forma irregular, aunque los cuatro modelos también tomaron en cuenta la mayoría de la superficie de la hoja en varias de las muestras, posiblemente porque el color de las regiones enfermas es más claro.

La clase *Healthy* presenta los Grad-CAMs más inconsistentes. Resnet101 y EfficientNet-B0 dan importancia a la mayoría de la superficie total de la hoja y en ocasiones parte de los bordes de estas, mientras de Resnet50 y EfficientNet-B1 se concentran en regiones pequeñas o medianas cercanas al centro de la hoja, aunque en algunas muestras EfficientNet-B1 asigna también importancia a zonas fuera de la hoja. El comportamiento inconsistente podría deberse a que las hojas sanas no tienen muchas características destacables a diferencia de las hojas enfermas en el *dataset*, las cuales poseen manchas de distintas formas y tamaños.

4.6. Análisis del desempeño

Al evaluarlos con el conjunto de prueba, las exactitudes del modelo Resnet50 y Resnet101 son 99,61 % y 99,56 % respectivamente. Esto evidencia que para los modelos Resnet propuestos, incrementar el tamaño de la arquitectura no mejora ni afecta significativamente el desempeño, lo que evidencia que la complejidad del conjunto de datos no amerita entrenar un modelo más grande. Resnet101 tuvo un desempeño ligeramente mejor en la clasificación de imágenes de la clase *Black rot* que el modelo Resnet50, aunque su rendimiento clasificando *Esca* disminuye en comparación. Sin embargo, las diferencias son tan pequeñas que el desempeño de ambos modelos es prácticamente equivalente. Ya que Resnet101 es un modelo más complejo y por tanto demora más entrenarlo, Resnet50 es una mejor opción para el problema.

En el caso de los modelos EfficientNet, las exactitudes obtenidas en la evaluación son de 99,23 % para EfficientNet-B0 y 99,12 % para EfficientNet-B1. Antes del cambio de configuración, EfficientNet-B1 tenía un 98,06 % de exactitud. Si no se incorpora la modificación de *early stopping*, incrementar el tamaño del modelo EfficientNet reduce la exactitud, por lo que la complejidad extra de B1 tiene efectos negativos en el rendimiento. Tras el cambio de configuración EfficientNet-B0 y B1 obtienen resultados similares, pero B1 requiere casi el doble de épocas de entrenamiento, por lo que EfficientNet-B0 es más apropiado para el problema.

La tabla 5 contiene el desempeño de cada modelo de acuerdo con la exactitud, número de épocas requeridas en el entrenamiento y precisión para cada una de las cuatro clases del *dataset*.

Tabla 5: Resultados de la evaluación de todos los modelos. Los mejores valores se destacan con negrita si es que no existen empates. Exactitud se abrevia como Acc. y precisión como Pre.

Fuente: Elaboración Propia.

	Acc.	Épocas	Pre. B. Rot	Pre. Esca	Pre. L. Blight	Pre. Healthy
Resnet50	99,61 %	10	99,36 %	99,17 %	100 %	100 %
Resnet101	99,56 %	17	98,54 %	99,79 %	100 %	100 %
EfficientNet-B0	99,23 %	23	97,71 %	99,36 %	100 %	100 %
EfficientNet-B1	99,12 %	40	98,51 %	98,13 %	100 %	100 %

Entre todos los modelos, el mejor desempeño general lo obtiene **Resnet50**. Sin embargo, todos los modelos propuestos se han adaptado correctamente a las condiciones del problema y han entregado buenos resultados, produciendo exactitudes de más del 99 % y precisiones mayores al 98 %. Todos los modelos lograron clasificar correctamente todas las muestras de clase *Leaf Blight* y *Healthy*, lo que podría explicarse por ser las clases más visualmente únicas dentro del conjunto de datos utilizado.

CAPÍTULO 5

CONCLUSIONES

En la agricultura, la detección temprana de enfermedades y plagas en los cultivos es esencial para mantener un buen rendimiento económico, pero el proceso de diagnóstico es complejo, lento y costoso, principalmente debido a la necesidad de contratar especialistas para el proceso. Este requerimiento causa dependencia en el juicio experto, lo que puede introducir el riesgo de diagnósticos errados dada la gran cantidad de patógenos existentes, pues muchos de estos causan síntomas similares en las plantas.

En este trabajo se abordó la problemática de la detección de enfermedades en plantas utilizando procesamiento de imágenes, proponiendo un enfoque basado en *Deep Learning*. Se construyeron cuatro modelos de redes neuronales convolucionales utilizando transferencia del aprendizaje para la clasificación de tres enfermedades presentes en hojas de vides; estos son Resnet50, Resnet101, EfficientNet-B0 y EfficientNet-B1. El propósito de incluir distintos tamaños de las arquitecturas es determinar si la complejidad de problema amerita emplear modelos más grandes para la clasificación. Una vez finalizado el entrenamiento, se hace uso de Grad-CAMs para generar una explicación visual de la manera en que cada modelo realiza sus predicciones.

Los modelos propuestos poseen un alto grado de adaptación al problema, lo que se demuestra en los resultados; tras ser entrenados sobre el *dataset* Plant Disease para la identificación de las enfermedades podredumbre negra, la esca de la vid y la *Isariopsis clavispora*, los modelos Resnet50, Resnet101, EfficientNet-B0 y EfficientNet-B1 obtienen un 99,61 %, 99,56 %, 99,23 % y 99,12 % de exactitud en la evaluación respectivamente. El modelo con mejor desempeño general fue Resnet50, siendo el modelo con mayor exactitud y el que requirió menos épocas de entrenamiento, pero todos los modelos propuestos demostraron un alto grado de adaptación al problema definido. Para tanto Resnet como EfficientNet, la arquitectura más sencilla entregó resultados ligeramente mejores, lo que evidencia que no es necesario plantear modelos más complejos para la resolución del problema. Sin embargo, estos resultados podrían cambiar si es que se desea ampliar el alcance de la problemática incorporando nuevas clases de enfermedades.

En conclusión, los modelos basados en CNNs que fueron implementados en este trabajo obtienen resultados notables, alcanzando exactitudes de más del 99 % en la clasificación de enfermedades presentes en vides. Este nivel de desempeño es altamente competitivo con los resultados observados en la literatura científica para la detección de enfermedades en distintos tipos de cultivos utilizando sistemas similares. Los Grad-CAMs creados para cada modelo evidencian que cada uno de estos ha aprendido correctamente las características más importantes de la manifestación de cada una de las enfermedades, destacando principalmente las zonas de las hojas en donde existen daños causados por la patología.

El buen desempeño en la clasificación que presentan los modelos propuestos evidencia que

enfoques basados en *Deep Learning* son una alternativa viable y con alto potencial para la detección de enfermedades en plantas con un elevado nivel de precisión. La agricultura es un campo en donde todavía existe un alto grado de dependencia en el monitoreo manual y juicio experto para el diagnóstico de enfermedades y plagas en los cultivos, por lo que el desarrollo de nuevas tecnologías de detección utilizando inteligencia artificial y *Deep Learning* podría contribuir significativamente a simplificar y agilizar el proceso de diagnóstico, reduciendo no sólo los costos de contratar especialistas, sino también minimizando potenciales daños a los cultivos y las consecuentes pérdidas económicas causadas por la detección tardía.

5.1. Trabajo futuro

Una posible mejora a la capacidad de generalización de los modelos propuestos sería entrenar a estos utilizando conjuntos de datos que contengan las muestras en un mayor número de entornos y condiciones, como por ejemplo sin extraer las hojas de la planta, con iluminación limitada o en superficies de distintos colores, pues el *dataset* utilizado solo contiene fotos de hojas aisladas con un fondo gris de color sólido. Esto permitiría que el modelo mantenga un alto grado de efectividad incluso si se cambian las condiciones con las que se tomaron las fotos de se quieren analizar, lo que sería altamente relevante en caso de que se desee utilizar los modelos en sistemas desplegados en terreno. Otra alternativa es establecer un algoritmo para aislar las hojas de su entorno removiendo el fondo, así permitiendo al modelo concentrarse sólo en la hoja misma durante el proceso de clasificación.

Para incrementar los posibles casos de uso de los modelos propuestos, estos podrían entrenarse incorporando datos de más enfermedades y/o plagas presentes en las vides como el oídio y el mildiu, así logrando detectar un mayor número de anomalías presentes en estas plantas, pues el conjunto de datos utilizado tiene solo tres. Otra mejora interesante sería incorporar a los datos de entrenamiento no solo muestras de hojas, sino también imágenes de los frutos y troncos, pues ambos pueden evidenciar síntomas causados por patologías. Esto podría contribuir a otorgar más flexibilidad a un modelo entrenado al ser este capaz de reconocer más tipos de muestras.

Para este trabajo los modelos propuestos fueron entrenados para reconocer enfermedades en hojas de vides, pero para un trabajo futuro sería beneficioso incorporar otros tipos de cultivos para así aumentar significativamente los casos de uso posibles para un sistema que incorpore dichos modelos. Una opción posible sería utilizar el conjunto de datos Plant Village [Hughes *et al.*, 2015] en su completitud, así incorporando a los modelos clasificadores la habilidad de detectar patologías en manzanos, plantas de tomates y papas, entre otros cultivos. Si se dispone de acceso a los cultivos sanos y enfermos mismos, la asistencia de especialistas en el diagnóstico de plagas y enfermedades y equipamiento para la toma de muestras, podrían generarse nuevos conjuntos de datos para el entrenamiento e incorporar aún más tipos de patologías reconocibles e identificables por los modelos.

Un posible uso práctico para una versión mejorada y más generalizada de los modelos pro-

puestos en este trabajo es el desarrollo de una aplicación móvil de uso gratuito para la detección de enfermedades en plantas. Una herramienta como ésta permitirá a los agricultores, especialmente a los agricultores pequeños, simplificar y agilizar el proceso de detección y diagnóstico de enfermedades en sus cultivos significativamente, así reduciendo el riesgo de sufrir pérdidas monetarias causadas por la detección tardía de amenazas y minimizando la dependencia y gastos en especialistas para el diagnóstico. El desarrollo de dicha herramienta permitirá brindar a los agricultores pequeños los beneficios del uso de tecnologías en sus labores a un bajo costo, así cumpliendo los objetivos estipulados en este trabajo.

Si es que se dispone de la colaboración de especialistas expertos tanto en el diagnóstico como en el tratamiento de enfermedades y plagas presentes en cultivos, sería posible ampliar la funcionalidad de la aplicación móvil de detección anteriormente propuesta para también enseñar a los potenciales usuarios las características de cada patología y las medidas que deben tomarse para combatir, erradicar y prevenir la reaparición de las amenazas detectadas, junto con informar las ventajas y desventajas de cada medida propuesta. Esto convertiría la aplicación móvil en una herramienta de detección y también de capacitación, lo que sería de enorme utilidad para agricultores pequeños que carecen de los recursos necesarios para contratar especialistas y del tiempo necesario para capacitarse a fondo.

ANEXOS

Resultados EfficientNet-B1 previo al cambio de configuración

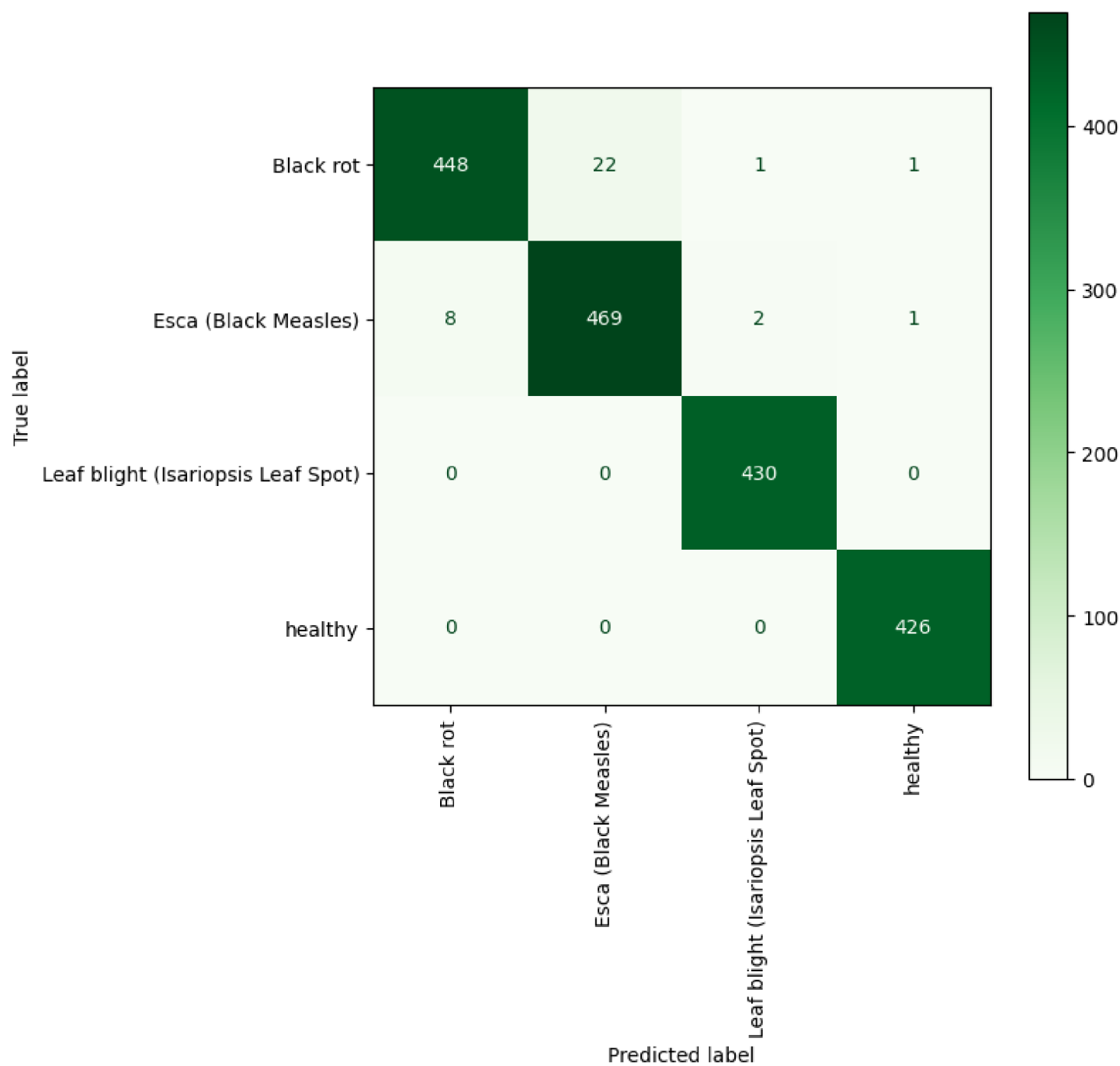


Figura 30: Matriz de confusión para el modelo EfficientNet-B1 previo al cambio de configuración al evaluarlo en el conjunto de prueba de Grape Disease.
Fuente: Elaboración propia.

Tabla 6: Resultados de la evaluación del modelo EfficientNet-B1 previo al cambio de configuración.

Fuente: Elaboración Propia.

	Precision	Recall	F1-Score	Support
Black rot	0.982456	0.949153	0.965517	472
Esca (Black Measles)	0.955193	0.977083	0.966014	480
Leaf blight (Isariopsis Leaf Spot)	0.993072	1.0	0.996524	430
Healthy	0.995327	1.0	0.997658	426
Accuracy	0.980642	0.980642	0.980642	1808
Macro average	0.981512	0.981559	0.981428	1808
Weighted average	0.980776	0.980642	0.980597	1808

REFERENCIAS BIBLIOGRÁFICAS

- [Al Bashish *et al.*, 2010] Al Bashish, D., Braik, M., y Bani-Ahmad, S. (2010). A framework for detection and classification of plant leaf and stem diseases. En *2010 international conference on signal and image processing*, pp. 113–118. IEEE.
- [Ali *et al.*, 2017] Ali, H., Lali, M., Nawaz, M. Z., Sharif, M., y Saleem, B. (2017). Symptom based automated detection of citrus diseases using color histogram and textural descriptors. *Computers and Electronics in agriculture*, 138:92–104.
- [Alpaydin, 2020] Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.
- [Atila *et al.*, 2021] Atila, Ü., Uçar, M., Akyol, K., y Uçar, E. (2021). Plant leaf disease classification using efficientnet deep learning model. *Ecological Informatics*, 61:101182.
- [Bankar *et al.*, 2014] Bankar, S., Dube, A., Kadam, P., y Deokule, S. (2014). Plant disease detection techniques using canny edge detection & color histogram in image processing. *Int. J. Comput. Sci. Inf. Technol*, 5(2):1165–1168.
- [Bhange y Hingoliwala, 2015] Bhange, M. y Hingoliwala, H. (2015). Smart farming: Pomegranate disease detection using image processing. *Procedia Computer Science*, 58:280–288.
- [Boza *et al.*, 2018] Boza, S., Rozas, J., y Rivers, R. (2018). Rechazos de exportaciones por incumplimiento de medidas no arancelarias: el caso de los productos agrícolas latinoamericanos en la frontera de estados unidos. *Estudios internacionales (Santiago)*, 50:37 – 56.
- [Cruz A., 2004] Cruz A., M. (2004). Enfermedades de la vid en el secano interior de la vii y viii regiones de chile. En *Boletín INIA - Instituto de Investigaciones Agropecuarias*.
- [Deng *et al.*, 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., y Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. En *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. IEEE.
- [Erickson y Kitamura, 2021] Erickson, B. J. y Kitamura, F. (2021). Magician’s corner: 9. performance metrics for machine learning models.
- [FAO, 2021] FAO (2021). Año internacional de la sanidad vegetal - informe final. Technical report, FAO.
- [Fuentes, 2021] Fuentes, F. (2021). Visión por computadora para el manejo de plagas y enfermedades en cultivos de papa.
- [Grandini *et al.*, 2020] Grandini, M., Bagli, E., y Visani, G. (2020). Metrics for multi-class classification: an overview. *arXiv preprint arXiv:2008.05756*.
- [He *et al.*, 2016] He, K., Zhang, X., Ren, S., y Sun, J. (2016). Deep residual learning for image recognition. En *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

- [Hughes *et al.*, 2015] Hughes, D., Salathé, M., y other (2015). An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv preprint arXiv:1511.08060*.
- [Khirade y Patil, 2015] Khirade, S. D. y Patil, A. (2015). Plant disease detection using image processing. En *2015 International conference on computing communication control and automation*, pp. 768–771. IEEE.
- [Kiani y Mamedov, 2017] Kiani, E. y Mamedov, T. (2017). Identification of plant disease infection using soft-computing: Application to modern botany. *Procedia computer science*, 120:893–900.
- [Krizhevsky *et al.*, 2012] Krizhevsky, A., Sutskever, I., y Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.
- [LeCun *et al.*, 2015] LeCun, Y., Bengio, Y., y Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.
- [Li *et al.*, 2016] Li, S., Jiao, J., Han, Y., y Weissman, T. (2016). Demystifying resnet. *arXiv preprint arXiv:1611.01186*.
- [Li *et al.*, 2021] Li, Z., Liu, F., Yang, W., Peng, S., y Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*.
- [Liu *et al.*, 2020] Liu, B., Ding, Z., Tian, L., He, D., Li, S., y Wang, H. (2020). Grape leaf disease identification using improved deep convolutional neural networks. *Frontiers in Plant Science*, 11:1082.
- [Mazzeo *et al.*, 2020] Mazzeo, P. L., Libetta, C., Spagnolo, P., y Distante, C. (2020). A siamese neural network for non-invasive baggage re-identification. *Journal of Imaging*, 6(11):126.
- [Oroojlooyjadid *et al.*, 2020] Oroojlooyjadid, A., Snyder, L. V., y Takáč, M. (2020). Applying deep learning to the newsvendor problem. *IIEE Transactions*, 52(4):444–463.
- [O'Shea y Nash, 2015] O'Shea, K. y Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.
- [Puelles T. y Sepúlveda R., 2012] Puelles T., J. y Sepúlveda R., P. (2012). Oídio de la vid. En *Cartilla Divulgativa INIA La Cruz*.
- [Pujari *et al.*, 2015] Pujari, J. D., Yakkundimath, R., y Byadgi, A. S. (2015). Image processing based detection of fungal diseases in plants. *Procedia Computer Science*, 46:1802–1808.
- [Ramesh *et al.*, 2018] Ramesh, S., Hebbar, R., Niveditha, M., Pooja, R., Shashank, N., Vinod, P., y other (2018). Plant disease detection using machine learning. En *2018 International conference on design innovations for 3Cs compute communicate control (ICDI3C)*, pp. 41–45. IEEE.

- [Saleem *et al.*, 2019] Saleem, M. H., Potgieter, J., y Arif, K. M. (2019). Plant disease detection and classification by deep learning. *Plants*, 8(11):468.
- [Sandhu y Kaur, 2019] Sandhu, G. K. y Kaur, R. (2019). Plant disease detection techniques: a review. En *2019 international conference on automation, computational and technology management (ICACTM)*, pp. 34–38. IEEE.
- [Saradhambal *et al.*, 2018] Saradhambal, G., Dhivya, R., Latha, S., y Rajesh, R. (2018). Plant disease detection and its solution using image classification. *International Journal of Pure and Applied Mathematics*, 119(14):879–884.
- [Selvaraju *et al.*, 2017] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., y Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. En *Proceedings of the IEEE international conference on computer vision*, pp. 618–626.
- [Simonyan y Zisserman, 2014] Simonyan, K. y Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- [Singh y Misra, 2017] Singh, V. y Misra, A. K. (2017). Detection of plant leaf diseases using image segmentation and soft computing techniques. *Information processing in Agriculture*, 4(1):41–49.
- [Tan y Le, 2019] Tan, M. y Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. En *International conference on machine learning*, pp. 6105–6114. PMLR.
- [Vujović y other, 2021] Vujović, Ž. y other (2021). Classification model evaluation metrics. *International Journal of Advanced Computer Science and Applications*, 12(6):599–606.
- [Xie *et al.*, 2020] Xie, X., Ma, Y., Liu, B., He, J., Li, S., y Wang, H. (2020). A deep-learning-based real-time detector for grape leaf diseases using improved convolutional neural networks. *Frontiers in plant science*, 11:751.
- [Yamashita *et al.*, 2018] Yamashita, R., Nishio, M., Do, R. K. G., y Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, 9:611–629.
- [Zhuang *et al.*, 2020] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., y He, Q. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76.