

UNIVERSIDAD AUTÓNOMA DE ZACATECAS



COMPARACIÓN DE ARQUITECTURAS DE REDES NEURONALES CONVOLUCIONALES PARA LA CLASIFICACIÓN DE ENFERMEDADES EN TOMATE

Valeria Maeda Gutiérrez

Tesis de Maestría

presentada a la Unidad Académica de Ingeniería Eléctrica
de acuerdo a los requerimientos de la Universidad para obtener el Grado de

MAESTRO EN CIENCIAS DE LA INGENIERÍA

Directores de tesis: Dr. Carlos A. Olvera Olvera y Dr. Carlos E. Galván Tejada

UNIDAD ACADÉMICA DE INGENIERÍA ELÉCTRICA

13 de septiembre de 2019

COMPARACIÓN DE ARQUITECTURAS DE REDES NEURONALES CONVOLUCIONALES PARA LA CLASIFICACIÓN DE ENFERMEDADES EN TOMATE

Valeria Maeda Gutiérrez

Directores de tesis: Dr. Carlos A. Olvera Olvera y Dr. Carlos E. Galván Tejada

RESUMEN

Las plantas de tomate son constantemente afectadas por diversas enfermedades. Un diagnóstico oportuno y preciso es de suma importancia para prevenir la calidad de los cultivos. Recientemente, el Aprendizaje Profundo, específicamente las Redes Neuronales Convolucionales han logrado resultados extraordinarios en muchos campos, incluyendo la clasificación de enfermedades de plantas. Este trabajo se centra en la técnica de “ajuste fino” basado en la comparación de diferentes arquitecturas de Redes Neuronales Convolucionales tales como: AlexNet, GoogleNet, Inception V3, ResNet 18 y ResNet 50. Finalmente, se realizó la evaluación de la comparación. El conjunto de datos utilizado para los experimentos fue compuesto por nueve clases con enfermedades y una clase sana, extraídas del conjunto de datos PlantVillage. Los modelos se evaluaron mediante un análisis estadístico multiclase con base en la exactitud, precisión, sensibilidad, especificidad, F-score, área bajo la curva y la curva de característica operativa del receptor. Los resultados presentaron valores significativos obtenidos por GoogleNet con 99.72% de AUC y 99.12% de sensibilidad. Es posible concluir que esta tasa de éxito significativo hace que el modelo de GoogleNet sea una herramienta útil para los agricultores, ayudando así a identificar y proteger los tomates de enfermedades.

Palabras clave: Enfermedades en tomate; Aprendizaje Profundo; Redes Neuronales Convolucionales; Clasificación.

COMPARISON OF CONVOLUTIONAL NEURAL NETWORK ARCHITECTURES FOR CLASSIFICATION OF TOMATO PLANT DISEASES

Valeria Maeda Gutiérrez

Thesis supervisors: Dr. Carlos A. Olvera Olvera y Dr. Carlos E. Galván Tejada

ABSTRACT

Tomato plants are highly affected by diverse diseases. A timely and accurate diagnosis plays an important role to prevent the quality of crops. Recently, Deep Learning, specifically, Convolutional Neural Networks (CNNs) have achieved extraordinary results in many applications, including the classification of plant diseases. This work is focused on fine-tuning based on the comparison of the state of the art architectures: AlexNet, GoogleNet, Inception V3, ResNet 18 and ResNet 50. An evaluation of the comparison is finally performed. The dataset used for the experiments is contained by nine different classes of tomato diseases and a healthy class from *PlantVillage*. The models were evaluated through a multiclass statistical analysis based on accuracy, precision, sensitivity, specificity, F-score, Area Under the Curve (AUC) and Receiving Operating Characteristic (ROC) curve. The results present significant values obtained by the GoogleNet technique with 99.72% of AUC and 99.12% of sensitivity. It is possible to conclude that this significant success rate makes the GoogleNet model a useful tool for farmers, helping to identify and protect tomatoes from the diseases mentioned.

Keywords: Tomato plant diseases; Deep Learning; Convolutional Neural Networks; Classification.

Dedicada a mis padres, hermano y Angélica.

Agradecimientos

A lo largo de este proceso, que significo un gran reto para mí, no fueron pocas las personas que estuvieron a mi lado. A cada una de ellas tengo algo que agradecerle, pues las palabras cálidas, los consejos y los nuevos aprendizajes que me otorgaron, fueron siempre un impulso para llegar hasta el día de hoy, en el que el esfuerzo por fin se ve recompensado.

A mis padres y hermano

Por estar para mí cada día, aún y cuando la distancia no siempre nos permitió estar en el mismo lugar. Gracias por su amor y motivación.

Al Dr. Carlos A. Olvera Olvera

Quien hizo esto posible. Gracias por la oportunidad que me brindo, desde aquel verano científico y por aquella invitación a este proyecto, por su apoyo y guía en cada paso de esta meta, porque gracias a todo eso es que he crecido profesional y personalmente.

Al Dr. Calos E. Galván Tejada

Agradezco la paciencia, la ayuda que siempre estuvo dispuesto a brindarme, por disipar cada duda que tenía, y por esa extraordinaria oportunidad de realizar una estancia en el Hospital Siglo XXI.

A los docentes

Quienes formaron parte de mi desarrollo, por todas sus enseñanzas, que marcaron siempre un punto clave en este proyecto, gracias por compartir sus conocimientos y hacer de esta investigación lo que es hoy en día.

Contenido General

	Pag.
Resumen	i
Abstract	ii
Lista de figuras	viii
Lista de tablas	x
Nomenclatura	xii
1 Introducción	1
1.1 Planteamiento del Problema	1
1.2 Justificación	3
1.3 Objetivos	4
1.3.1 Objetivo General	4
1.3.2 Objetivos Específicos	5
1.4 Hipótesis	5
1.5 Estado del Arte	5
1.6 Estructura de la tesis	12
2 Redes Neuronales Convolucionales	13
2.1 Aprendizaje Profundo	13
2.2 Redes Neuronales Convolucionales	14
2.2.1 Inspiración biológica	14
2.3 Estructura de CNNs	15
2.3.1 Convolución	15
2.3.2 Capa de entrada	15
2.3.3 Capa de convolución	15
2.3.4 Capa de agrupación	16
2.3.5 Capa completamente conectada	16
2.4 Arquitecturas de CNN	17
2.4.1 AlexNet	17
2.4.2 GoogleNet	18
2.4.3 Inception V3	18
2.4.4 Residual Networks (ResNet)	18

	Pag.
2.5 Enfermedades de la planta de tomate	20
2.5.1 Bacterianas	22
2.5.2 Virales	23
2.5.3 Hongos y oomicetos	24
2.5.4 Insectos	27
3 Materiales y métodos	28
3.1 Adquisición de datos	29
3.1.1 <i>PlantVillage - Dataset</i>	29
3.2 Configuración de los modelos	31
3.3 Entrenamiento de los modelos	32
3.3.1 Transferencia de Aprendizaje	32
3.3.2 Clasificación	34
3.4 Medidas de desempeño	35
3.4.1 Matriz de confusión	35
3.4.2 Matriz de confusión multiclase	36
3.4.3 Exactitud	37
3.4.4 Precisión	37
3.4.5 Sensibilidad	37
3.4.6 Especificidad	38
3.4.7 F-score	38
3.4.8 Área bajo la curva ROC	38
3.5 Implementación	40
3.5.1 Características del área de trabajo	40
4 Comparativa de arquitecturas de CNN para la clasificación de enfermedades en tomate	41
4.1 Resultados y discusión	41
4.1.1 Selección de subconjuntos	46
Conclusiones	58
Referencias	60

Lista de figuras

Figura	Pag.
2.1 Diferentes categorías de enfermedades en tomate.	21
2.2 Muestra ilustrativa de la enfermedad mancha bacteriana.	22
2.3 Muestra ilustrativa de la enfermedad virus del mosaico.	23
2.4 Muestra ilustrativa de la enfermedad rizado amarillo del tomate.	24
2.5 Muestra ilustrativa de la enfermedad tizón temprano.	25
2.6 Muestra ilustrativa de la enfermedad tizón tardío.	25
2.7 Muestra ilustrativa de la enfermedad mancha negra.	26
2.8 Muestra ilustrativa de la enfermedad mancha de la hoja.	26
2.9 Muestra ilustrativa de la enfermedad araña roja.	27
3.1 Diagrama de flujo de la metodología seguida.	28
3.2 Imágenes de muestra obtenidas de <i>PlantVillage - Dataset</i>	29
3.3 Representación de transfer learning.	33
3.4 Flujo de trabajo de una CNN.	34
3.5 Matriz de confusión.	35
3.6 Matriz de confusión multiclase.	36
4.1 Exactitud vs iteraciones.	43
4.2 Matriz de confusión de GoogleNet.	43
4.3 Curva ROC.	45

Figura	Pag.
4.4 Curva de aprendizaje.	56

Lista de tablas

Tabla	Pag.
1.1 Resumen de técnicas y comparación entre los estudios de clasificación de diversas enfermedades de plantas.	11
2.1 Propiedades de las CNN pre-entrenadas.	19
3.1 Detalles del conjunto de datos.	30
3.2 Valor de hiperparámetros en los experimentos.	31
3.3 Características de la computadora.	40
4.1 Medidas de rendimiento (%) para cada modelo pre-entrenado.	41
4.2 Desempeño de cada clase.	44
4.3 Subconjuntos	46
4.4 Subconjunto 1.	47
4.5 Medidas de rendimiento para el subconjunto 1.	47
4.6 Subconjunto 2.	48
4.7 Medidas de rendimiento para el subconjunto 2.	48
4.8 Subconjunto 3.	49
4.9 Medidas de rendimiento para el subconjunto 3.	49
4.10 Subconjunto 4.	50
4.11 Medidas de rendimiento para el subconjunto 4.	50
4.12 Subconjunto 5.	51

Tabla	Pag.
4.13 Medidas de rendimiento para el subconjunto 5.	51
4.14 Subconjunto 6	52
4.15 Medidas de rendimiento para el subconjunto 6.	52
4.16 Subconjunto 7.	53
4.17 Medidas de rendimiento para el subconjunto 7.	53
4.18 Subconjunto 8.	54
4.19 Medidas de rendimiento para el subconjunto 8.	54
4.20 Subconjunto 9.	55
4.21 Medidas de rendimiento para el subconjunto 9.	55
4.22 Resultados generales de los modelos con base al <i>AUC</i>	56

Nomenclatura

<i>ANN</i>	Redes Neuronales Artificiales
<i>AUC</i>	Área Bajo la Curva
<i>CNN</i>	Redes Neuronales Convolucionales
<i>DL</i>	Aprendizaje Profundo
<i>GPU</i>	Unidad de Procesamiento Gráfico
<i>ML</i>	Aprendizaje Automático
<i>ROC</i>	Característica Operativa del Receptor
<i>TL</i>	Transferencia de Aprendizaje

Capítulo 1

Introducción

1.1 Planteamiento del Problema

El tomate rojo ocupa un lugar notable en la economía agrícola mexicana. De hecho, México es el primer proveedor mundial con una participación en el mercado internacional de 25.11% del valor de las exportaciones mundiales [1]. De acuerdo con el *Servicio de Información Agroalimentaria y Pesquera (SIAP)*, se estima que las exportaciones de tomate crecerán hasta 3.84 millones de toneladas en 2024. Del mismo modo, el estado de Zacatecas, ubicado en el centro-norte de la República Mexicana, es fundamental en la producción del tomate, dado que ocupa el quinto puesto de las principales entidades productoras [2]. Durante el año 2018 fueron sembradas 13,302 hectáreas de las cuales 165 fueron afectadas por enfermedades, plagas y cambios climáticos.

La producción del tomate rojo se ve afectada por enfermedades que son causadas por bacterias, hongos, oomicetos, arácnidos y virus que son responsables de millones de dólares en pérdidas de producción. Además de causar grandes daños a los cultivos disminuyendo su producción, ocasionan una mala calidad de los productos, limitan la disponibilidad de alimentos y amenazan la seguridad alimentaria [3, 4]. Por lo cual, para lograr un desarrollo sostenible adecuado existe la necesidad de minimizar pérdidas económicas y a su vez de producción de manera que la identificación de enfermedades es un punto clave para prevenir estos problemas.

Proporcionar un monitoreo temprano es esencial para elegir un tratamiento correcto y evitar la propagación de enfermedades [5]. Existen distintos procedimientos para su identificación y detección. Uno de ellos es la inspección visual, este método tradicional es realizado por agricultores [6]. A pesar de ello, esta tarea requiere seguimiento continuo puesto que pueden diagnosticar erróneamente los síntomas debido a que los juzgan en base a sus experiencias, solicita gran cantidad de tiempo y no es eficiente en cultivos grandes. Detectar enfermedades de plantas en un laboratorio es otra opción, este proceso se basa en obtener muestras del campo, llevando a cabo experimentos de diagnóstico y evaluaciones microscópicas como ELISA y PCR. Si bien son métodos precisos, también son altamente costosos [7]. Considerando el acceso restringido a recursos y experiencia limitada, se encuentra la necesidad de realizar procesos automáticos.

El diagnóstico oportuno de enfermedades de plantas se ha convertido en una preocupación para los agricultores. Debido a lo anterior, en busca de la protección de plantas, los investigadores han contribuido significativamente mediante el uso de tecnologías como *Aprendizaje Automático* (ML, por sus siglas en inglés), *Procesamiento Digital de Imágenes*, *Visión Computacional*, *Algoritmos Genéticos* y *Aprendizaje Profundo* (DL, por sus siglas en inglés)[8]. Estas áreas han dado posibles soluciones en el campo de la agricultura, no obstante, la investigación en base a enfermedades de plantas (no solo en cultivos de tomate) sigue siendo inmadura y requiere de atención para producir sistemas prácticos. Una de las herramientas de procesamiento de datos exitosas aplicadas a un gran conjunto de datos son las *Redes Neuronales Convolucionales* (CNNs, por sus siglas en inglés), son una clase de *Redes Neuronales Artificiales* (ANN, por sus siglas en inglés) profundas y avanzadas, que tienen muchas aplicaciones, las cuales incluyen tareas complejas como la clasificación de imágenes y el reconocimiento de objetos que han brindado la oportunidad de mejorar la precisión de la clasificación en muchos campos, incluida la agricultura. La mayoría de las arquitecturas profundas no se aplican para esta tarea. Por otro lado, estos clasificadores sufren una falta de interpretabilidad debido a que se consideran como “cajas negras” que dan buenos resultados pero sin explicación alguna. Una alta exactitud del modelo no es suficiente para una buena clasificación, es necesario aplicar el

uso de métricas que determinen el comportamiento del clasificador.

Por lo anteriormente mencionado, la presente investigación propone la “*Comparativa de arquitecturas de redes neuronales convolucionales para la clasificación de enfermedades en tomate*” con el objetivo de determinar cuál es la arquitectura que mejor modela el problema y logra una buena clasificación de enfermedades en hojas del cultivo de tomate, validándola por medio de diferentes métricas. De esta manera, los modelos de CNNs pueden ser implementados como un sistema automatizado diseñado para identificar enfermedades de la planta, lo que podría ser de gran ayuda para técnicos y expertos en agricultura.

1.2 Justificación

Uno de los grandes desafíos para el sector agrícola son las enfermedades de plantas. La identificación precisa y rápida podría ayudar a desarrollar un tratamiento temprano y reducir sustancialmente pérdidas económicas. El reconocimiento de los síntomas visuales es esencial para el diagnóstico de las enfermedades de plantas. Sin embargo, este método es demasiado subjetivo. Los cultivos hortícolas representan un importante segmento económico-social de la producción a nivel mundial. La *Organización de las Naciones Unidas para la Alimentación y la Agricultura* (ONUAA) estima que las enfermedades, plagas y las malas hierbas causan aproximadamente el 25% de las fallas en los cultivos [9].

En contexto con el presente trabajo, la producción del cultivo de tomate aumenta continuamente y por ende su comercialización, la cual genera millones de empleos de manera directa e indirecta. En México su producción anual creció en un 4.8% entre los años 2006 - 2016 [10]. De igual manera, nuestro país se ha posicionado como el principal proveedor a nivel mundial. Durante el año 2016 la producción nacional se centró en cinco entidades, ubicándose Zacatecas en el quinto lugar con un 5.7% detrás Baja California (6.7%), Michoacán (7.0%), San Luis Potosí (9.2%) y Sinaloa, primer lugar, con un 27.6% [11]. A pesar de ello, es una de las hortalizas que presenta mayores pérdidas, de hasta un 50% total de su producción. A

nivel nacional esto simboliza un tema de alta importancia por el porcentaje de pérdidas que se llegan a dar, uno de los principales factores que afectan su producción son las enfermedades de plantas; un ejemplo de ello es el estado de Sonora, en el que se ha registrado pérdidas de hasta un 100% [12].

De acuerdo con la *Secretaría de Agricultura y Desarrollo Rural* (SAGARPA) en el año 2018 en el estado de Zacatecas hubo un total de 165 hectáreas pérdidas por siniestralidad, lo cual significa afectación por enfermedades o plagas [13]. Esto no sólo se traduce a pérdidas monetarias, sino también afecta a la calidad y cantidad de los productos.

Dado el impacto que causan las enfermedades de plantas a nivel mundial, una identificación temprana es un enfoque desafiante. Gracias a los avances de desarrollo de Hardware y a la mejora de los métodos de aprendizaje, las CNNs están logrando progresos significativos en la resolución de problemas. Debido a esto, han surgido varias arquitecturas poderosas como AlexNet [14], GoogleNet [15], Inception V3 [16], ResNet [17], y DenseNet [18]. Actualmente, estos modelos se han empleado recientemente en la agricultura, y han logrado obtener éxito y popularidad. En resumen, uno de los objetivos de la investigación es introducir CNNs como enfoque de alto potencial para abordar la problemática de las enfermedades de plantas en tomate, comparando modelos de CNN con el fin de encontrar el modelo que mejor desempeñe dicha tarea.

1.3 Objetivos

1.3.1 Objetivo General

Comparar y evaluar el desempeño de las arquitecturas de CNNs para la clasificación de enfermedades en tomate utilizando imágenes digitales.

1.3.2 Objetivos Específicos

- Construir diferentes modelos de CNNs utilizando imágenes de enfermedades en tomate para su clasificación.
- Comparar el desempeño de los modelos de CNNs.
- Evaluar los modelos de CNNs mediante diferentes métricas.
- Analizar y evaluar el modelo que obtuvo mejor desempeño.
- Determinar los requisitos mínimos para la cantidad de datos, por ejemplo, el número de imágenes necesarias para la clasificación precisa de las enfermedades en tomate.

1.4 Hipótesis

La comparación y evaluación en términos de: área bajo la curva, sensibilidad, especificidad, precisión, F-score y exactitud, de las arquitecturas de redes neuronales convolucionales para abordar el problema de la clasificación de enfermedades en plantas de tomate, permiten la selección de la arquitectura óptima para la clasificación automática de estas enfermedades.

1.5 Estado del Arte

En los últimos años, diversas soluciones agrícolas se han desarrollado en función de CNNs. De acuerdo con Gikunda y Jouandeu [19], la implementación de CNNs en “granjas inteligentes” se clasifica de la siguiente manera:

- Gestión del ganado, donde se abordan soluciones para la producción ganadera y bienestar animal.
- Gestión del medio ambiente.
- Gestión de la planta, la cual brinda soluciones orientadas al bienestar de los cultivos y su producción. Esta abarca clasificación, detección y predección (especies, enfermedades, plagas y producción de rendimiento).

Una serie de trabajos han sido desarrollados mediante el uso e implementación de CNNs que han mostrado resultados significativos en el estado del arte, en particular en la áreas de detección, identificación y clasificación de enfermedades. Sin embargo, es notable que la mayoría de las investigaciones hayan sido publicadas después del 2015, esto indica qué tan reciente y moderna es esta técnica en el dominio de la agricultura.

Basic study of automated diagnosis of viral plant diseases using convolutional neural networks.

En el trabajo de Kawasaki et al. [20], se propone un sistema de detección de enfermedades de pepinos basado en CNN, el cual utiliza el marco de aprendizaje Caffe [21]. Bajo la estrategia de validación cruzada cuádruple, el sistema logra una exactitud promedio de 94.9% al clasificar pepinos. Los autores concluyeron que el modelo solamente utilizando imágenes de entrenamiento puede adquirir automáticamente las características necesarias para la clasificación y alcanzar un alto rendimiento.

Using deep learning for image-based plant disease detection.

Por otro lado, Mohanty et al. [22], evaluaron dos arquitecturas de CNN: AlexNet y GoogleNet para identificar 14 especies de cultivos y 26 enfermedades mediante el uso del conjunto de datos *PlantVillage*. Emplearon diferentes distribuciones de prueba para su entrenamiento con el objetivo de medir el rendimiento de las arquitecturas: 80-20, 60-40, 50-50,40-60 y 20-80%. Finalmente, el desempeño de los modelos se basó en la capacidad de predecir el par de enfermedades correcto del cultivo. GoogleNet logró el mejor rendimiento, alcanzando una puntuación de F-score de 0.9934 y 99.35% de exactitud general.

Deep neural network based recognition of plant diseases by leaf image classification.

En la investigación de Sladojevic et al. [23], se desarrolló un modelo de reconocimiento de enfermedades de plantas basado en la clasificación de imágenes de hojas de plantas por medio del uso de CNN. Su modelo fue capaz de reconocer 13 tipos diferentes de enfermedades. Además,

utilizaron Caffe como marco de aprendizaje profundo, obteniendo como resultado experimental una exactitud final de 96.3%.

Classification of apple tree disorders using convolutional neural networks.

Igualmente, Natchtigall et al. [24], emplearon CNNs para detectar y clasificar daños y deficiencias nutricionales en manzanos a partir de imágenes de hojas. Elaboraron su conjunto de datos recolectando hojas sanas, dos clases de daños causados por desequilibrios nutricionales, dos clases de enfermedades y una clase de daños por herbicidas. AlexNet fue utilizado como arquitectura y posteriormente, realizaron una comparación con otro modelo: Multilayer Perceptron (MLP). Asimismo, diseñaron un experimento para evaluar a siete expertos voluntarios. Los resultados permitieron concluir que la arquitectura de CNN logra mejor desempeño alcanzando 97.3% de exactitud, en seguida de los expertos (96%) y finalmente, MLP con el menor resultado, logrando 77.3%.

Basic investigation on a robust and practical plant diagnostic system.

Otro enfoque para la clasificación de enfermedades en la hoja de pepino utilizando CNN fue el propuesto por Fujita et al. [25], su base de datos fue compuesta por 7,520 imágenes de hojas sanas y enfermas. Sin embargo, la mitad de las imágenes utilizadas se tomaron en malas condiciones, pero a pesar de ello, su sistema logró 82.3% de exactitud bajo la estrategia de validación cruzada cuádruple.

Deep learning for tomato diseases: classification and symptoms visualization.

En la investigación de Brahimi et al. [4], emplean CNNs como algoritmo de aprendizaje para clasificar enfermedades de tomate, utilizaron dos arquitecturas, las cuales fueron: AlexNet y GoogleNet. Las conclusiones de este trabajo pudieron sugerir que el enfoque de DL supera a otras técnicas de clasificación, por ejemplo: Bosques Aleatorios (RF, por sus siglas en inglés), Máquinas de Vectores de Soporte (SVM, por sus siglas en inglés), los cuales fueron comparados con las CNNs. De igual manera, calcularon para cada modelo métricas como exactitud, macro precisión, macro F-score y macro recall. Su modelo obtuvo 99.18% de exactitud.

Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning.

DeChant et al. [26], diseñaron un sistema, el cual fue capaz de identificar automáticamente la enfermedad “tizón de la hoja del maíz del norte”. Utilizaron CNNs que fueron entrenadas para clasificar pequeñas regiones que contenían o no lesiones; las predicciones fueron combinadas en mapas de calor separados y luego se alimentó una CNN final, que fue entrenada para clasificar la imagen completa, la cual contenía imágenes de la planta sanas y enfermas. En conclusión el sistema obtuvo una exactitud de 96.7% en las imágenes de prueba del sistema.

Identification of rice diseases using deep convolutional neural networks.

Por otra parte, en la investigación de Lu et al. [27], realizaron un método para la identificación de enfermedades de arroz basado en CNN, siendo AlexNet la arquitectura empleada para construir el clasificador. Bajo la estrategia de validación cruzada de 10 veces, el modelo logró 95.48% de exactitud. Por otro lado, compararon el resultado obtenido con modelos poco profundos como SVM que alcanzó 91%, el método de propagación hacia atrás (backpropagation, BP) que logró 92% y por último, el método de optimización por enjambre de partículas (particle swarm optimization, PSO) con el 88%. Además, demostraron que el enfoque de DL logra mejores resultados que los modelos convencionales de ML.

Automatic image-based plant disease severity estimation using deep learning.

En el trabajo de Wang et al. [28], utilizaron imágenes de manzana obtenidas del conjunto de datos de *PlantVillage*. Los autores entrenaron una serie de CNNs para diagnosticar la gravedad de la enfermedad. Los modelos fueron entrenados mediante el método de transferencia de aprendizaje. El modelo que mejor desempeño obtuvo fue VGG 16 debido a que arrojó 90.4% de exactitud general en el conjunto de prueba.

Deep learning for plant diseases: detection and saliency map visualization.

En la investigación de Brahim et al. [29], se realizó una comparación de múltiples CNN, tales como: AlexNet, DenseNet 169, Inception V3, ResNet 34, SqueezeNet 1.1 y VGG 13, para clasificar enfermedades de plantas aplicando tres estrategias diferentes. Dos de ellas se basan en la transferencia de aprendizaje, mientras que el otro método, hace referencia a entrenar la CNN desde cero (scratch) a partir de una configuración aleatoria de pesos. El conjunto de datos fue dividido en 80-20% (entrenamiento - prueba). El modelo que mejor desempeño alcanzó fue Inception V3 con 99.76% de exactitud. Finalmente, concluyeron que la estrategia de aprendizaje más exitosa fue la de transferencia de aprendizaje.

Es importante señalar que una de las principales diferencias entre este trabajo y el que se propone, son las CNNs implementadas para la comparación, puesto que, en base a los resultados de este trabajo, la selección de CNN se ha limitado a solo tres de las que se utilizaron, incluyendo a GoogleNet y otras versiones de ResNet que han presentado resultados significativos en la literatura.

CNN transfer learning for automatic image-based classification of crop disease.

Wang et al. [30], utilizaron un conjunto de datos de 2,430 imágenes que fueron tomadas de un entorno natural, las cuales contienen dos especies de cultivos y ocho enfermedades. Con el objetivo de abordar la problemática del sobreajuste causado por su pequeño conjunto de datos, los autores analizaron el rendimiento de su entrenamiento por scratch utilizando el método de transferencia de aprendizaje. En el caso de la transferencia de aprendizaje, primeramente entrenaron con el conjunto de datos de *PlantVillage* con el fin de obtener un modelo pre-entrenado y después realizaron un nuevo entrenamiento con su conjunto de datos, con tal de ajustar los parámetros. Asimismo, la CNN utilizada comprendida de cinco capas convolucionales, logró 90.84% de exactitud. Finalmente, los resultados experimentales demostraron que la combinación de CNN y la transferencia de aprendizaje es efectiva para la clasificación de enfermedades de plantas con datos a pequeña escala.

Tomato crop disease classification using pre-trained deep learning algorithm.

Igualmente, Rangarajan et al. [31], entrenaron AlexNet y VGG 16 con el fin de clasificar seis enfermedades y una clase saludable de tomate. El rendimiento de los modelos fue evaluado modificando el número de imágenes, configurando diferentes tamaños de lote, variando los pesos y la tasa de aprendizaje. Con base a esto, concluyeron que AlexNet proporciona una buena exactitud con mejor tiempo de ejecución a comparación de VGG 16.

Cabe señalar que, tomando en cuenta que en este trabajo también realiza la clasificación de enfermedades presentes en las plantas de tomate, el desarrollo de la metodología propuesta se basa en los resultados informados por esta comparación, lo que permite el apoyo en la delimitación del trabajo y selección de arquitecturas para implementar, pudiendo descartar la implementación de VGG 16 debido a las desventajas que presenta en comparación con AlexNet, especialmente en el costo computacional.

Analysis of transfer and residual learning for detecting plant diseases using images of leaves.

En el trabajo de Khandelwal y Raman [32], se abordó la problemática de la detección de enfermedades de plantas. Inception V3 y ResNet 50 fueron los modelos asignados para esta tarea. El conjunto de datos comprendía de 86,198 imágenes, de 25 cultivos y 57 clases. Finalmente, su ResNet 50 fue capaz de lograr 99.374% de exactitud promedio significativamente alta, por medio de la técnica de transferencia de aprendizaje, con una división estándar de 80-20% (entrenamiento - prueba), además, concluyeron que ResNet 50 supera a Inception V3, esto fue visible en términos de mayor exactitud en las primeras épocas, así como una exactitud general al final del experimento.

Con base a los trabajos relacionados, se decidió limitar la inclusión de CNN. AlexNet, GoogleNet, Inception V3, ResNet18 y ResNet50, fueron seleccionadas debido al comportamiento significativo con el que han sido reportadas y avaladas por sus resultados.

A continuación, la Tabla 1.1 muestra el estado del arte de las CNNs en la agricultura, en particular en el área de la clasificación y detección de enfermedades de plantas.

Tabla 1.1 Resumen de técnicas y comparación entre los estudios de clasificación de diversas enfermedades de plantas.

Autores	Arquitectura (s)	Métricas	Resultado
Kawasaki, Y. et al. - 2015	Personalizada	Exactitud, Sensitividad, Especificidad	94.90%
Sladojevic, S. et al. - 2016	CaffeNet	Exactitud	96.30%
Mohanty, S.P. et al. - 2016	AlexNet, GoogleNet	F1-score, Precisión, Recall, Exactitud	99.34%
Nachtigall, L.G. et al. - 2016	AlexNet	Recall, Precisión, Exactitud	97.30%
Fujita, E. et al. - 2016	Personalizada	Sensitividad, Exactitud, Especificidad	82.30%
Brahimi, M. et al. - 2017	AlexNet, GoogleNet	Exactitud, Macro-Precisión Macro-Recall	99.18%
DeChant, C. et al. - 2017	Personalizada	Exactitud, Precisión, Recall, F1-score	96.70%
Lu, Y. et al. - 2017	AlexNet	Exactitud	95.48%
Wang, G. et al. - 2017	VGG 16, VGG 19 Inception V3, ResNet 50	Exactitud	90.4%
Brahimi, M. et al. - 2018	AlexNet, DenseNet 169, Inception V3, ResNet 34, SqueezeNet 1.1, VGG 13	Exactitud	99.76%
Wang, J. et al. - 2018	Personalizada	Exactitud	90.84%
Rangarajan A.K. et al. - 2018	AlexNet, VGG 16	Exactitud	97.49%
Khandelwal, I. et al. - 2018	Inception V3, ResNet 50	Exactitud	99.374%

1.6 Estructura de la tesis

A continuación se describe la estructura en la que se organiza la presente investigación.

Capítulo 2. Redes neuronales convolucionales.

Se presenta una amplia descripción de los temas de referencia requeridos para la realización de la investigación, con el objetivo de facilitar al lector la correcta comprensión de este documento.

Capítulo 3. Materiales y métodos.

Este capítulo describe de manera detallada la metodología del trabajo de investigación, en donde se explican las técnicas, herramientas y recursos utilizados.

Capítulo 4. Comparativa de arquitecturas de CNN para la clasificación de enfermedades en tomate.

Se muestran las conclusiones y se discuten los resultados obtenidos mediante la comparación y evaluación de arquitecturas.

Capítulo 2

Redes Neuronales Convolucionales

2.1 Aprendizaje Profundo

¿Cómo es posible que un cerebro lento y pequeño, ya sea biológico o electrónico, perciba, comprenda, prediga y manipule un mundo mucho más grande y complicado que él mismo?

Esta frase alude a ideas sobre cómo los conceptos de aprendizaje se inspiraron en procesos y algoritmos descubiertos en la naturaleza.

El DL, es un subconjunto del ML, que a su vez es subcampo de la Inteligencia Artificial (AI). La disciplina del ML generalmente emplea modelos de entrenamiento que usan datos para predecir, clasificar, y detectar datos no vistos. El DL, puede entenderse como un marco general para modelos construidos por múltiples capas de procesamiento que le permiten aprender representaciones jerárquicas en diferentes niveles de abstracción [33]. El paradigma del DL, ha mejorado significativamente el estado del arte en tareas que resultan difíciles de llevar a cabo por una máquina. Evidentemente, no es un concepto nuevo, sino que ha experimentado durante el paso del tiempo mejoras. Se pueden identificar tres factores importantes, uno de ellos es el aumento de datos que, en general se puede ver es una de la escala que impulsa el DL. Por otro lado, el tamaño de los modelos que pueden procesar los datos de manera efectiva, por lo tanto

requiere de mucha potencia computacional, lo que ocasionó el creciente rendimiento computacional y específicamente el progreso de las Unidades de Procesamiento Gráfico (GPU, por sus siglas en inglés).

2.2 Redes Neuronales Convolucionales

Una CNN, es uno de los algoritmos más populares para el DL, es un tipo de aprendizaje automático en el que un modelo aprende a realizar tareas de clasificación directamente desde imágenes, vídeo, texto o sonido. Son particularmente útiles para encontrar patrones en imágenes para reconocer objetos, caras, escenas. Aprenden directamente de los datos de la imagen usando patrones para clasificar las imágenes y eliminando la necesidad de la extracción manual de características.

Presentadas por LeCun et al. [34], a principios de la década de los noventa, las redes neuronales convolucionales son un ejemplo de una arquitectura neuronal especializada, la cual incluye conocimiento sobre la invariancia de formas bidimensionales utilizando patrones de conexión local y con restricción en los pesos. Cuentan con una arquitectura multicapa, donde cada capa está constituida por un número determinado de convoluciones con funciones de activación, ya sea ReLu para poder obtener los resultados. El tipo de conexiones de las neuronas de la CNN está inspirado en la organización de la corteza visual de los animales, las cuales se pueden representar de manera matemática como una operación de convolución.

2.2.1 Inspiración biológica

Las CNNs son reconocidas por estar inspiradas en la corteza visual. La corteza visual del cerebro que se localiza en el lóbulo occipital, mismo que está situado en la parte trasera del cráneo, juega un papel importante en el procesamiento de la información visual. La información visual que viene de los ojos pasa a través de una serie de estructuras del cerebro y alcanza a la corteza visual. Las partes de la corteza visual que reciben las entradas sensoriales se conocen como “corteza visual primaria” o bien, área $V1$. La información sensorial se

maneja mejor mediante áreas visuales relacionadas con el reconocimiento de objetos. Luego, la circunvolución oral inferior, es uno de los niveles más altos del flujo ventral del procesamiento visual, asociado con la representación de características complejas de objetos, como la forma global o bien la percepción de la cara [35].

2.3 Estructura de CNNs

2.3.1 Convolución

La convolución es una operación matemática que trabaja con dos funciones (f y g) y produce una tercera (h), que es una integral que expresa la cantidad de superposición de una función a medida que se desplaza sobre la otra función. Denotada por la Ecuación (2.1).

$$h = f * g = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau \quad (2.1)$$

2.3.2 Capa de entrada

Respecto a las tareas de clasificación de objetos, la capa de entrada se representa con mayor frecuencia como una cuadrícula tridimensional cuyas dimensiones dependen de:

$$I = W * H * D \quad (2.2)$$

donde I es el tamaño de la capa de entrada, W corresponde al ancho de la imagen, H es la altura de la imagen de entrada, y finalmente D que es la profundidad o el número de canales de la imagen.

2.3.3 Capa de convolución

Una capa convolucional de una red neuronal es una de las capas principales y está diseñada para resaltar las características de la imagen y su transformación. Más adelante, se utilizan capas más profundas para obtener características más complejas y, en última instancia, determinar la clase del objeto reconocido. Este tipo de capas son los llamados filtros multidimensionales que representan matrices de pesos de conexión de neuronas de la capa anterior con neuronas

de la capa convolucional. Se llaman filtros porque la operación de convolución es muy similar a la operación de filtración de imagen. Cada valor de la señal de neurona de la capa anterior, ubicada en un área determinada correspondiente al filtro central, se multiplica por el valor correspondiente del núcleo del filtro en CNN. Los valores del núcleo del filtro se denominan los pesos del acoplamiento neuronal de la capa convolucional.

2.3.4 Capa de agrupación

La agrupación es una transformación no lineal de la entrada, que da como resultado un mapa de entidades submuestreado espacialmente, o en todas las dimensiones si se agrupan canales cruzados. El objetivo de la agrupación sirve para mantener el control sobre el tamaño de las representaciones e imponer la regularización en la red. Las dos operaciones de agrupación más comunes son la agrupación promedio, donde la unidad de salida será el promedio matemático de las unidades vecinas, y la agrupación máxima, donde la unidad de salida será el valor máximo en la región. Espacialmente, la operación de agrupación transforma los mapas de características de tal manera que serán invariables localmente a las transformaciones.

2.3.5 Capa completamente conectada

Esta capa se puede utilizar como la capa de salida de la CNN, lo que da como resultado la probabilidad de que una imagen de entrada pertenezca a una determinada clase. Las dimensiones del volumen de salida son $[1 * 1 * N]$, donde N es el número de clases de salida que evalúan.

2.4 Arquitecturas de CNN

Son arquitecturas comúnmente conocidas que se usan en la práctica como extractores de características, que luego se utilizan para tareas correspondientes como clasificación y detección. De igual manera, para estas arquitecturas a menudo es posible obtener pesos pre-entrenados, lo cual es útil si se cuentan con pequeños conjuntos de datos o se desea acelerar el entrenamiento. A continuación, se presentan las arquitecturas de CNN empleadas en la realización de la investigación.

2.4.1 AlexNet

AlexNet fue propuesta por Alex Krizhevsky et al [14]. En el año 2012, el modelo ganó el desafío más complejo, donde el Reconocimiento Visual a Gran Escala (*Large Scale Visual Recognition (ILSVRC)*) de ImageNet [36] que evalúa los algoritmos para la detección de objetos y clasificación de imágenes a gran escala. AlexNet contiene 60 millones de parámetros y 650,000 neuronas, consta de cinco capas convolucionales y tres capas completamente conectadas. Las dos primeras capas convolucionales van seguidas de la normalización y una capa de agrupación máxima, la tercera y la cuarta capa están directamente conectadas y la quinta capa convolucional va seguida de una capa de agrupación máxima. La salida de la arquitectura entra en una serie de dos capas completamente conectadas, y la segunda de ellas alimenta a una capa de softmax. Para evitar el sobreajuste de las capas completamente conectadas, los autores emplearon un método de regularización llamado “dropout” con una proporción de 0.5 [37]. Otra característica del modelo es el uso de la Unidad Lineal Rectificada (ReLU, por sus siglas en inglés) que se aplica a cada una de las primeras siete capas. Finalmente, los autores mencionaron que usar ReLU Nonlinearity podría entrenarse mucho más rápido que al usar funciones de activación de saturación que la Tangente y Sigmoide.

2.4.2 GoogleNet

La arquitectura de GoogleNet fue presentada en el trabajo de Szegedy et al. [15] y fue ganadora del ILSVRC en el 2014. GoogleNet posee siete millones de parámetros y contiene nueve módulos iniciales, cuatro capas convolucionales, cuatro capas de agrupación máxima, tres capas de agrupamiento promedio, cinco capas completamente conectadas y tres capas de softmax para los principales clasificadores auxiliares de la red. Además, utiliza la regularización de dropout en la capa totalmente conectada y aplica la activación ReLU en todas las capas convolucionales. Sin embargo, esta red es mucho más profunda y amplia con 22 capas totales, pero contiene un menor número de parámetros en comparación con AlexNet.

2.4.3 Inception V3

Es una arquitectura convolucional profunda ampliamente utilizada para tareas de clasificación [16]. El concepto del modelo fue introducido por Szegedy en la arquitectura de GoogleNet, donde propone Inception V3 al actualizar el módulo de inicio [38]. Inception V3 tiene múltiples bloques de construcción simétricos y asimétricos, donde cada bloque tiene varias ramas de convoluciones, agrupamiento promedio, agrupamiento máximo, concatenadas, de dropout y completamente conectadas. Esta red contiene 42 capas y posee 29.3 millones de parámetros, lo que significa que el costo computacional es aproximadamente 2.5 más alto que el de GoogleNet. Finalmente, los autores concluyeron que la combinación del menor número de parámetros y la regularización adicional con el suavizado de etiquetas de los clasificadores auxiliares normalizados por lotes permite entrenar una red de alta calidad en conjuntos de entrenamiento de tamaño relativamente modesto [16].

2.4.4 Residual Networks (ResNet)

Los modelos de ResNet fueron desarrollados por He et al. [17], que surgieron como una familia de arquitecturas profundas que mostraron buenos comportamientos de convergencia y una precisión bastante convincente. Ganaron el primer lugar en el desafío de clasificación de ILSVRC y *Common Objects in Context (COCO)* en 2015. ResNet se construyó con varias

unidades residuales aplicadas y se desarrolló con diferentes números de capas: 18, 34, 50, 101, 152 y 1202. Sin embargo, el número de operaciones puede variar dependiendo de las diferentes arquitecturas [17]. Por todo lo anterior, las unidades residuales se componen de capas convolucionales y de agrupación. ResNet es similar a VGG net [39], pero ResNet es aproximadamente ocho veces más profunda que VGG [38]. La arquitectura de ResNet 18 representa una buena compensación entre la profundidad y rendimiento, esta red está compuesta por cinco capas convolucionales, una de agrupación promedio y una capa totalmente conectada con un softmax. ResNet 50 contiene 49 capas convolucionales y una capa completamente conectada al final de la red. Finalmente, para ahorrar recursos de computación y tiempo de entrenamiento se seleccionó ResNet 18 y ResNet 50 para el desarrollo de este trabajo.

En seguida, la siguiente Tabla 2.1 enumera las CNNs empleadas en la investigación y algunas de sus propiedades. La profundidad de la red se define como el mayor número de capas convolucionales secuenciales o completamente conectadas en una ruta desde la capa de entrada a la capa de salida, finalmente, las entradas de las CNNs son imágenes RGB.

Tabla 2.1 Propiedades de las CNN pre-entrenadas.

Arquitectura	Profundidad	Parámetros (Millones)	Tamaño de la imagen de entrada
AlexNet	8	60	227 x 227
GoogleNet	22	7	224 x 224
Inception V3	48	23.9	299 x 299
ResNet 18	18	11.7	224 x 224
ResNet 50	50	25.6	224 x 224

2.5 Enfermedades de la planta de tomate

En la industria agrícola, las enfermedades de plantas han sido durante mucho tiempo uno de los principales problemas, no sólo porque afecta directamente a la producción, lo que se traduce en pérdidas económicas importantes, sino porque además es una de las principales amenazas para la seguridad alimentaria.

En la actualidad, a pesar de los avances tecnológicos y los conocimientos empíricos del cultivo de tomate, los niveles de producción que se pierden cada año son bastantes altos, lo que representa una afectación importante para las personas que se dedican a esta actividad, así como para el consumidor, pues se estima que cerca de 795 millones de personas alrededor del mundo no tienen acceso a alimentos de calidad [40].

Actualmente, las enfermedades se tratan mediante el uso de pesticidas, que significan un costo adicional en la producción y que no sólo tienen un impacto perjudicial para el medio ambiente, sino que también pueden afectar al producto final el cual es obtenido por el consumidor. Es por esta razón, que la clasificación y detección de enfermedades debe ser uno de los temas centrales cuando se habla de la producción del tomate, pues el realizar esta tarea tendría como beneficio el poder actuar a tiempo evitando que si una planta se encuentra enferma, esta se propague en todo el cultivo, empleando tareas de tratamiento tempranas y haciendo un uso rentable, responsable y eficaz de los pesticidas.

Las enfermedades de plantas se producen debido a diversos factores ambientales, entre los que se encuentran los cambios de temperatura, en la precipitación, la sombra, humedad, altos niveles de CO_2 , o bien al uso excesivo o incorrecto de fertilizantes que afectan a las plantas y cultivos de manera diversa, según la enfermedad que se trate [40].

El tomate puede ser afectado por alrededor de 200 plagas y enfermedades, que son clasificadas de la siguiente manera: bacterianas, virales, por hongos y oomicetos e insectos.

En la Figura 2.1 se muestran las principales enfermedades en tomate que se van a tratar en el presente estudio.

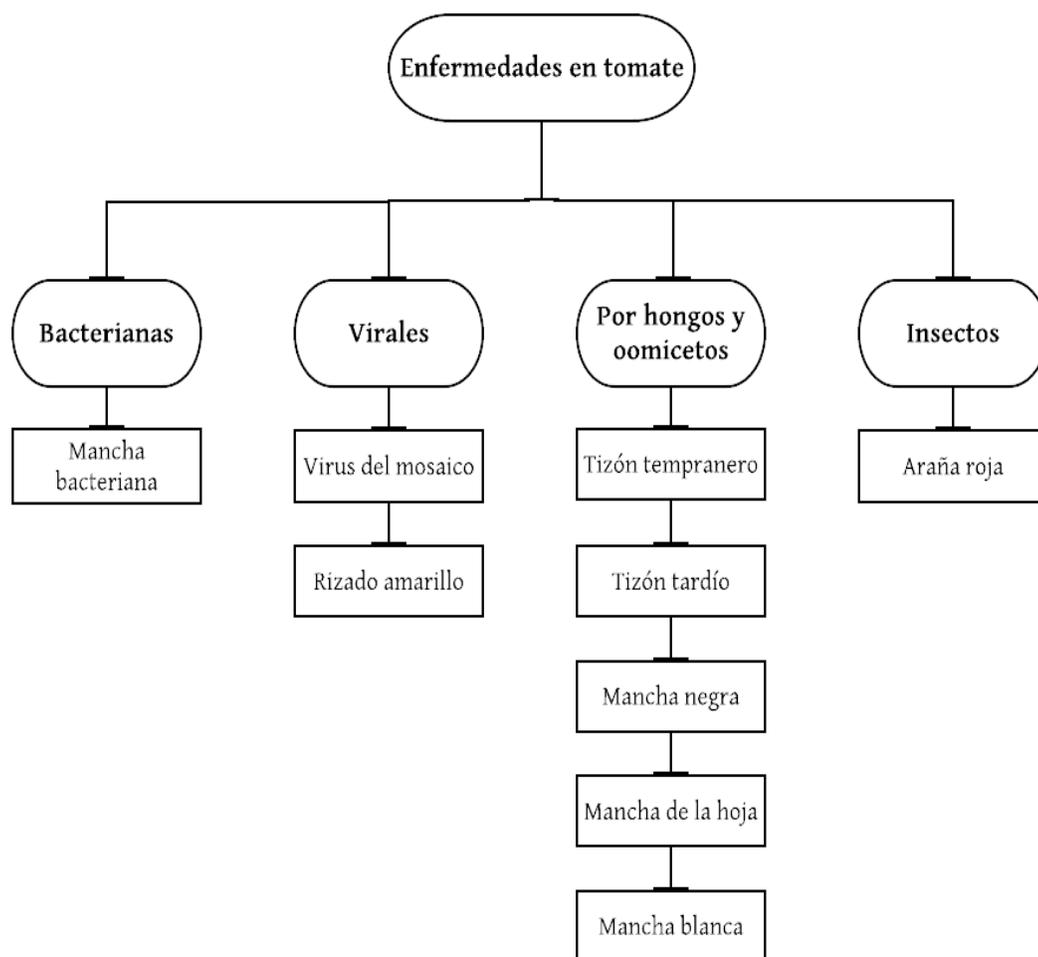


Figura 2.1 Diferentes categorías de enfermedades en tomate.

2.5.1 Bacterianas

Incluyen organismos unicelulares, relativamente pequeños, que pueden ser detectados usando microscopía de luz con gran aumento. Estas no tienen núcleo diferenciados, por lo que son clasificadas como procariotas, sin embargo, sí cuentan con ácido nucleico, aunque no dentro de una membrana celular. Su acelerada reproducción ocurre entre 20 minutos a algunas horas, lo que permite que las poblaciones bacterianas crezcan exponencialmente cuando las condiciones son favorables. Las bacterias pueden tener distintas formas, ya sea esféricas, elipsoidales o filamentosas, no obstante, la gran mayoría que ataca al cultivo de tomate tienen forma redondeada. Cabe destacar que existen 217 géneros de bacterias que han sido relacionadas con el tomate.

2.5.1.1 Mancha bacteriana



Figura 2.2 Muestra ilustrativa de la enfermedad mancha bacteriana.

- **Enfermedad:** Mancha bacteriana (*Xanthomonas campestris* pv. *Vesicatoria*).
- **Causa:** Bacteria.
- **Síntomas:** Las lesiones de la mancha empiezan como manchas empapadas de agua. Comúnmente, se vuelven más numerosas y se unen para formar áreas necróticas en las hojas, dándoles una apariencia deteriorada hasta que caen al suelo [41].

2.5.2 Virales

Los virus son pequeñas entidades infecciosas, invisibles a la microscopía óptica, y con una estructura bastante simple que se conforma de una cubierta de proteína, la cápside, dentro de la que se encuentra el ácido nucleico que generalmente es ácido ribonucleico en los virus vegetales. Una gran cantidad de virus pueden ser inoculados de manera artificial en el tomate, mientras que más de cien lo hacen de forma natural. Los virus que afectan a la planta de tomate tienen forma de estructuras simétricas cuyo tamaño se mide en nanómetros. Por otro lado, causan varios síntomas que pueden ser confusos, por lo que su identificación resulta difícil. La planta puede ser afectada por varios virus a la vez, y la mayoría de ellos son transmitidos por picadura de insectos.

2.5.2.1 Virus del mosaico



Figura 2.3 Muestra ilustrativa de la enfermedad virus del mosaico.

- **Enfermedad:** Virus del mosaico (*ToMV*).
- **Causa:** Virus.
- **Síntomas:** Consiste en alteraciones de la forma y color de los folíolos, alternándose áreas cloróticas con otras de color verde normal y oscuro (mosaicos). Los folíolos se deforman apareciendo rizados o abarquillados. Reduce el tamaño de la planta y el número de frutos. Cuanto mayor es la infección mayor es la repercusión [41].

2.5.2.2 Rizado amarillo del tomate



Figura 2.4 Muestra ilustrativa de la enfermedad rizado amarillo del tomate.

- **Enfermedad:** Rizado amarillo del tomate (*TYLCV*).
- **Causa:** Virus.
- **Síntomas:** Las hojas se reducen de tamaño y se curvan hacia arriba, aparecen arrugadas y muestran amarillamiento de las venas y los márgenes de la hoja. Las plantas enteras aparecen atrofiadas y espesas. Las flores pueden no desarrollarse y dejarse caer [41].

2.5.3 Hongos y oomicetos

Los hongos fitopatógenos suelen ser organismos pequeños y más complejos que las bacterias, y aunque en ocasiones pueden ser reconocidos a simple vista, la mayoría de las veces es necesaria la ayuda de un microscopio para detectarlos. En las plantas de tomate los hongos producen estructuras especializadas que son muy resistentes a elementos adversos y perfectamente se adaptan para su conservación. Durante su formación, se extienden a las diferentes partes de la planta de tomate y rápidamente las esporas germinan y producen una estructura de unión denominada “apresorio” que sirve para ayudar en la penetración del huésped. Algunos hongos prefieren tejidos jóvenes, mientras que otros se inclinan por algunos más viejos.

Por último, los hongos muestran características específicas según sea la parte de la planta que estén afectando.

2.5.3.1 Tizón temprano



Figura 2.5 Muestra ilustrativa de la enfermedad tizón temprano.

- **Enfermedad:** Tizón temprano (*Alternaria solani*).
- **Causa:** Hongo.
- **Síntomas:** Los primeros síntomas comienzan como lesiones de forma oval con una región clorótica a través de la lesión. El tejido de las hojas entre las venas es destruido. A medida que avanza la enfermedad, las hojas quedan gravemente afectadas y los tallos del tomate pueden infectarse con el hongo que conduce al cáncer de tallo [41].

2.5.3.2 Tizón tardío



Figura 2.6 Muestra ilustrativa de la enfermedad tizón tardío.

- **Enfermedad:** Tizón tardío (*Phytophthora infestans*).
- **Causa:** Oomiceto.
- **Síntomas:** Afecta todas las partes de la planta. A medida que la enfermedad progresa el follaje se va marchitando y toda la planta puede morir. Las lesiones de la planta comienzan como regiones irregulares formadas por la humedad y el fruto puede infectarse [41].

2.5.3.3 Mancha negra



Figura 2.7 Muestra ilustrativa de la enfermedad mancha negra.

- **Enfermedad:** Mancha negra (*Passalora fulva*).
- **Causa:** Hongo.
- **Síntomas:** Atacan esencialmente los folículos, muestran en primer lugar manchas de color verde pálido. Estas alcanzan varios milímetros de diámetro y presentan un contorno difuminado. El hongo puede formar esporas en la cara superior del limbo que finalmente las hojas se amarillean totalmente y se secan [41].

2.5.3.4 Mancha de la hoja



Figura 2.8 Muestra ilustrativa de la enfermedad mancha de la hoja.

- **Enfermedad:** Mancha de la hoja (*Septoria lycopersici*).
- **Causa:** Hongo.
- **Síntomas:** Puede ocurrir en cualquier etapa del desarrollo del tomate y comenzar como pequeñas manchas empapadas de agua o manchas circulares blancas grisáceas en la parte inferior de las hojas viejas. Su centro se necrosa bastante rápido y toma un tinte bronceado a blanquecino [41].

2.5.4 Insectos

Los ácaros, como los insectos, son artrópodos y pertenecen a la clase arácnida. Muchas de las familias de ácaros incluyen plagas de plantas, y las más conocidas son las *tetranychidae*, especies a las que comúnmente se les conoce como ácaros o arañas amarillas, rojas o verdes. Algunas de estas especies se alimentan del tomate, estas pueden ser responsables de daños significativos, especialmente en cultivos protegidos de plantas ornamentales y vegetales.

2.5.4.1 Araña roja



Figura 2.9 Muestra ilustrativa de la enfermedad araña roja.

- **Enfermedad:** Araña roja (*Tetranychus urticae*).
- **Causa:** Arácnido.
- **Síntomas:** Los ácaros pueden ser visibles como diminutos puntos en la parte inferior de las hojas, estas se vuelven amarillas y pueden caer de la planta. Son capaces de causar daños serios en poco tiempo [41].

La araña roja o araña de dos puntos (*Tetranychus urticae*) es una plaga que afecta a numerosos cultivos en todo el mundo. A pesar de su tamaño pequeño, son capaces de causar daños serios en poco tiempo debido a su gran capacidad reproductiva. Existen más de 1200 especies y varias de ellas son consideradas plagas de suma importancia: la araña roja (*Tetranychus urticae*) es, por mucho, la especie más importante en invernaderos y en muchos cultivos a campo abierto.

Capítulo 3

Materiales y métodos

En este capítulo se describe a detalle la metodología seguida para la clasificación de enfermedades en tomate, representada en la Figura 3.1. Los componentes individuales de la metodología se describen de forma puntual en la sección correspondiente a cada uno de ellos.

De manera general, el capítulo engloba la descripción del conjunto de datos obtenido, el proceso de entrenamiento de las CNN, y las medidas de desempeño, que juzgan el comportamiento de las redes. En este punto, es importante retomar los objetivos de este trabajo, puesto que la metodología que se encuentra plasmada, es la clave para lograrlos.

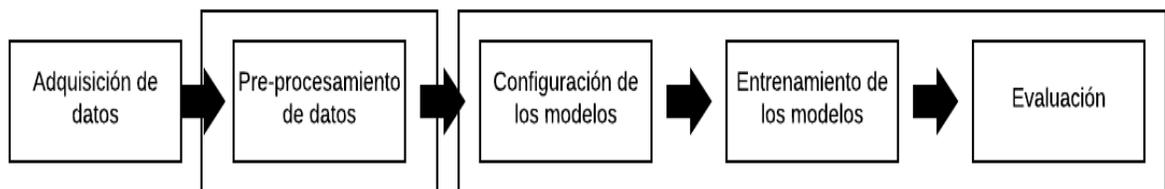


Figura 3.1 Diagrama de flujo de la metodología seguida.

3.1 Adquisición de datos

La adquisición de datos es una etapa importante para desarrollar cualquier aplicación. En particular, los modelos profundos requieren de un gran conjunto de datos para evitar el sobreajuste, lo cual presenta un gran desafío. Actualmente, solo pocos conjuntos de datos están disponibles públicamente para plantas enfermas, tal es el caso de *PlantVillage* [42], mientras que otros fueron producidos por los autores para sus necesidades de investigación.

3.1.1 *PlantVillage* - Dataset

Es un repositorio de acceso abierto, publicado en línea en el sitio web “www.plantvillage.org”. Las imágenes de la base de datos se tomaron en estaciones de investigación experimentales tanto públicas como privadas de Estados Unidos de América por un grupo de técnicos. De los ensayos de campo de cultivo infectados con una enfermedad los técnicos retiraron la hoja de la planta y estas se colocaron contra una hoja de papel que proporcionaba un fondo gris o negro. Todas las imágenes fueron tomadas bajo las mismas condiciones de luz, para cada hoja se capturaron de cuatro a siete imágenes con una cámara digital estándar (Sony DSC - Rx100/ 13 20.2 megapíxeles). El registro de los datos contiene 54,323 imágenes de 14 cultivos (manzana, arándano, cereza, maíz, uva, naranja, durazno, pimiento, papa, frambuesa, soja, calabaza, fresa, tomate) y abarca 38 tipos de enfermedades de plantas. Sin embargo, para fines de la investigación solamente fueron extraídas las imágenes de la hoja de tomate.

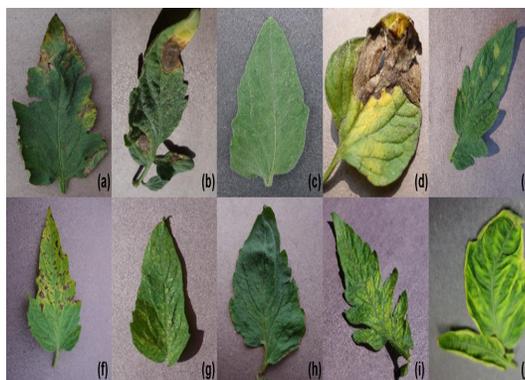


Figura 3.2 Imágenes de muestra obtenidas de *PlantVillage* - Dataset.

En la Figura 3.2 se muestra un ejemplo de cada una de las clases del nuevo conjunto de datos ((Fila superior de izquierda a derecha) etiquetadas como: **(a)** Mancha bacteriana **(b)** Tizón temprano **(c)** Hoja sana **(d)** Tizón tardío **(e)** Mancha negra (Fila inferior de izquierda a derecha) labelled as: **(f)** Mancha de la hoja **(g)** Araña roja **(h)** Mancha blanca **(i)** Virus del mosaico **(j)** Rizado amarillo del tomate).

Por otro lado, la Tabla 3.1 hace mención al número de imágenes proporcionadas para cada una de las clases. Es importante señalar que las imágenes fueron etiquetadas por un experto, por lo que se tiene certeza de que las imágenes de entrenamiento y prueba (validación) están correctamente clasificadas. En esta etapa se realizó el pre-procesamiento de las imágenes, puesto que fueron recortadas a $224 * 224$ ó $299 * 299$ píxeles, de acuerdo con el tamaño de entrada de cada arquitectura. Finalmente, la base de datos fue dividida en 80% para entrenamiento y 20% para prueba. La elección de la división partió de los resultados de la investigación de Mohanty et al. [22], debido a que esa distribución obtuvo el mejor rendimiento.

Tabla 3.1 Detalles del conjunto de datos.

Etiqueta	Clases	Número de imágenes
(1)	Mancha bacteriana	2,127
(2)	Tizón temprano	1,000
(10)	Hoja sana	1,591
(3)	Tizón tardío	1,909
(4)	Mancha negra	952
(5)	Mancha de la hoja	1,771
(6)	Araña roja	1,676
(7)	Mancha blanca	1,404
(9)	Virus del mosaico	373
(8)	Rizado amarillo del tomate	5,357
	Total	18,160

3.2 Configuración de los modelos

Con la finalidad de realizar una comparación equitativa entre las arquitecturas se estandarizaron los hiperparámetros descritos en la Tabla 3.2.

Tabla 3.2 Valor de hiperparámetros en los experimentos.

Hiperparámetros	Valor
Algoritmo de optimización	SGDM
Momento	0.9000
Tasa de aprendizaje	1.0000e−03
Regularización L_2	1.0000e−04
Épocas	30
Tamaño de lote	32

Cada uno de los modelos de CNN fueron entrenados usando el *Descenso de Gradiente Estocástico* (SGD, por sus siglas en inglés), este se ha convertido en el algoritmo de optimización dominante y ha demostrado compensar de manera adecuada la precisión y la eficiencia, además, se ejecuta con rapidez y converge fácilmente [43]. El SGD es simple y efectivo, por lo tanto requiere un ajuste de los hiperparámetros de los modelos, particularmente de la *tasa de aprendizaje*, que se usa en la optimización, dado que determina la rapidez con la que se ajustan los pesos de las CNNs para obtener un mínimo local o global de la función de pérdida. El *momento* apoya a acelerar la convergencia considerablemente del SGD en la dirección adecuada amortiguando las oscilaciones [44]. La *tasa de aprendizaje* fue establecida a 1.0000e−03, ese valor corresponde a la velocidad a la que una función se mueve a través del espacio de búsqueda. Una pequeña tasa conduce a resultados precisos. La combinación de la *regularización L_2* y SGD da como resultado la caída de peso en las redes, donde en cada actualización los pesos se escalan un poco más pequeños que uno. Los modelos ejecutaron un total de 30 épocas, donde cada época es el número de iteraciones de entrenamiento, la elección se hizo de acuerdo a los resultados de la propuesta de Mohanty et al. [22], porque convergen consistentemente después del primer paso hacia abajo en la tasa de aprendizaje. Finalmente, el

tamaño de lote es un subconjunto del conjunto de entrenamiento que se utiliza para evaluar el gradiente de la función de pérdida y actualiza el peso.

3.3 Entrenamiento de los modelos

En esta sección, las cinco arquitecturas de CNN (AlexNet, GoogleNet, Inception V3, ResNet 18 y ResNet 50) son entrenadas con el conjunto de datos descrito en la sección anterior. Con el objetivo de construir un clasificador para enfermedades en tomate, se utilizó el enfoque de DL, especialmente CNN. Este tipo de redes se entrenan directamente utilizando imágenes “crudas”, que aprenden a extraer características de los datos sin la intervención de humanos, lo que significa que crean construcciones de características de manera automática.

3.3.1 Transferencia de Aprendizaje

En el enfoque de Transferencia de Aprendizaje (TL, por sus siglas en inglés), primeramente las CNNs se entrenan utilizando un conjunto de datos diferente, a esto se le llama “pre-entrenamiento”. En este caso las CNNs se entrenaron mediante un conjunto de datos de gran tamaño como ImageNet [45], que contiene 1.2 millones de imágenes con 1,000 clases. Debido a que la base de datos de tomate es insuficiente para entrenar CNNs, es recomendable aplicar este enfoque (TL), por lo tanto se emplean los pesos de las redes adquiridos en el pre-entrenamiento.

De acuerdo con Yang y Pan [46], el TL se define como: dado un dominio fuente D_S y una tarea de aprendizaje T_S , un dominio de destino D_T y una tarea de aprendizaje T_T , el TL tiene como objetivo ayudar a mejorar el aprendizaje de la función predictiva f_T en D_T utilizando el conocimiento en D_S y T_S , donde $D_S \neq D_T$ o $T_S \neq T_T$. En otras palabras, TL es una técnica de ML, que se define como la transferencia de conocimiento, que se aprendió anteriormente en un dominio y se reutiliza para capacitar en otra tarea o dominio relacionado. Tiene como objetivo mejorar el aprendizaje aprovechando el conocimiento del D_S .

En la investigación de Torrey y Shavlik [47], identifican tres razones por las cuales el TL mejora el aprendizaje:

1. El rendimiento inicial que puede lograr en la T_T utilizando solo el conocimiento transferido, antes de que se realice un aprendizaje adicional, lo que se traduce a un rendimiento más alto al comienzo del aprendizaje.
2. La cantidad de tiempo que lleva aprender completamente la T_T en base al conocimiento transferido en comparación al tiempo que conlleva a una red aprender desde cero. Significa que existe una pendiente más pronunciada en la curva de aprendizaje.
3. El rendimiento final es alcanzable en la T_T , lo cual hace referencia a un rendimiento asintótico alto.

Después de que las CNNs fueron entrenadas mediante ImageNet, se deben de ajustar a través de la capacitación adicional con los datos específicos para el problema, en este caso se entrenó con el conjunto de datos de tomate. A este proceso se le llama *ajuste fino*, el cual consiste en reemplazar las últimas tres capas de las CNNs: capa completamente conectada, softmax y la capa de salida, para abordar el problema, donde el número de clases es 10, puesto que las primeras capas se encargan de la extracción de características, bordes y formas. La Figura 3.3 representa el TL, donde muestra el reemplazo de las capas.

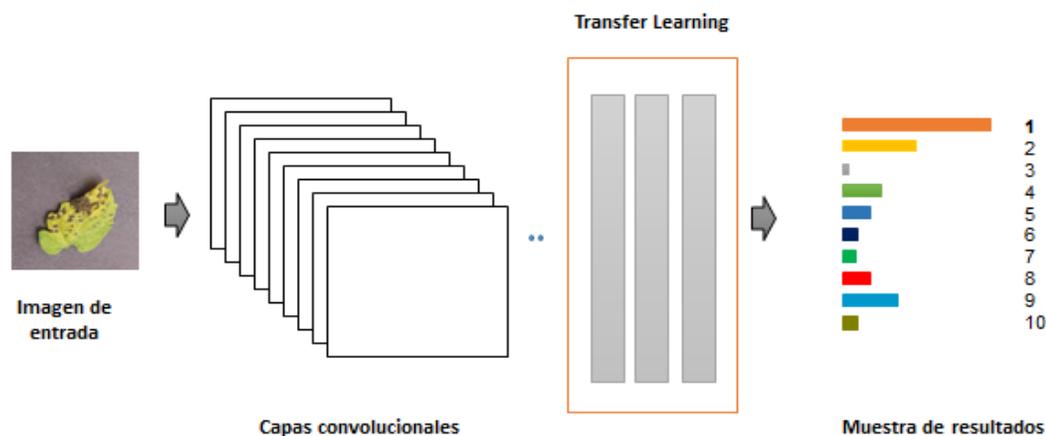


Figura 3.3 Representación de transfer learning.

3.3.2 Clasificación

La última etapa para la identificación de la enfermedad es la clasificación; es el modelado basado en delinear clases de salida en función de un conjunto de entrada, e igualmente es la tarea de aprendizaje supervisado más utilizada cuando se emplean CNNs. Dado un tensor de imagen de dimensiones [Altura, Ancho, Canles], el modelo aprende a predecir su etiqueta, una variable categórica. La función *softmax*, también conocida como función exponencial normalizada, estima la probabilidad de una observación que pertenece a una determinada clase.

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}} \quad (3.1)$$

En otras palabras, la Ecuación (3.1) aplica la función a cada elemento z_i del vector de entrada z y lo normaliza dividiéndolo por la suma todos sus exponecniales. Esta normalización asegura que la suma de los componentes del vector de salida $\sigma(z)$ sea igual a uno. Esta restricción permite que el entrenamiento converja más rápido.

En la Figura 3.4 se muestra el flujo de trabajo de una CNN asignada a la tarea de clasificación, sin embargo, se centra en la implementación de la capa de *softmax*, que debe tener el mismo número de neuronas que la capa de salida y que, además, se encuentra ubicada antes de ella. La capa de salida o capa de clasificación, que determina la enfermedad presente en la hoja de la planta.

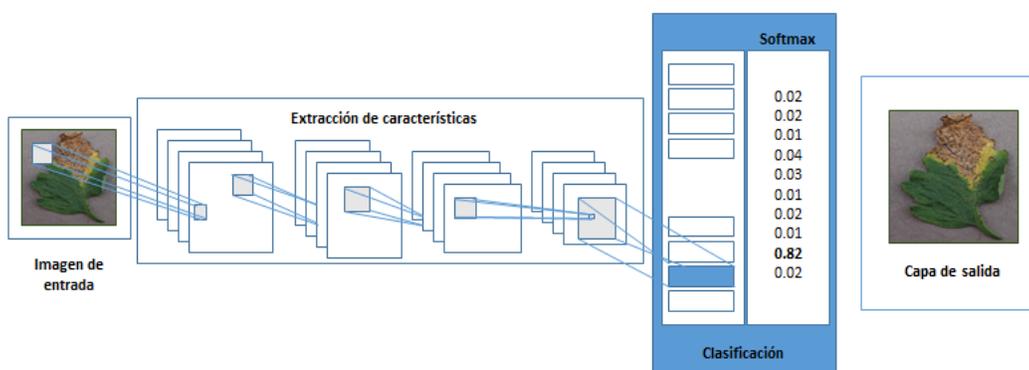


Figura 3.4 Flujo de trabajo de una CNN.

3.4 Medidas de desempeño

Una métrica es una función que es utilizada para juzgar el desempeño de un algoritmo. Por lo tanto, la selección de una métrica para la evaluación del mismo, es clave para discriminar y obtener un clasificador óptimo.

A fin de estimar y comparar el rendimiento de los modelos, se utilizaron las siguientes métricas:

3.4.1 Matriz de confusión

La precisión de una clasificación se puede evaluar calculando el número de ejemplos de la clase correctamente reconocidos, el número de ejemplos correctamente reconocidos que no pertenecen a la clase, ejemplos que fueron asignados incorrectamente a la clase o que no fueron reconocidos como ejemplos de clase. Estos cuatro recuentos constituyen una matriz de confusión, la cual se muestra en la Figura 3.5.

		Valor Real	
		Positivos	Negativos
Valor Predicho	Positivos	Verdadero Positivo (TP)	Falso Positivo (FP)
	Negativos	Falso Negativo (FN)	Verdadero Negativo (TN)

Figura 3.5 Matriz de confusión.

La matriz de confusión es un mecanismo para visualizar el desempeño del clasificador, en la Figura 3.5, las filas representan la predicción del clasificador, mientras que las columnas son el valor real de cada clase. Las celdas que se encuentran fuera de la diagonal corresponden a las observaciones clasificadas incorrectamente.

Además, contiene cuatro parámetros que son:

- **Verdaderos Positivos** (TP, por sus siglas en inglés): Indica la porción de buena predicción para los objetos positivos.
- **Verdaderos Negativos** (TN, por sus siglas en inglés): Indica la porción de buena predicción para los objetos negativos.

- **Falsos Positivos** (FP, por sus siglas en inglés): Indica la porción de objetos negativos que fueron clasificados como positivos.
- **Falsos Negativos** (FN, por sus siglas en inglés): Indica la porción de objetos positivos que se clasifican como negativos.

3.4.2 Matriz de confusión multiclase

La Figura 3.6 muestra la matriz de confusión para múltiples clases. Como se muestra en la Figura 3.6, TP_A es el número de muestras positivas verdaderas en la clase A, es decir, el número de muestras que se clasifican correctamente de la clase A, y E_{BA} son las muestras de la clase A que se clasificaron incorrectamente como clase B. Por lo tanto, el FN en la clase A (FN_A) es la suma de E_{AB} y E_{AC} que indica la suma de todas las muestras de la clase A que se clasificaron incorrectamente como clase B o C.

Simplemente, el FN de cualquier clase que se encuentre en una columna se puede calcular agregando los errores en esa clase/columna. Finalmente, con la matriz de confusión de $m \times m$ hay m clasificaciones correctas y $m^2 - m$ de posibles errores [48].

		Valor Real		
		A	B	C
Valor Predicho	A	TP_A	E_{BA}	E_{CA}
	B	E_{AB}	TP_B	E_{CB}
	C	E_{AC}	E_{BC}	TP_C

Figura 3.6 Matriz de confusión multiclase.

En las estadísticas tradicionales, un FP también se conoce como “error tipo I” y un FN se conoce como “error tipo II”. Al realizar un seguimiento de estos parámetros, se puede lograr un análisis más detallado acerca del desempeño del modelo calculando diferentes métricas basadas en sus combinaciones.

3.4.3 Exactitud

Es la métrica básica de desempeño más utilizada. Calcula el porcentaje de muestras que se clasifican correctamente y se representa en la Ecuación (3.2)

$$Exactitud = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.2)$$

3.4.4 Precisión

Definida como el número de TP dividido por el número de TP más FP , viene dada por la Ecuación (3.3). Esta medida, también conocida como “valor predictivo”, evalúa el poder predictivo del modelo, en otras palabras, es cuán preciso es el modelo de aquellos positivos predichos y cuántos de ellos son realmente positivos.

$$Precisión = \frac{TP}{TP + FP} \quad (3.3)$$

3.4.5 Sensibilidad

También llamada “tasa verdadera positiva” o “recall”. Corresponde a la probabilidad de clasificar correctamente la enfermedad presente en las hojas. Para estimar la sensibilidad se utiliza la Ecuación (3.4). Donde TP son las hojas enfermas que se clasificaron debidamente como “enfermas” y FN u hojas enfermas que se clasificaron como “sanas”.

$$Sensibilidad = \frac{TP}{TP + FN} \quad (3.4)$$

3.4.6 Especificidad

Para estimar la especificidad (“tasa negativa verdadera”) se emplea la Ecuación (3.5), la cual representa la cantidad de hojas sanas que obtuvieron un resultado negativo, donde TN son las hojas sanas que se diagnosticaron como “sanas” y FP, las cuales fueron hojas sanas que se diagnosticaron “enfermas”.

$$Especificidad = \frac{TN}{TN + FP} \quad (3.5)$$

3.4.7 F-score

Se define como la media armónica ponderada de la precisión y recall. Se calcula con la Ecuación (3.6). Esta medida se centra en el análisis de la clase positiva. Un valor alto de esta métrica indica que el modelo tiene mejor desempeño en dicha clase. (El valor de F-score varia entre 0 (peor valor posible) y 1 (mejor valor posible)).

$$F - score = 2 * \frac{Precisión * Recall}{Precisión + Recall} \quad (3.6)$$

3.4.8 Área bajo la curva ROC

La curva ROC es una representación gráfica y bidimensional del rendimiento del clasificador. En el eje vertical se gráfica la sensibilidad (verdaderos positivos) contra la especificidad (verdaderos negativos). A fin de comparar clasificadores, es posible que se desee reducir el rendimiento de la ROC a un solo valor escalar que represente el rendimiento esperado. Un método común es calcular el AUC [49, 50], puesto que el AUC es una parte del área del cuadrado de la unidad y su valor siempre estará entre 0.0 y 1.0, ningún clasificador realista debe tener un AUC menor a 0.5. El AUC tiene una propiedad estadística significativa, el cual menciona que el AUC de un clasificador es equivalente a la propiedad de que un clasificador clasifique una instancia positiva elegida aleatoriamente más alta que una instancia negativa elegida aleatoriamente.

No obstante, para el presente trabajo se empleó la *ROC Multiclase*, dado que con más de dos clases la situación se vuelve compleja si se desea gestionar todo el espacio. Con n clases, la

matriz de confusión se convierte en una matriz $n \times n$ que contiene las n clasificaciones correctas (entradas diagonales principales) y $n^2 - n$ posibles errores (entradas fuera de la diagonal). Un método para gestionar n clases es producir n diferentes gráficos ROC, uno para cada clase, esto se llama “formulación de referencia de clase”. Específicamente si C que es el conjunto de todas las clases, el gráfico ROC i traza el rendimiento de la clasificación de la clase C_i como la clase positiva y todas las demás como la negativa [51].

Por otro lado, el *AUC Multiclase* es adoptado por Hand y Till [52], los cuales otorgan una generalización multiclase de las AUC. En resumen, los autores deseaban una medida que no sea sensible a la distribución de las clases y a los costos de error. Derivaron una función probabilística que mide la discriminabilidad de clases no ponderada por pares. La medida que ellos mencionan se llama M y es representada por la Ecuación (3.7):

$$M = \frac{2}{c(c-1)} \sum_{i < j} A(i, j), \quad (3.7)$$

donde las clases múltiples han sido etiquetadas como $0, 1, 2, \dots, c-1$ ($c > 2$), lo cual su regla de clasificación otorgará estimaciones de probabilidad de que cada punto de prueba pertenezca a cada clase. Por lo tanto, se define $A(i|j)$ como la probabilidad de que un miembro de la clase j dibujado al azar tenga una probabilidad estimada menor de pertenecer a la clase i que un miembro de la clase i dibujado al azar. Una de las reglas de clasificación es que a menudo producen probabilidades estimadas que están delimitadas por 0 y 1, además, una de sus propiedades atractivas de M (y, por supuesto, AUC) es que es invariable a las transformaciones monotónicas de las probabilidades estimadas.

3.5 Implementación

Los experimentos se realizaron con MATLAB 2018b aunado a Deep Learning Toolbox, que proporciona un marco para diseñar e implementar CNNs, donde las aplicaciones y los gráficos ayudan a visualizar las activaciones de la red y monitorear el progreso del entrenamiento de las redes. Mientras que, el análisis estadístico de cada arquitectura se realizó con el software libre *R* (versión 3.5.2) [53] con sus paquetes *pROC* (versión 3.5.3) [54] y *caret* (versión 6.0-84) [55].

3.5.1 Características del área de trabajo

La eficiencia es importante cuando se trabaja con CNNs. Actualmente, se ha implementado como herramienta el uso de GPU, las cuales son componentes de hardware especializado que se desarrollaron originalmente para aplicaciones gráficas. Sin embargo, han sido explotadas para entrenar CNNs de manera eficiente, debido a que un entrenamiento sin GPU sería extremadamente costoso. Es importante mencionar que el entrenamiento de los modelos requieren demasiado tiempo de entrenamiento (alrededor de dos a cuatro horas en una computadora de alto rendimiento), mientras, la clasificación es rápida en una GPU.

Por ello, se usó una PC (cuyas características se encuentran en la Tabla 3.3) para el proceso de entrenamiento y prueba de los modelos para la clasificación de enfermedades en tomate.

Tabla 3.3 Características de la computadora.

Hardware y Software	Características
Memoria	16 Gb
Procesador	Intel Core i7-7700 CPU @ 3.60 GHz
GPU	NVIDIA GeForce GTX 1070 X 8 Gb
Sistema Operativo	Windows 10, 64 bits

Capítulo 4

Comparativa de arquitecturas de CNN para la clasificación de enfermedades en tomate

4.1 Resultados y discusión

En este capítulo se exponen los resultados obtenidos para cada etapa de la metodología, la cual fue descrita en el **Capítulo 3**. El objetivo principal de la investigación es comparar y evaluar el desempeño de las CNNs para la clasificación de enfermedades en tomate mediante diferentes métricas, tales como: *exactitud*, *precisión*, *sensibilidad*, *especificidad*, *F-score* y *AUC*. Es importante señalar, que la experimentación se realizó de manera equitativa para cada una de las CNNs, esto incluye la misma configuración de hiperparámetros y su implementación.

Tabla 4.1 Medidas de rendimiento (%) para cada modelo pre-entrenado.

Medidas de desempeño	AlexNet	GoogLeNet	Inception V3	ResNet 18	ResNet 50
Exactitud	98.93	99.39	98.65	99.06	99.15
Precisión	98.73	99.28	98.28	98.75	98.94
Sensibilidad	98.38	99.12	97.84	98.67	98.72
Especificidad	99.88	99.93	99.84	99.89	99.90
F-Score	98.54	99.20	98.05	98.70	98.81
AUC	99.28	99.72	99.20	99.20	99.55
Tiempo (min)	94.17	133.16	649.85	147.73	476.78

Los resultados de la clasificación para las cinco arquitecturas de CNN se muestran en la Tabla 4.1. En donde, es posible observar que todos los modelos anteriormente entrenados mostraron un desempeño similar estadísticamente significativo. Comenzando con el *AUC*, Inception V3 y ResNet 18 obtuvieron los resultados más bajos con el 99.2%, seguidos por AlexNet con 99.28% y ResNet 50 con 99.55%, mientras que GoogleNet logró el mejor *AUC* con 99.72%, lo que representa una clasificación de enfermedades casi perfecta. Por otro lado, en la *exactitud*, Inception V3 obtuvo el rendimiento más bajo con 98.65%, seguido por AlexNet, ResNet 18 y ResNet 50 con 98.93%, 99.06% y 99.15% respectivamente, de nueva cuenta GoogleNet fue el resultado más alto con 99.39%. De la misma manera, el orden de los resultados con base a la métrica de *precisión* fue igual a la anterior, donde el porcentaje alto lo obtuvo GoogleNet (99.26%) y por el contrario Inception V3 con 98.28%. Finalmente, en las mediciones de *sensibilidad*, *especificidad* y *F-score*, el desempeño de Inception V3 fue deficiente con 97.84%, 99.84% y 98.05%, en comparación con GoogleNet que obtuvo el mejor porcentaje en las métricas anteriores con 99.12%, 99.93% y 99.20%. Además, en función del tiempo de procesamiento que le tomó a cada CNN llevar a cabo el proceso de entrenamiento, AlexNet mostró el mejor rendimiento al tomar el menor tiempo, por otro lado, nuevamente Inception V3 fue la CNN que tardó mayor cantidad de tiempo al momento del entrenamiento.

Por consiguiente, en la Figura 4.1 se graficó el comportamiento de cada una de las arquitecturas, en la cual se compara la exactitud del modelo que va teniendo conforme avanzan las iteraciones. Como se puede observar, el modelo de AlexNet resultó ser demasiado inestable para modelar el conjunto de datos a lo largo del entrenamiento. No obstante, ResNet 18, ResNet 50, GoogleNet e Inception V3 mostraron una pendiente más pronunciada en la curva de aprendizaje. Finalmente, con base a los resultados obtenidos en cada una de las medidas de desempeño, se determinó que **GoogleNet** fue el modelo que mejor se adaptó a la tarea de clasificación de enfermedades en tomate, mientras que Inception V3, siendo uno de los modelos más profundos obtuvo un desempeño pobre, sin embargo las demás arquitecturas como bien se mencionó anteriormente mostraron resultados significativos.

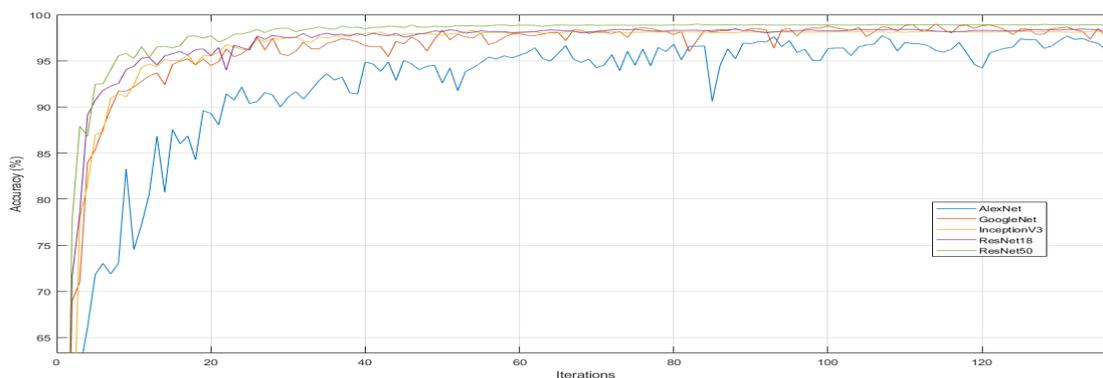


Figura 4.1 Exactitud vs iteraciones.

Dado que GoogleNet obtuvo mejores resultados, se continuo evaluando su comportamiento. Es posible evaluar visualmente el rendimiento del clasificador y determinar cuáles clases fueron resaltadas por las neuronas del modelo.

En la Figura 4.3 se presenta la matriz de confusión de GoogleNet, en donde las filas están relacionadas con la clase predicha y las columnas representan el valor actual/real. Las celdas que corresponden a la diagonal se asocian con las observaciones que fueron correctamente clasificadas, mientras que las que quedan fuera se relacionan con observaciones que fueron clasificadas incorrectamente. En segundo lugar, se procedió a obtener las medidas de desempeño de cada una de las clases, con el fin de conocer qué clase obtuvo el porcentaje más bajo de error.

		Valor real									
		1	2	3	4	5	6	7	8	9	10
Valor predicho	1	422	0	0	0	0	0	2	0	0	0
	2	1	197	3	1	0	0	0	0	0	0
	3	0	1	377	0	0	0	1	0	0	0
	4	0	0	0	187	0	0	0	0	0	0
	5	0	1	0	0	351	0	0	0	0	0
	6	0	1	1	2	0	334	1	0	1	0
	7	0	0	0	0	3	1	227	0	0	0
	8	2	0	0	0	0	0	0	1071	0	0
	9	0	0	0	0	0	0	0	0	73	0
	10	0	0	0	0	0	0	0	0	0	318

Figura 4.2 Matriz de confusión de GoogleNet.

En la Tabla 4.2 se muestra el desempeño de las clases con base a la *precisión*, *sensibilidad*, *especificidad* y *F-score*.

Tabla 4.2 Desempeño de cada clase.

Etiqueta	Precisión	Sensibilidad	Especificidad	F-score
(1)	99.52	99.29	99.93	99.41
(2)	97.52	98.50	98.85	98.00
(3)	99.47	98.95	99.93	99.21
(4)	100	98.42	100	99.20
(5)	99.71	99.15	99.96	99.43
(6)	98.23	99.70	99.81	98.96
(7)	98.57	98.57	99.88	98.57
(8)	99.81	100	99.92	99.90
(9)	100	98.64	100	99.31
(10)	100	100	100	100

Haciendo mención a la Tabla 4.2, dos de las diez clases han producido el 100% de los resultados de la clasificación, debido a que esas enfermedades tienen una apariencia y características distintivas en comparación con las demás. El modelo de GoogleNet identificó correctamente las clases que pertenecen a la enfermedad: mancha negra, virus del mosaico y la clase sana. Esto es con base en la métrica de precisión, por el contrario la enfermedad del tizón temprano fue la que obtuvo menor porcentaje de precisión.

A pesar de ser la tercera red con “poca profundidad” cuenta con un número considerablemente menor de parámetros (7 millones), sin embargo, la superioridad de GoogleNet en comparación de las arquitecturas anteriormente mencionadas se debe al módulo de inicio inspirado en la arquitectura *Network in Network*, propuesta en la investigación de Lin et al.[56]. El módulo de inicio utiliza convoluciones paralelas de: 1 x 1, 3 x 3 y 5 x 5 junto a una capa de agrupación máxima en paralelo, lo que le permite capturar una variedad de características, además, filtra información a lo largo de las capas lo que reduce el tamaño de la red.

Por último, la Figura 4.3 se presenta la curva ROC multiclase obtenida en función del rendimiento de GoogleNet. Estas curvas generalmente, presentan una tasa verdadera positiva en el eje Y y una tasa verdadera negativa en el eje X , lo cual significa que en la esquina superior izquierda del gráfico es el punto ideal. Es importante mencionar que este tipo de curvas se usan típicamente en la clasificación binaria, sin embargo, para extender su uso a una clasificación multiclase, es necesario binarizar la salida, por lo cual se genera un promedio de la curva. Como se puede observar el desempeño del modelo es excelente, puesto que logra un valor estadísticamente significativo de 99.72%, lo cual significa que el modelo de GoogleNet presentó una tasa de sensibilidad y especificidad que es capaz de clasificar las enfermedades en tomate con solo el 0.28% de error.

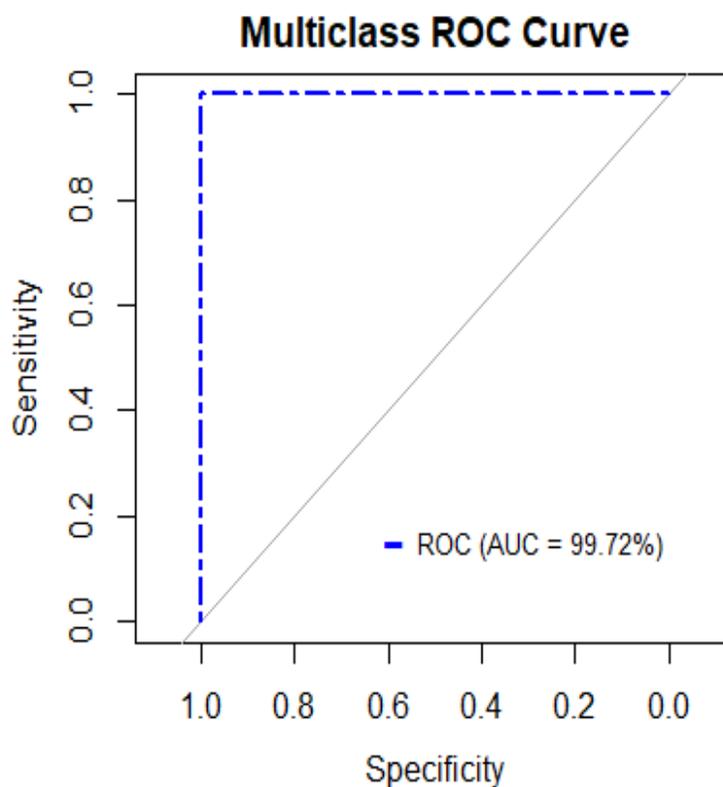


Figura 4.3 Curva ROC.

4.1.1 Selección de subconjuntos

Después de haber evaluado y comparado las cinco diferentes arquitecturas de CNN, se concluyó que GoogleNet fue el modelo que mejor resultado obtuvo en la clasificación de enfermedades en tomate, y retomando uno de los objetivos particulares de la investigación, es necesario determinar los requisitos mínimos para la cantidad de datos, en otras palabras esta sección se encargará de establecer el número de imágenes necesarias para la clasificación de enfermedades utilizando GoogleNet como arquitectura.

Haciendo mención al conjunto de datos principal (Tabla 3.1), que contiene un total de 18,160 imágenes, se realizaron subconjuntos como se muestra en la Tabla 4.3, en donde el primer subconjunto contine solamente el 10%, el segundo 20% ... hasta llegar al 90% de los datos con base al total de imágenes.

Tabla 4.3 Subconjuntos

Subconjunto	Número total de imágenes
(1)	1,816
(2)	3,632
(3)	5,440
(4)	7,264
(5)	9,080
(6)	10,986
(7)	12,712
(8)	14,528
(9)	16,344

Luego, la metodología que se utilizó, el pre-procesamiento de datos, la configuración de hiperparámetros, el entrenamiento del modelo y las medidas de desempeño fueron las mismas que se mencionan en el **Capítulo 3**.

En este apartado, se observan tanto las Tablas (4.4, 4.6, 4.8, 4.10, 4.12, 4.14, 4.16, 4.18 y 4.20) de subconjuntos y los resultados que cada una de ellas obtuvieron.

Tabla 4.4 Subconjunto 1.

Etiqueta	Número de imágenes	80%	20%
(1)	213	160	45
(2)	100	80	20
(3)	191	153	38
(4)	95	76	19
(5)	177	142	35
(6)	168	135	33
(7)	140	112	28
(8)	536	429	107
(9)	37	30	7
(10)	159	128	31
Total	1,816	1,453	363

Tabla 4.5 Medidas de rendimiento para el subconjunto 1.

Medidas de rendimiento	Resultado
Exactitud	92.56 %
Precisión	82.32 %
Sensibilidad	83.45 %
Especificidad	99.16 %
F-Score	82.80 %
AUC	95.08 %
Tiempo	5 min 53 seg

Tabla 4.6 Subconjunto 2.

Etiqueta	Número de imágenes	80%	20%
(1)	425	340	85
(2)	200	160	40
(3)	382	306	76
(4)	191	153	38
(5)	354	283	71
(6)	335	268	67
(7)	281	225	56
(8)	1,071	856	215
(9)	75	60	15
(10)	318	255	63
Total	3,632	2,906	726

Tabla 4.7 Medidas de rendimiento para el subconjunto 2.

Medidas de rendimiento	Resultado
Exactitud	96.55 %
Precisión	94.93%
Sensibilidad	95.57 %
Especificidad	99.60 %
F-Score	95.11 %
AUC	97.14 %
Tiempo	13 min 6 seg

Tabla 4.8 Subconjunto 3.

Etiqueta	Número de imágenes	80%	20%
(1)	638	510	128
(2)	300	240	60
(3)	572	458	114
(4)	284	227	57
(5)	530	424	106
(6)	501	401	100
(7)	421	337	84
(8)	1,606	1,284	322
(9)	111	89	22
(10)	477	382	95
Total	5,440	4,352	1,088

Tabla 4.9 Medidas de rendimiento para el subconjunto 3.

Medidas de rendimiento	Resultado
Exactitud	97.79 %
Precisión	96.85 %
Sensibilidad	96.61 %
Especificidad	99.76 %
F-Score	96.71 %
AUC	98.47 %
Tiempo	22 min 39 seg

Tabla 4.10 Subconjunto 4.

Etiqueta	Número de imágenes	80%	20%
(1)	851	681	170
(2)	400	320	80
(3)	764	611	153
(4)	381	305	76
(5)	708	566	142
(6)	670	536	134
(7)	562	450	112
(8)	2,142	1,713	429
(9)	150	120	30
(10)	636	509	127
Total	7,264	5,811	1,453

Tabla 4.11 Medidas de rendimiento para el subconjunto 4.

Medidas de rendimiento	Resultado
Exactitud	97.93 %
Precisión	97.81 %
Sensibilidad	96.99 %
Especificidad	99.77 %
F-Score	97.34 %
AUC	99.12 %
Tiempo	32 min 55 seg

Tabla 4.12 Subconjunto 5.

Etiqueta	Número de imágenes	80%	20%
(1)	1,063	850	213
(2)	500	400	100
(3)	954	763	191
(4)	476	381	95
(5)	886	709	177
(6)	838	670	168
(7)	702	562	140
(8)	2,678	2,142	536
(9)	187	150	37
(10)	796	637	159
Total	9,080	7,264	1,816

Tabla 4.13 Medidas de rendimiento para el subconjunto 5.

Medidas de rendimiento	Resultado
Exactitud	98.56 %
Precisión	98.34 %
Sensibilidad	98.03 %
Especificidad	99.84 %
F-Score	98.18 %
AUC	99.36 %
Tiempo	32 min 55 seg

Tabla 4.14 Subconjunto 6 .

Etiqueta	Número de imágenes	80%	20%
(1)	1,276	1,021	255
(2)	600	480	120
(3)	1,145	916	229
(4)	571	457	114
(5)	1,063	850	213
(6)	1,006	805	201
(7)	842	674	168
(8)	3,214	2,571	643
(9)	224	179	45
(10)	955	764	191
Total	10,896	8,717	2,179

Tabla 4.15 Medidas de rendimiento para el subconjunto 6.

Medidas de rendimiento	Resultado
Exactitud	98.76 %
Precisión	98.39 %
Sensibilidad	98.11 %
Especificidad	99.86 %
F-Score	99.24 %
AUC	99.08 %
Tiempo	60 min 38 seg

Tabla 4.16 Subconjunto 7.

Etiqueta	Número de imágenes	80%	20%
(1)	1,489	1,191	298
(2)	700	560	140
(3)	1,336	1,069	267
(4)	666	533	133
(5)	1,240	992	1,048
(6)	1,173	938	235
(7)	983	787	196
(8)	3,750	3,000	750
(9)	261	209	52
(10)	1,114	891	223
Total	12,712	10,170	2,542

Tabla 4.17 Medidas de rendimiento para el subconjunto 7.

Medidas de rendimiento	Resultado
Exactitud	99.05 %
Precisión	98.76 %
Sensibilidad	98.77 %
Especificidad	99.89 %
F-Score	98.76 %
AUC	99.36 %
Tiempo	78 min 7 seg

Tabla 4.18 Subconjunto 8.

Etiqueta	Número de imágenes	80%	20%
(1)	1,701	1,361	341
(2)	800	640	160
(3)	1,527	1,222	305
(4)	762	610	152
(5)	1,417	1,134	283
(6)	1,341	1,072	269
(7)	1,123	898	225
(8)	4,285	3,428	857
(9)	299	239	60
(10)	1,273	1,018	255
Total	14,528	11,622	2,907

Tabla 4.19 Medidas de rendimiento para el subconjunto 8.

Medidas de rendimiento	Resultado
Exactitud	99.34 %
Precisión	99.10%
Sensibilidad	99.20 %
Especificidad	99.92 %
F-Score	99.15 %
AUC	99.50 %
Tiempo	94 min 36 seg

Tabla 4.20 Subconjunto 9.

Etiqueta	Número de imágenes	80%	20%
(1)	1,914	1,531	383
(2)	900	720	180
(3)	1,718	1,374	344
(4)	857	686	171
(5)	1,594	1,275	319
(6)	1,508	1,207	301
(7)	1,264	1,011	253
(8)	4,821	3,856	965
(9)	336	269	67
(10)	1,432	1,146	286
Total	16,344	13,075	3,269

Tabla 4.21 Medidas de rendimiento para el subconjunto 9.

Medidas de rendimiento	Resultado
Exactitud	98.74 %
Precisión	98.62 %
Sensibilidad	97.82 %
Especificidad	99.85 %
F-Score	98.20 %
AUC	99.43 %
Tiempo	112 min 36 seg

A continuación se presenta la “curva de aprendizaje de rendimiento” de los modelos, este tipo de curva es calculada sobre la métrica por la cual se evaluará y se seleccionará el modelo. En este caso, la evaluación de los resultados obtenidos se realizó a través del enfoque basado en la curva ROC, que se ha utilizado ampliamente para medir o visualizar el rendimiento del clasificador aunado con el AUC.

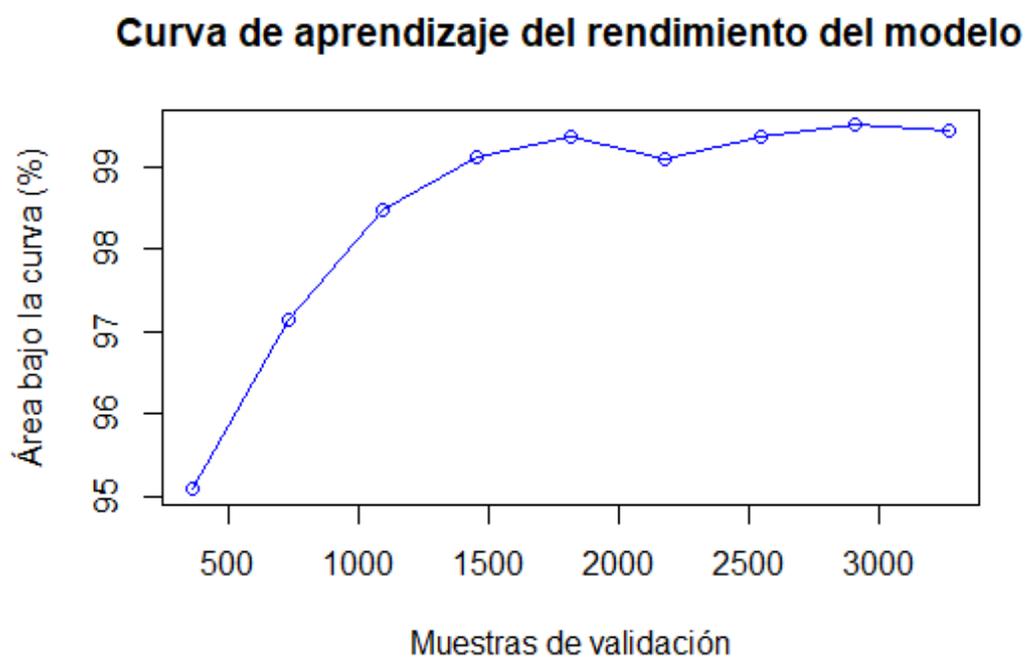


Figura 4.4 Curva de aprendizaje.

Tabla 4.22 Resultados generales de los modelos con base al *AUC*

Medida	1	2	3	4	5	6	7	8	9
<i>AUC</i>	95.08%	97.14%	98.47%	99.12%	99.36%	99.08%	99.36%	99.50%	99.43%

Como se puede observar en la Figura 4.4 y en la Tabla 4.22 todos los modelos alcanzaron un valor altamente significativo puesto que un valor arriba del 95% en el *AUC* se interpreta como excelente. Por otro lado, realizando la comparación de los mismos, los modelos que se desempeñaron de buena manera al momento de la validación fueron el 5, 7, 8 y 9. Sin embargo, con base a los parámetros estadísticos el modelo 8 es el mejor.

El objetivo de este apartado es establecer la mínima cantidad de imágenes con las cuales un modelo obtenga buen desempeño, el modelo del subconjunto 8 obtiene el 99.50% de *AUC* en 94 min 36 seg con 14,528 imágenes, no obstante implica mayor costo computacional, requiere una mayor cantidad de datos y más tiempo de procesamiento. Dada la importancia en relación a la cantidad de datos, se concluye que el modelo 5 es el mejor, debido a que logra el 99.36% de *AUC* en 32 min 55 seg, con 9,080 imágenes. La diferencia entre ambos es mínima pero el modelo 8 utiliza 5,448 de más y la desigualdad entre *AUCs* es de 0.14, lo cual se traduce a demasiado costo computacional y de procesamiento de información por muy poca mejora.

Por otro lado, el modelo 5, obtiene excelentes resultados a comparación de los modelos de CNN, AlexNet obtuvo el 99.28% (94.17 min), Inception V3 99.20% (649.85 min), y ResNet 18 99.20% (147.73 min) de *AUC* respectivamente. Por lo tanto, nuevamente se determina que el modelo 5 procesa de manera rápida, obtiene un buen balance de *AUC* y consigue resultados más altos.

Conclusiones

El Deep Learning ha logrado gran popularidad en el campo de investigación, debido al rápido crecimiento en el poder computacional. Por lo tanto, la presente investigación propone el enfoque del deep learning mediante la combinación de arquitecturas de CNN pre-entrenadas y la técnica de fine-tuning, para clasificar enfermedades en tomate, puesto que es necesario contar con modelos rápidos y precisos para detectar enfermedades a tiempo. Es por ello que el objetivo principal de este trabajo se centra en la comparación del desempeño de AlexNet, GoogleNet, Inception V3, ResNet 18 y ResNet 50, empleando diferentes métricas que evalúan su rendimiento. Los resultados experimentales demostraron que los modelos anteriormente mencionados fueron capaces de clasificar nueve enfermedades en hojas de tomate y una clase saludable y además, obtuvieron un rendimiento estadísticamente significativo, puesto que con base a la métrica seleccionada, el *AUC*, un valor arriba del 90% se interpreta como un excelente comportamiento.

En general, se concluye que el modelo que más se adecuó a la tarea fue GoogleNet con tan solo 22 capas y 7 millones de parámetros, logró un mayor desempeño en comparación a las demás, alcanzando un porcentaje de 99.72% presentando una mínima tasa de error de 0.28%, lo que demuestra que empleando la combinación de estas dos técnicas los modelos presentan porcentajes altamente deseables, de lo contrario Inception V3, una de las arquitecturas profundas obtuvo un bajo rendimiento, demostrando así que el número de capas no es directamente proporcional al desempeño del modelo.

A través del desarrollo de la investigación, se llevó a cabo una segunda experimentación, que consistía en encontrar la mínima cantidad de datos. Empleando la misma metodología pero

variando la cantidad de imágenes, se entrenaron nueve modelos utilizando GoogleNet como arquitectura. Es por ello, que se concluye de la segunda experimentación que el modelo 5 fue el indicado debido al bajo costo de entrenamiento, menor cantidad de imágenes y procesamiento, por lo que se determina como cantidad mínima 9,080 imágenes, dado que este modelo fue capaz de clasificar correctamente las clases de tomate, logrando el 99.36% de *AUC* con el 0.64% de error, y que además, obtiene mejores resultados en comparación a AlexNet, Inception V3 y ResNet 18, en relación al tiempo de procesamiento y al valor del *AUC*.

Se concluye que las tecnologías inteligentes pueden ayudar a proporcionar orientación y servir como apoyo a los agricultores con el objetivo de identificar a tiempo problemas en el campo, por ejemplo clasificar y detectar enfermedades a tiempo minimizando pérdidas económicas. Por lo tanto, se requiere que este tipo de herramientas lleguen a los agricultores, siendo efectivas y precisas.

Como **trabajo futuro** se propone aumentar el número de enfermedades a identificar con el objetivo de tener una mayor diversidad de datos, aunado a la creación e implementación de una aplicación móvil que detecte enfermedades en tiempo real y así ofrecer un sistema de diagnóstico completo a los agricultores. Por otro parte, respecto al cultivo de tomate, se propone crear una base de datos que abarque desde el inicio del cultivo hasta su cosecha. Finalmente, sería interesante crear una CNN y compararla con los resultados obtenidos.

Referencias

- [1] SAGARPA, “Planeación agrícola nacional 2017 - 2030.” [on line:] <https://www.gob.mx/sagarpa/documentos/planeacion-agricola-nacional-2017-2030>. Accessed: 2018-12-15.
- [2] SAGARPA, “Atlas agroalimentario 2012-2018.” [on line:] https://nube.siap.gob.mx/gobmx_publicaciones_siap/pag/2018/Atlas-Agroalimentario-2018. Accessed: 2018-09-21.
- [3] I. M. Hanssen and M. Lapidot, “Major tomato viruses in the mediterranean basin,” in *Advances in virus research*, vol. 84, pp. 31–66, Elsevier, 2012.
- [4] M. Brahim, K. Boukhalfa, and A. Moussaoui, “Deep learning for tomato diseases: classification and symptoms visualization,” *Applied Artificial Intelligence*, vol. 31, no. 4, pp. 299–315, 2017.
- [5] J. G. Barbedo, “Factors influencing the use of deep learning for plant disease recognition,” *Biosystems Engineering*, vol. 172, pp. 84–91, 2018.
- [6] H. Al-Hiary, S. Bani-Ahmad, M. Reyalat, M. Braik, and Z. ALRahamneh, “Fast and accurate detection and classification of plant diseases,” *International Journal of Computer Applications*, vol. 17, no. 1, pp. 31–38, 2011.
- [7] S. Verma, A. Chug, A. P. Singh, S. Sharma, and P. Rajvanshi, “Deep learning-based mobile application for plant disease diagnosis: A proof of concept with a case study on tomato plant,” in *Applications of Image Processing and Soft Computing Systems in Agriculture*, pp. 242–271, 2019.
- [8] A. Kamilaris and F. X. Prenafeta-Boldú, “Deep learning in agriculture: A survey,” *Computers and electronics in agriculture*, vol. 147, pp. 70–90, 2018.
- [9] O. de las Naciones Unidas para la Alimentación y Agricultura, “Plagas y enfermedades de las plantas.” [on line:] <http://www.fao.org/emergencias/tipos-de-peligros-y-de-emergencias/plagas-y-enfermedades-de-las-plantas/es/>. Accessed: 2018-09-21.
- [10] J. Ruiz Martínez, A. A. Vicente, J. C. Montañéz Saenz, R. Rodríguez Herrera, and C. N. Aguilar González, “Un tesoro perecedero en México: el tomate, tecnologías para prolongar su vida de anaquel,” *Investigación y Ciencia*, vol. 20, no. 54, pp. 57–63, 2012.

- [11] F. I. en Relacion con la Agricultura, “Panorama agroalimentario tomate rojo 2017.” [on line:] <https://www.fira.gob.mx/InfEspDtoXML/abrirArchivo.jsp?abreArc=65310>. Accessed: 2018-09-21.
- [12] F. E. Martínez-Ruiz, L. Cervantes-Díaz, C. E. Aíl-Catzím, L. G. Hernández-Montiel, C. L. D. T. Sánchez, and E. O. Rueda-Puente, “Hongos fitopatógenos asociados al tomate (*solanum lycopersicum* l.) en la zona árida del noroeste de México: la importancia de su diagnóstico,” *European Scientific Journal, ESJ*, vol. 12, no. 18, p. 232, 2016.
- [13] S. de Información Agroalimentaria y Pesquera, “Avance de siembras y cosechas resumen por estado.” [on line:] http://infosiap.siap.gob.mx:8080/agricola_siap_gobmx/ResumenDelegacion.do. Accessed: 2018-09-21.
- [14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, pp. 1097–1105, 2012.
- [15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.
- [16] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.
- [17] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- [18] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.
- [19] P. K. Gikunda and N. Jouandeau, “State-of-the-art convolutional neural networks for smart farms: A review,” in *Intelligent Computing-Proceedings of the Computing Conference*, vol. 997, pp. 763–775, 2019.
- [20] Y. Kawasaki, H. Uga, S. Kagiwada, and H. Iyatomi, “Basic study of automated diagnosis of viral plant diseases using convolutional neural networks,” in *International Symposium on Visual Computing*, vol. 9475, pp. 638–645, 2015.
- [21] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” in *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 675–678, 2014.

- [22] S. P. Mohanty, D. P. Hughes, and M. Salathé, “Using deep learning for image-based plant disease detection,” *Frontiers in plant science*, vol. 7, p. 1419, 2016.
- [23] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, “Deep neural networks based recognition of plant diseases by leaf image classification,” *Computational intelligence and neuroscience*, vol. 2016, pp. 1–12, 2016.
- [24] L. G. Nachtigall, R. M. Araujo, and G. R. Nachtigall, “Classification of apple tree disorders using convolutional neural networks,” in *Tools with Artificial Intelligence (ICTAI), 2016 IEEE 28th International Conference on*, pp. 472–476, 2016.
- [25] E. Fujita, Y. Kawasaki, H. Uga, S. Kagiwada, and H. Iyatomi, “Basic investigation on a robust and practical plant diagnostic system,” in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 989–992, 2016.
- [26] C. DeChant, T. Wiesner-Hanks, S. Chen, E. L. Stewart, J. Yosinski, M. A. Gore, R. J. Nelson, and H. Lipson, “Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning,” *Phytopathology*, vol. 107, no. 11, pp. 1426–1432, 2017.
- [27] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, “Identification of rice diseases using deep convolutional neural networks,” *Neurocomputing*, vol. 267, pp. 378–384, 2017.
- [28] G. Wang, Y. Sun, and J. Wang, “Automatic image-based plant disease severity estimation using deep learning,” *Computational intelligence and neuroscience*, vol. 2017, pp. 1–8, 2017.
- [29] M. Brahim, M. Arsenovic, S. Laraba, S. Sladojevic, K. Boukhalfa, and A. Moussaoui, “Deep learning for plant diseases: Detection and saliency map visualisation,” in *Human and Machine Learning*, pp. 93–117, 2018.
- [30] J. Wang, L. Chen, J. Zhang, Y. Yuan, M. Li, and W. Zeng, “Cnn transfer learning for automatic image-based classification of crop disease,” in *Chinese Conference on Image and Graphics Technologies*, pp. 319–329, 2018.
- [31] A. K. Rangarajan, R. Purushothaman, and A. Ramesh, “Tomato crop disease classification using pre-trained deep learning algorithm,” *Procedia computer science*, vol. 133, pp. 1040–1047, 2018.
- [32] I. Khandelwal and S. Raman, “Analysis of transfer and residual learning for detecting plant diseases using images of leaves,” in *Computational Intelligence: Theories, Applications and Future Directions-Volume II*, pp. 295–306, 2019.
- [33] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, p. 436, 2015.

- [34] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.
- [35] J. V. Haxby, E. A. Hoffman, and M. I. Gobbini, "The distributed human neural system for face perception," *Trends in cognitive sciences*, vol. 4, no. 6, pp. 223–233, 2000.
- [36] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.
- [37] K. Zhang, Q. Wu, A. Liu, and X. Meng, "Can deep learning identify tomato leaf disease?," *Advances in Multimedia*, vol. 2018, pp. 1–10, 2018.
- [38] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," *Computers and Electronics in Agriculture*, vol. 161, pp. 272–279, 2018.
- [39] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [40] A. Fuentes, S. Yoon, H. Youngki, Y. Lee, and D. Park, "Characteristics of tomato plant diseases—a study for tomato plant disease identification," in *Proc. Int. Symp. Inf. Technol. Converg*, vol. 1, pp. 226–231, 2016.
- [41] D. Blancard, *Tomato diseases: identification, biology and control: A Colour Handbook*. CRC Press, 2012.
- [42] D. Hughes, M. Salathé, *et al.*, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," *arXiv preprint arXiv:1511.08060*, 2015.
- [43] R. Kleinberg, Y. Li, and Y. Yuan, "An alternative view: When does sgd escape local minima?," *arXiv preprint arXiv:1802.06175*, 2018.
- [44] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.
- [45] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, 2009.
- [46] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.
- [47] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pp. 242–264, 2010.

- [48] A. Tharwat, “Classification assessment methods,” *Applied Computing and Informatics*, 2018.
- [49] A. P. Bradley, “The use of the area under the roc curve in the evaluation of machine learning algorithms,” *Pattern recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.
- [50] J. A. Hanley and B. J. McNeil, “The meaning and use of the area under a receiver operating characteristic (roc) curve.,” *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.
- [51] T. Fawcett, “An introduction to roc analysis,” *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.
- [52] D. J. Hand and R. J. Till, “A simple generalisation of the area under the roc curve for multiple class classification problems,” *Machine learning*, vol. 45, no. 2, pp. 171–186, 2001.
- [53] B. D. Ripley, “The r project in statistical computing,” *MSOR Connections. The newsletter of the LTSN Maths, Stats & OR Network*, vol. 1, no. 1, pp. 23–25, 2001.
- [54] X. Robin, N. Turck, A. Hainard, N. Tiberti, F. Lisacek, J.-C. Sanchez, and M. Müller, “proc: an open-source package for r and s+ to analyze and compare roc curves,” *BMC bioinformatics*, vol. 12, no. 1, p. 77, 2011.
- [55] M. Kuhn *et al.*, “Building predictive models in r using the caret package,” *Journal of statistical software*, vol. 28, no. 5, pp. 1–26, 2008.
- [56] M. Lin, Q. Chen, and S. Yan, “Network in network, corr abs/1312.4400,” *URL <http://arxiv.org/abs/1312.4400>*, 2013.