



**UNIVERSIDAD POLITÉCNICA SALESIANA  
SEDE QUITO**

**CARRERA DE INGENIERÍA ELECTRÓNICA**

**DISEÑO DE UN DETECTOR DEL HONGO MONILIASIS EN CULTIVOS DE  
CACAO PARA PEQUEÑOS PRODUCTORES**

Trabajo de titulación previo a la obtención  
del Título de Ingeniero Electrónico

**AUTOR:** Jefferson Paul Faconda Silva  
Edison Matias Suquillo Pulupa

**TUTOR:** Carlos Augusto Cuichán Morales

Quito - Ecuador

Mayo 2022

**CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE  
TITULACIÓN**

Nosotros, Jefferson Paul Faconda Silva con documento de identificación N° 1727219154 y Edison Matias Suquillo Pulupa con documento de identificación N° 1722214218; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 17 de mayo del año 2022.

Atentamente,



---

Jefferson Paul Faconda Silva  
1727219154



---

Edison Matias Suquillo Pulupa  
1722214218

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE  
TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Nosotros, Jefferson Paul Faconda Silva con documento de identificación N° 1727219154 y Edison Matias Suquillo Pulupa con documento de identificación N° 1722214218, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto Técnico: “Diseño de un detector del hongo moniliasis en cultivos de cacao para pequeños productores”, el cual ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

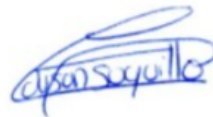
Quito, 17 de mayo del año 2022.

Atentamente,



---

Jefferson Paul Faconda Silva  
1727219154



---

Edison Matias Suquillo Pulupa  
1722214218

## **CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN**

Yo, Carlos Augusto Cuichán Morales con documento de identificación N° 1714389721, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DISEÑO DE UN DETECTOR DEL HONGO MONILIASIS EN CULTIVOS DE CACAO PARA PEQUEÑOS PRODUCTORES, realizado por Jefferson Paul Faconda Silva con documento de identificación N° 1727219154 y por Edison Matias Suquillo Pulupa con documento de identificación N° 1722214218, obteniendo como resultado final el trabajo de titulación bajo la opción proyecto técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 17 de mayo del año 2022.

Atentamente,



---

Ing. Carlos Augusto Cuichán Morales, MSc  
1714389721

## ÍNDICE DE CONTENIDO

<b>CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN</b> .....	i
<b>CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA</b> .....	ii
<b>CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN</b> .....	iii
<b>ÍNDICE DE CONTENIDO</b> .....	iv
<b>ÍNDICE DE FIGURAS</b> .....	vii
<b>ÍNDICE DE TABLAS</b> .....	x
<b>RESUMEN</b> .....	xi
<b>ABSTRAC</b> .....	xii
<b>INTRODUCCIÓN</b> .....	1
<b>CAPITULO 1</b> .....	3
<b>ANTECEDENTES</b> .....	3
1.1 PLANTEAMIENTO DEL PROBLEMA .....	3
1.2 JUSTIFICACIÓN .....	3
1.3 OBJETIVOS .....	4
1.3.1 Objetivo General .....	4
1.3.2 Objetivos Específicos .....	4
<b>CAPITULO 2</b> .....	5
<b>MARCO TEÓRICO</b> .....	5
2.2 SECTOR CACAOTERO EN EL ECUADOR .....	5
2.2 CARACTERÍSTICAS DEL HONGO MONILIASIS .....	5
2.2.1 Moniliasis .....	5
2.2.2 Síntomas del Hongo Moniliasis .....	6
2.2.3 Ciclo de vida del Hongo Moniliasis .....	6
2.2.4 Daños ocasionados por el Hongo Moniliasis .....	7
2.3 CIFRAS DE PÉRDIDAS DE CACAO POR MONILIASIS EN SANTO DOMINGO DE LOS TSÁCHILAS .....	7
2.4 TRATAMIENTO CLÁSICO DEL HONGO MONILIASIS .....	8
2.4.1 Control cultural .....	8
2.4.2 Control químico .....	9
2.4.3 Daños derivados al control cultural y químico en el cacao .....	10

2.5 TRATAMIENTO TECNOLÓGICO DEL HONGO MONILIASIS.....	10
2.5.1 Uso de drones.....	10
2.5.2 Uso de sensores.....	11
2.5.3 IoT para el control de plagas.....	11
2.6 PROCESAMIENTO DE IMÁGENES.....	11
2.7 IMPORTANCIA DE UN ALGORITMO .....	11
2.8 MATLAB .....	12
2.8.1 Reconocimiento de imágenes con MATLAB.....	12
2.9 TÉCNICAS DE PROCESAMIENTO DE IMÁGENES PARA EL RECONOCIMIENTO DE IMÁGENES. ....	13
2.9.1 Reconocimiento de imágenes mediante Machine Learning .....	13
2.9.2 Reconocimiento de imágenes mediante técnicas tradicionales de procesamiento de imágenes .....	13
2.9.3 Reconocimiento de imágenes mediante Deep Learning.....	14
2.9.4 Transferencia de Aprendizaje .....	14
2.9.5 Red neuronal Convolutiva.....	15
2.9.6 Red GoogLeNet .....	16
2.9.7 Resnet_18.....	16
<b>CAPITULO 3.....</b>	<b>17</b>
<b>DESARROLLO DEL ALGORITMO DE PROCESAMIENTO DE IMÁGENES. ....</b>	<b>17</b>
3.1 DESCRIPCIÓN DE LOS REQUERIMIENTOS EN LAS DISTINTAS ETAPAS PARA EL DE DESARROLLO DEL ALGORITMO. ....	17
3.1.1 Definir tipo de análisis.....	18
3.1.2 Crear una base de datos.....	19
3.1.3 Preparar imágenes de entrada con sus etiquetas .....	20
3.1.4 Normalizar Datos .....	20
3.1.5 Definir Arquitectura de la Red.....	21
3.1.6 Selección de Opciones de Entrenamiento de la Red Neuronal.....	21
3.1.7 Evaluar Estadísticas de Red Neuronal .....	21
3.1.8 Analizar Resultados de Clasificación .....	21
3.1.9 Guardar e implementar Red Neuronal .....	22
3.2 ALGORITMO Y LÓGICA DE PROGRAMACIÓN .....	22
3.2.1 Selección de Imágenes.....	22
3.2.2 Aumento de datos de imágenes para entrenamiento de la red.....	23

3.2.3 Creación de la Red Neuronal Convolutiva mediante transferencia de aprendizaje GoogLeNet.....	24
3.2.4 Creación de la Red Neuronal Convolutiva mediante transferencia de aprendizaje ResNet-18 .....	29
3.2.5 Creación de la Red Neuronal Convolutiva Básica.....	29
3.3 COMPILACIÓN Y DESARROLLO.....	32
3.3.1 Verificación de la red.....	32
3.3.2 Tiempos de Procesamiento .....	33
3.4 INTERFAZ GRÁFICA E IMPLEMENTACIÓN DE LA RED .....	34
<b>CAPÍTULO 4.....</b>	<b>37</b>
<b>PRUEBAS Y RESULTADOS DE LOS ALGORITMOS PROPUESTOS.....</b>	<b>37</b>
4.1 PRUEBAS REALIZADAS .....	39
4.1.1 Rendimiento de distintos tipos de redes.....	39
4.1.2 Exactitud con distintos tamaños de imágenes y distancia variable.....	43
4.1.3 Comparación del comportamiento de las redes.....	43
4.2 RESULTADOS .....	45
4.2.1 Rendimiento de distintos tipos de redes.....	45
4.2.2 Exactitud con distintos tamaños de imágenes y distancia variable.....	50
4.2.3 Comparación del comportamiento de las redes.....	56
4.2.4 Resultados Obtenidos.....	68
<b>CAPÍTULO 5.....</b>	<b>71</b>
<b>CONCLUSIONES Y RECOMENDACIONES.....</b>	<b>71</b>
5.1 CONCLUSIONES.....	71
5.2 RECOMENDACIONES .....	72
<b>REFERENCIAS .....</b>	<b>73</b>
ANEXO 1: RESULTADOS DE LOS ENTRENAMIENTOS DE LAS DISTINTAS REDES.....	75
A. Básica .....	75
B. GoogLeNet.....	76
C. RestNet-18.....	77

## ÍNDICE DE FIGURAS

Figura 2.1 Ciclo de vida del hongo moniliasis en el cacao .....	6
Figura 2.2 Daños que ocasiona la Moniliasis.....	7
Figura 2.3 Podas de mantenimiento en el cacao .....	9
Figura 2.4 Red Neuronal Convolutacional.....	15
Figura 3.1 Cuadro de procedimiento para la implementación del algoritmo.....	17
Figura 3.2 Fase 1 .....	18
Figura 3.3 Fase 2 .....	18
Figura 3.4 Fase 3 .....	19
Figura 3.4 Almacenamiento de imágenes .....	20
Figura 3.5 Creación del imds .....	22
Figura 3.6 conteo de imágenes .....	22
Figura 3.7 Dividir Datos.....	23
Figura 3.8 Aumento de Imágenes .....	23
Figura 3.9 Tamaño de las imágenes .....	23
Figura 3.10 Creación de la base de dato con el aumento de imágenes .....	24
Figura 3.11 Red existente GoogLeNet .....	24
Figura 3.12 Primera Capa de la red GoogLeNet.....	25
Figura 3.13 Capa “fullyConnectedLayer” por defecto de la red GoogLeNet.....	26
Figura 3.14 Capa “fullyConnectedLayer” modificada de la red GoogLeNet.....	26
Figura 3.15 Capa “classificationLayer” modificada de la red GoogLeNet.....	27
Figura 3.16 Importación de la base de datos “auds_test” y “auds_validation”.....	27
Figura 3.17 Visualización de la base de datos “auds_test” .....	28
Figura 3.18 Parámetros de entrenamiento para la CNN.....	28
Figura 3.19 CNN con ResNet_18 .....	29
Figura 3.20 CNN Básica .....	31
Figura 3.21 Redes guardadas .....	32
Figura 3.22 Importar red entrenada.....	32
Figura 3.23 Número de etiquetas establecidas “4” .....	33
Figura 3.24 Nombre de las etiquetas establecidas.....	33
Figura 3.25 Tiempo de Procesamiento.....	34



Figura 3.26 Portada .....	35
Figura 3.27 Ventana de selección. ....	35
Figura 3.28 Selección de la imagen. ....	36
Figura 3.29 Resultado obtenido. ....	36
Figura 4.1 Computadora usada para entrenamientos .....	37
Figura 4.2 Tarjeta de video disponible en el PC .....	38
Figura 4.3 Maquina 1 utilizada para evaluar redes neuronales .....	38
Figura 4.4 Maquina 2 utilizada para evaluar redes neuronales .....	39
Figura 4.5 Imágenes de frutos de cacao para entrenamiento. ....	40
Figura 4.6 Imágenes de cacaos para pruebas. ....	42
Figura 4.7 Imágenes para pruebas.....	44
Figura 4.8 Características de la red básica. ....	45
Figura 4.9 Proceso de entrenamiento de la red básica. ....	46
Figura 4.10 Proceso de entrenamiento de la red GoogLeNet .....	48
Figura 4.11 Entrenamiento de la red Resnet-18.....	49
Tabla 4.11 Exactitud de las redes en imágenes de 6[MPx] a tres distancias. ....	51
Figura 4.12 Exactitud de las redes propuestas con imágenes a 6[MPx] .....	51
Figura 4.13 Exactitud de las redes propuestas con imágenes a 20[MPx]. ....	53
Figura 4.14 Exactitud de las redes propuestas con imágenes a 48[MPx]. ....	55
Figura 4.15 Matriz de confusión para la red básica. ....	57
Figura 4.16 Técnicas de visualización para cacaos buenos. Red básica. ....	57
Figura 4.17 Técnicas de visualización para cacaos fase 1. Red básica. ....	58
Figura 4.18 Técnicas de visualización para cacaos fase 2. Red básica. ....	59
Figura 4.19 Técnicas de visualización para cacaos fase 3. Red básica. ....	59
Figura 4.20 Matriz de confusión para la red Googlenet.....	60
Figura 4.21 Técnicas de visualización para cacaos buenos. Red Googlenet. ....	61
Figura 4.22 Técnicas de visualización para cacaos fase 1. Red Googlenet. ....	61
Figura 4.23 Técnicas de visualización para cacaos fase 2. Red Googlenet. ....	62
Figura 4.24 Técnicas de visualización para cacaos fase 3. Red Googlenet. ....	63
Figura 4.25 Matriz de confusión para la red Resnet_18. ....	64
Figura 4.26 Técnicas de visualización para cacaos sanos. Red Resnet_18.....	65
Figura 4.27 Técnicas de visualización para cacaos fase 1. Red Googlenet. ....	65
Figura 4.28 Técnicas de visualización para cacaos fase 2. Red Googlenet. ....	66
Figura 4.29 Técnicas de visualización para cacaos fase 3. Red Googlenet. ....	67

Figura 4.30 Tiempo para reconocimiento de cada cacao. PC1. ....	68
Figura 4.31 Resultado del Cacao sin el hongo Moniliasis. ....	69
Figura 4.32 Resultado del Cacao con el hongo Moniliasis en Fase 1. ....	69
Figura 4.33 Resultado del Cacao con el hongo Moniliasis en Fase 2. ....	69
Figura 4.34 Resultado del Cacao con el hongo Moniliasis en Fase 3. ....	70
Figura 4.35 Resultado al cargar una imagen diferente al cacao. ....	70

## ÍNDICE DE TABLAS

Tabla 2.1 Porcentaje de infestación de moniliasis .....	8
Tabla 2.2 Toxicidad de Fungicidas .....	10
Tabla 4.1. Número de imágenes de la base de datos de entrenamiento. ....	40
Tabla 4.2. Base de datos de imágenes en 6 Megapíxeles.....	41
Tabla 4.3. Base de datos de imágenes en 20 Megapíxeles.....	41
Tabla 4.4. Base de datos de imágenes en 48 Megapíxeles.....	41
Tabla 4.5. Parámetros de entrenamiento de la red básica .....	45
Tabla 4.6 Resultados de entrenamiento de la red básica.....	47
Tabla 4.7 Parámetros de entrenamiento de la red GoogLeNet. ....	47
Tabla 4.8 Resultados de la red GoogLeNet.....	48
Tabla 4.9 Parámetros de entrenamiento de la red Resnet_18. ....	49
Tabla 4.10 Resultados de entrenamiento de la red Resnet_18. ....	50
Tabla 4.11 Exactitud de las redes en imágenes de 6[MPx] a tres distancias. ....	51
Tabla 4.12. Tiempo de ejecución en PC1. Calidad 6 [MPx].....	52
Tabla 4.13 Exactitud de las redes en imágenes de 20[MPx] a tres distancias. ....	53
Tabla 4.14. Tiempo de ejecución en PC1. Calidad 20 [MPx].....	54
Tabla 4.15. Exactitud de las redes en imágenes de 48[MPx] a tres distancias. ....	54
Tabla 4.16. Tiempo de ejecución en PC1. Calidad 48 [MPx].....	56
Tabla 4.17. Tiempo para reconocimiento de cada cacao. PC1. ....	67

## RESUMEN

En la actualidad el sector cacaotero en Ecuador ocupa grandes áreas, en la provincia de Sto. Domingo de los Tsáchilas parroquia Valle Hermoso existe gran cantidad de pequeños productores de cacao, para quienes la detección del Hongo Moniliasis es de gran complejidad, sin embargo, realizan un control eficiente de manera manual para eliminar el hongo. Para este trabajo se define tres fases del hongo que varía por el porcentaje del tamaño, de la mancha y la aparición del polvo gris en la mazorca de cacao.

El presente trabajo tiene como objetivo el diseño de un detector del hongo moniliasis en cultivos de cacao para pequeños productores mediante la utilización del software Matlab el cual incluye técnicas de procesamiento de imágenes, se investiga los diferentes métodos que se pueden utilizar para la detección de imágenes, optando por la opción Deep Learning la cual permite crear redes neuronales convolucionales básicas y utilizar redes convolucionales ya creadas que se pueden adaptar a una nueva tarea mediante la transferencia de aprendizaje.

Se crean y se adaptan varios algoritmos para detectar con severidad la mancha negra con polvo gris que es causada por el Hongo Moniliasis en la mazorca de cacao, para el desarrollo de estos algoritmos se elige tres redes neuronales convolucionales las cuales son: Red Neuronal Básica que es diseñada desde cero, Red Neuronal Convolutacional GoogLeNet y Red Neuronal Convolutacional ResNet-18, estas dos últimas redes se encuentran disponibles en la base de datos de Matlab.

Utilizando la herramienta App Designer de Matlab se crea una interfaz amigable para el usuario, en la cual se selecciona una imagen de la mazorca de cacao y esta procede a definir la fase en la que se encuentra el hongo.

## ABSTRAC

Currently the cocoa sector in Ecuador occupies large areas, in the province of Sto. Domingo de los Tsáchilas, Valle Hermoso parish, there are a large number of small cocoa producers, for whom the detection of the Moniliasis Fungus is highly complex, however, they carry out an efficient control manually to eliminate the fungus. For this work, three phases of the fungus are defined that vary by the percentage of the size, the stain and the appearance of the gray powder in the cocoa pod.

The present work has as objective the design of a detector of the moniliasis fungus in cocoa crops for small producers through the use of Matlab software which includes image processing techniques, the different methods that can be used for the detection of images are investigated. , opting for the Deep Learning option which allows you to create basic convolutional neural networks and use already created convolutional networks that can be adapted to a new task by transferring learning.

Several algorithms are created and adapted to severely detect the black spot with gray dust that is caused by the Moniliasis Fungus in the cocoa pod. For the development of these algorithms, three convolutional neural networks are chosen, which are: Basic Neural Network that is designed from scratch, GoogLeNet Convolutional Neural Network and ResNet-18 Convolutional Neural Network, these last two networks are available in the Matlab database.

Using the Matlab App Designer tool, a user-friendly interface is created, in which an image of the cocoa pod is selected and it proceeds to define the phase in which the fungus is found.

## **INTRODUCCIÓN**

En los últimos años, los avances tecnológicos se han desarrollado a tal punto que las actividades que tradicionalmente hacían los seres humanos son realizadas por un sistema automatizado. La tecnología está inmersa en casi cualquier campo o ciencia que tenga como objetivo resolver y facilitar un problema en particular. En este sentido también, su aplicación está presente en el campo agrícola.

En la actualidad los cultivos agrícolas cacaoteros ocupan grandes áreas en la región costera del Ecuador, por lo cual el mapeo de los cultivos y la detección de plagas son difíciles de realizar por los pequeños productores de cacao, la detección del hongo debe ser eficiente y se lo hace de forma manual para que la producción tenga mayores rendimientos.

Debido a esto, en el presente trabajo se plantea incorporar el procesamiento de imágenes mediante redes neuronales convoluciones (CNN, por sus siglas en inglés) para la detección del hongo Moniliasis en el cacao, para lo cual se desarrolla el algoritmo en el software MatLab.

Los capítulos del presente proyecto técnico están organizados de la siguiente manera:

### **CAPÍTULO 1: Antecedentes**

Se explica el Planteamiento del Problema, Justificación y Objetivos.

### **CAPÍTULO 2: Marco Teórico**

En este capítulo se investiga sobre el sector cacaotero en el Ecuador con la influencia del hongo Moniliasis, de igual manera el procesamiento de imágenes con técnicas empleadas en la actualidad.

### **CAPÍTULO 3: Desarrollo del Algoritmo de Procesamiento**

Este capítulo trata sobre el desarrollo del algoritmo propuesto mediante redes neuronales convolucionales (CNN) especificando los diferentes parámetros para la detección de las diferentes etapas del Hongo Moniliasis.

### **CAPÍTULO 4: Pruebas y Resultados**

En este capítulo se explica a detalle las diferentes pruebas que se realizan al algoritmo para su verificación.

## CAPÍTULO 5: Conclusiones y Recomendaciones

Por último, se trata las conclusiones y recomendaciones pertinentes del trabajo.

# **CAPITULO 1**

## **ANTECEDENTES**

En el presente capítulo se describe el planteamiento del problema, la justificación, los objetivos a realizar y la metodología empleada.

### **1.1 PLANTEAMIENTO DEL PROBLEMA**

El cultivo de cacao en Ecuador está representado principalmente por pequeños productores, aproximadamente el 70%, lo cual representa uno de los principales rubros del sector Agropecuario del Ecuador.

Las principales problemáticas a las que se ven expuestos los pequeños productores de cacao, en la actualidad ya existen dispositivos automatizados capaces de detectar este tipo de hongo como son los sensores inteligentes que son capaces de monitorear hasta 6 lecturas que envían los datos a la nube para su visualización desde un smartphone, PC, Tablet, etc. De igual manera existen Drones de pulverización agrícola de alta calidad para fumigación, pero estos dispositivos mencionados tienen un alto costo en el mercado agrícola de Precisión en el Ecuador lo cual lo hace inaccesibles para los pequeños productores. Además de esto, dichos dispositivos requieren de un sistema de detección, el cual puede ser constituido por la propuesta realizada en este proyecto.

Este proyecto es de bajo costo, para así brindar la posibilidad al pequeño productor de cacao de detectar de manera manual el hongo Moniliasis, mediante un algoritmo que se basa en el procesamiento de imágenes de alta calidad.

### **1.2 JUSTIFICACIÓN**

En la actualidad, el sector agrícola es una parte fundamental del ingreso económico del país, especialmente los productores de cacao que generan aproximadamente \$157 millones de dólares en el mercado interno del Ecuador (Sánchez et al., 2020) ,donde los mayores productores de cacao son pequeños agricultores para los cuales representa su principal ingreso económico y por lo cual no cuenta con la tecnología adecuada para la detección a tiempo de este tipo de hongo “Moniliasis”, por este motivo se pretende desarrollar un sistema para la detección de la enfermedad fúngica “Moniliasis”, en el cacao; el cual se basa en un algoritmo



de procesamiento de imágenes basado en fotografías de alta calidad, las cuales permiten una detección temprana del hongo permitiendo al agricultor tomar medidas adecuadas.

También este sistema puede adaptarse a equipos automatizados que traten este tipo de plagas.

## **1.3 OBJETIVOS**

### **1.3.1 Objetivo General**

Diseñar un detector basado en un algoritmo mediante el procesamiento de imágenes de alta calidad para la identificación del hongo Moniliasis.

### **1.3.2 Objetivos Específicos**

- Analizar las características del hongo Moniliasis en el Ecuador para la determinación de los requerimientos de un sistema de detección.
- Registrar las diferentes etapas de la enfermedad provocada por el hongo Moniliasis mediante imágenes para el reconocimiento de las principales características de la enfermedad en la planta de cacao.
- Simular el detector diseñado en software Matlab para la identificación del Hongo Moniliasis.
- Verificar el algoritmo para la determinación de su funcionamiento mediante imágenes de alta calidad.

## **CAPITULO 2**

### **MARCO TEÓRICO**

#### **2.2 SECTOR CACAOTERO EN EL ECUADOR**

El sector cacaotero se desarrolla en latitudes que van desde los 10 grados al norte hasta los 10 grados al sur de la línea ecuatorial. Este fruto produce granos que son utilizados en las industrias para producir chocolates y grasas en las industrias cosmetológicas.

Ecuador es uno de los principales exportadores de cacao (CCN51 y Sabor Arriba), lo cual lo convierte en uno de los exportadores más competitivos en América latina en el sector cacaotero. Datos de ICCO (International Cocoa Organization), revela que Ecuador produce el 7% a nivel mundial. Según ANECACAO la producción cacaotera se distribuye con el 70% en pequeños productores, el 20% en medianos y el 10% sobrante en grandes productores.

El sistema SAIKU del Servicio de Rentas Internas-SRI presento información de las empresas dedicadas a la producción de cacao a nivel nacional del año 2019, registraron un ingreso de \$157 millones de dólares americanos por la venta de cacao del cual el 33,07% pertenece a Guayaquil siendo la principal provincia productora de cacao, seguido del 23,89% perteneciente a la provincia de Los Ríos, Santo Domingo de los Tsáchilas pertenece el 10,40%, a Esmeraldas 10,13% y Manabí 6,01%. (Sánchez et al., 2020)

#### **2.2 CARACTERISTICAS DEL HONGO MONILIASIS**

Uno de los principales problemas en el sector cacaotero en el Ecuador es el hongo moniliasis por lo cual se investiga sobre sus principales características.

##### **2.2.1 Moniliasis**

El primer hallazgo que se tiene de este hongo data al año 1895 en la provincia de Los Ríos en Ecuador, donde se cosecharon mazorcas con síntomas de la Moniliasis, pero desapareció sin causar daños en la plantación. Años más tarde, esta enfermedad produjo varias pérdidas en las plantaciones de cacao haciendo que tenga bajos rendimientos en la producción lo que provocó pérdidas económicas para los agricultores.

Esta enfermedad es causada por el hongo *Moniliophthora roreri*, comúnmente nombrada monilla, ataca a las mazorcas en cualquier etapa del desarrollo, siendo más vulnerable cuando menor sea el crecimiento. (Balón González & Donoso Bruque, 2017).

### 2.2.2 Síntomas del Hongo Moniliasis

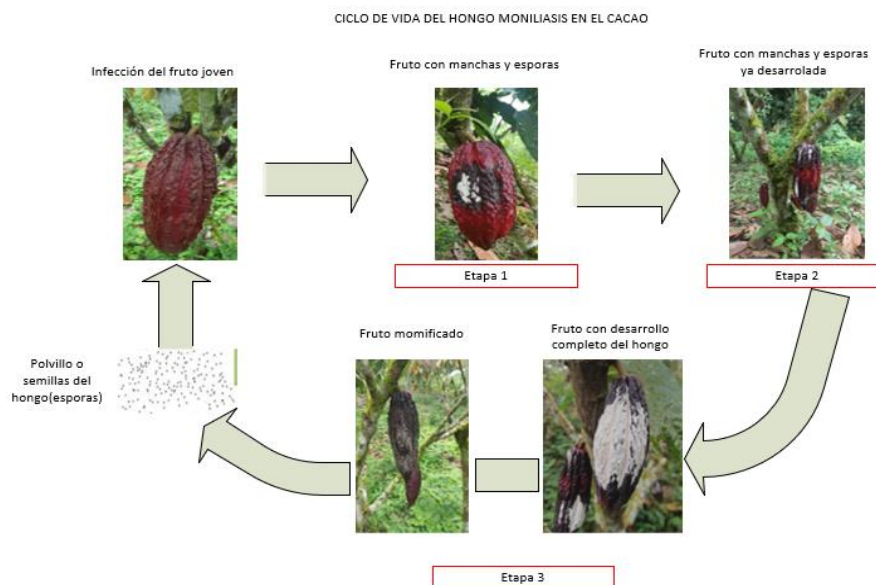
Los síntomas se presentan en mazorcas adultas (tres meses en adelante). El síntoma más común de la Moniliasis es una mancha de color oscura, que se expande hasta cubrir todo el fruto. Otra señal común de la Moniliasis es el aspecto de madurez prematura, lo que significa que los frutos cambian de color, dando la impresión de madurez normal en las mazorcas que todavía no tienen el tamaño ni la edad para ser cosechadas. (Balón González & Donoso Bruque, 2017).

### 2.2.3 Ciclo de vida del Hongo Moniliasis

El tiempo de desarrollo del hongo en la mazorca donde presenta síntomas visibles de la enfermedad comúnmente es de un mes y el tiempo del desarrollo completo de la enfermedad es alrededor de tres meses.

Como se observa en la figura 2.1, los indicios de esta enfermedad varían dependiendo de la edad de la mazorca. En las mazorcas que tienen menos de tres meses de edad, el primer síntoma de la infección es la aparición de una mancha color oscuro con esporas color blanco (etapa 1). Con el pasar del tiempo la infección se torna más visible ya que la mancha oscura y las esporas se expande por todo el fruto (etapa 2), finalmente con el pasar del tiempo las esporas están maduras provocando que la mazorca se momifique, esta última etapa de la enfermedad es grave ya que puede infectar a los frutos sanos por las esporas que se transporta con el viento (etapa 3). (Enríquez, 2019).

Figura 2.1 Ciclo de vida del hongo moniliasis en el cacao



Elaborado por: Jefferson Faconda y Edison Suquillo

## 2.2.4 Daños ocasionados por el Hongo Moniliasis

La moniliasis es una enfermedad exclusiva del fruto que determina su podredumbre total o parcial en función de la edad en la que se contagie. Las frutas infectadas antes de los tres meses de edad pierden el 100% de sus granos, mientras que las frutas infectadas pierden solo hasta el 10% después de los cuatro meses como se observa en la figura 2.2 La moniliasis no solo afecta al rendimiento de los cultivos sino también a la calidad de la materia prima ya que el productor, en su deseo de explotar parte de la fruta enferma, quita y extrae los granos fácilmente desprendibles mezclándolos con los granos de frutos sanos.(Figuroa Yovanny, 2017)

Figura 2.2 Daños que ocasiona la Moniliasis.



Elaborado por: Jefferson Faconda y Edison Suquillo

## 2.3 CIFRAS DE PÉRDIDAS DE CACAO POR MONILIASIS EN SANTO DOMINGO DE LOS TSÁCHILAS

Un estudio realizado por el INEC en el 2020 hay una superficie plantada de cacao alrededor de 590.579 hectáreas a nivel nacional, del cual el 4.79% de esta superficie representa a la provincia Santo Domingo de los Tsáchilas.(INEC, 2021).

Una encuesta realizada al “Santo Cacao” del GAD provincial, los resultados mostraron que debido al hongo moniliasis el 47,2% de los productores pierden aproximadamente un saco y medio alrededor de 50 – 100 libras por hectárea, pero el 44,9% afirma que pierde de uno a tres sacos alrededor de uno a tres quintales por hectárea.(Celi & Figuroa, 2017).

En la siguiente tabla se aprecia en porcentaje la infección del hongo moniliasis en la provincia de Sto. Domingo de los Tsáchilas, parroquia Valle hermoso.

Tabla 2.1 Porcentaje de infestación de moniliasis

Parroquia	Monilla
Valle Hermoso	10.04 %
La Concordia	12.82 %
Santo Domingo	12.89 %
El Esfuerzo	20.77 %
Puerto Limón	28.43 %
San Jacinto	35.6 %
Nuevo Israel	33.08 %
Luz de América	32.60 %
Santa María del Toachi	47.15 %

Tabla Porcentaje de infestación de moniliasis en las parroquias de Santo Domingo de los Tsáchilas Fuente:(Celi & Figueroa, 2017).

## 2.4 TRATAMIENTO CLÁSICO DEL HONGO MONILIASIS

Para el control de enfermedades en el sector cacaotero el agricultor debe tener en cuenta que es más importante la prevención que el control, así mismo la correcta aplicación de prácticas culturales como es la poda, control de malezas, fertilización y como última opción debe ser la aplicación de plaguicidas químicos. (de Ciencias et al., 2016).

### 2.4.1 Control cultural

El control cultural se fundamenta en utilizar diferentes prácticas agrícolas para alterar el ambiente en el cual se desarrolla la Moniliasis con la finalidad de evitar la propagación en el cultivo de cacao se realiza las siguientes actividades:

- Las plantas deben tener una fertilización adecuada, debido a que las plantas bien fertilizadas son menos propensas al ataque plagas y enfermedades.
- El agricultor debe ser capaz de diferenciar las distintas enfermedades.

- Para mantener la aireación en la planta de cacao se deben realizar podas de mantenimiento como se puede observar en la figura 3.
- Remover las mazorcas con cualquier síntoma de la enfermedad para evitar que dicha mazorca se convierta en una fuente de infección.
- En épocas de lluvia es recomendable realizar el control de enfermedades al menos 2 veces a la semana, en las demás épocas del año es aconsejable realizar esta labor cada 10 días. (De & Tsáchilas, 2017).

Figura 2.3 Podas de mantenimiento en el cacao



Elaborado por: Jefferson Faconda y Edison Suquillo

#### **2.4.2 Control químico**

Este tipo de control se basa en la utilización de químicos que ayuden al control del hongo Moniliasis, dichos compuestos pueden parar el crecimiento y reproducción del hongo. Es aconsejable la utilización de fungicidas protectantes y sistémicos, la aplicación de los fungicidas protectantes deben ser quincenales. En cambio, los fungicidas sistémicos deben ser utilizados en intervalos más largos dependiendo de sus especificaciones. (De & Tsáchilas, 2017). Dependiendo del producto y las condiciones climáticas variarán la dosis, en productos en polvo pudiendo variar de 1 a 2.5 kg/ha y en soluciones de 250 cc/ha. (de Ciencias et al., 2016).

En la siguiente tabla se presenta la toxicidad de los fungicidas dependiendo el color de su tapa.

Tabla 2.2 Toxicidad de Fungicidas

Agroquímicos	
Etiqueta verde	no tóxico
Etiqueta azul	ligeramente tóxico
Etiqueta amarilla	tóxico
Etiqueta Roja	altamente tóxico

Fuente:(De & Tsáchilas, 2017).

### 2.4.3 Daños derivados al control cultural y químico en el cacao

Técnicamente, los niveles de daño debido al control cultural se asocian a la falta de conocimiento del personal para mantener la inocuidad del producto durante todo el desarrollo de la mazorca de cacao (SENASA, 2017). Por otro lado, los daños derivados por el control químico se desarrollan a largo plazo debido a que trae consecuencias adversas como la contaminación ambiental, las plagas se hacen más resistentes y eliminación de insectos polinizadores.(Figueroa Yovanny, 2017).

## 2.5 TRATAMIENTO TECNOLÓGICO DEL HONGO MONILIASIS

Los avances tecnológicos en la actualidad permiten controlar el Hongo Moniliasis en el sector Agrícola.

### 2.5.1 Uso de drones.

Los drones más recomendables en este campo son el multirrotor-cuadrícóptero (que tiene un tiempo aproximado de vuelo de 30 minutos y cobertura por vuelo de 65 hectáreas) y el de ala fija (tiempo promedio de vuelo de 30 a 90 minutos y cobertura por vuelo de 120 a 3.800 hectáreas). Disponen de un aspecto importante como son los sensores que utiliza, que capturan imágenes rojo-verde-azul (RGB) e infrarrojo cercano (NIR). Tomando fotografías que deben venir geolocalizadas, para ser ubicadas exactamente y formar un mapa de plantación, posteriormente ser sobrepuestas. Existen cámaras de tipo agrícola con filtros especializados que las hacen más costosas.(Edwin, 2019).

### **2.5.2 Uso de sensores.**

Se utiliza distintos tipos de sensores térmicos o lumínicos para detectar plagas, dichos sensores pueden ser instalados en drones para la detección remota de alteraciones de los cultivos. Que almacenan la información adquirida, permite la confección de mapas de infección y, en ocasiones, la delimitación de focos a tratar, evitando tratamientos generales del cultivo. (Planas de Martí, 2018).

### **2.5.3 IoT para el control de plagas.**

El Internet de las cosas (IoT) es un conjunto de objetos como sensores y actuadores conectados a Internet a través de redes fijas e inalámbricas donde pueden adquirir datos de forma autónoma y autoconfigurarse de forma inteligente en base a hechos del mundo físico; permitiendo que estos sistemas se conviertan en participantes activos en diversos procesos públicos, comerciales, científicos y personales. El IoT está impulsado por una combinación de métodos y procesos que incluyen el uso de sensores, una red y un dispositivo final, que tienen como principal objetivo proporcionar al usuario final una representación de los datos de forma visual, legible y práctica. Cuando se habla de IoT enfocado a la agricultura, no solo se refiere a dispositivos móviles como tabletas o drones, ya que este concepto implica una interconexión de sistemas de comunicación agrícola compatibles y colaborativos que comparten información (sensores, cámaras, lámparas, estaciones meteorológicas., etc.).(Montoya et al., 2017).

## **2.6 PROCESAMIENTO DE IMÁGENES**

En la actualidad una herramienta que se utiliza para el control de plagas son sistemas basados en el procesamiento de imágenes.

El procesamiento de imágenes tiene como finalidad cambiar o mejorar el aspecto de las imágenes para resaltar en ellas ciertos detalles que a simple vista no se los puede observar. Una imagen puede ser generada de distintas maneras como, por ejemplo, electrónicamente, fotográficamente o por monitores de televisión. El procesamiento de imágenes se puede realizar por métodos digitales en una computadora o por medio de métodos ópticos.(Diana, Estefanía, 2017).

## **2.7 IMPORTANCIA DE UN ALGORITMO**

Una de las principales herramientas para el procesamiento de imágenes son los algoritmos computacionales.



El algoritmo es una técnica que permite resolver problemas o casos de estudio, ya que realiza secuencias repetitivas y detalladas que son fundamentales para llegar a una solución, el diseño de estos algoritmos requiere muchos conocimientos de programación y creatividad. Esto combinado con una adaptación a la arquitectura de una computadora que contribuye al correcto funcionamiento del algoritmo.

El algoritmo para el reconocimiento de objetos, personas y animales por medio procesamiento digital de imágenes usando cualquier tipo de software en tu ordenador, permite interactuar con ellos y definen los procesos que aportan resultados útiles, de esta forma partidos a los recursos hardware del sistema informático.(Guillermo Romero-Macas, 2020)

Para el desarrollo de los sistemas de detección de plagas existen varios softwares y lenguajes de programación.

## **2.8 MATLAB**

MATLAB son las siglas de "MATRIX LABORATORY". Es un programa para realizar operaciones numéricas utilizando vectores y matrices, por lo que también puede trabajar con números escalares (reales y complejas), con cadenas y con otras estructuras de información complejas.

Matlab es un lenguaje poderoso para la informática técnica, es tanto un entorno como un lenguaje de programación. Uno de los principales beneficios al utilizar Matlab es que permite construir herramientas que pueden ser reutilizadas. Además, se puede crear funciones y programas desde cero (llamados archivos M) en código Matlab, y se puede agruparlos en la Caja de herramientas (también conocida como librerías): un conjunto de archivos M especializados para trabajar en clases específicas de problemas.(Matlab, 2006).

### **2.8.1 Reconocimiento de imágenes con MATLAB**

Para el reconocimiento de imágenes Matlab cuenta con las siguientes opciones:

**Aplicaciones de etiquetado de imágenes:** Para obtener datos concretos y preprocesados se deben garantizar con éxito el reconocimiento de imágenes. Con las diferentes aplicaciones o herramientas proporcionados por el software MATLAB que puede controlar el proceso de etiquetado de imágenes.

**Mediante algoritmos de aprendizaje profundo y aprendizaje automático:** El software MATLAB ofrece una gran variedad de posibilidades y técnicas de aprendizajes profundo y automáticos que aceleran tareas y proporcionan mayor eficacia al momento de detectar imágenes. (Mathworks, 2020)

## **2.9 TÉCNICAS DE PROCESAMIENTO DE IMÁGENES PARA EL RECONOCIMIENTO DE IMÁGENES.**

El reconocimiento de imágenes puede acelerar las tareas pesadas y procesar imágenes con mayor rapidez o precisión que el escaneo manual de imágenes. El reconocimiento de imágenes es una tecnología importante en muchas aplicaciones y es el principal impulsor de las aplicaciones de aprendizaje profundo como: Inspección de visual, Clasificación de imágenes, Conducción automatizada y Robótica.(Mathworks, 2020)

### **2.9.1 Reconocimiento de imágenes mediante Machine Learning**

Los algoritmos de Machine Learning utilizan métodos computacionales para "adquirir" información directamente de los datos, sin depender de una ecuación predefinida como plantilla. Los algoritmos mejoran su rendimiento de forma adaptativa a medida que aumenta el número de muestras disponibles para el entrenamiento.

Machine Learning utiliza dos tipos de técnicas para su aprendizaje: aprendizaje supervisado, que entrena una muestra en datos de entrada y salida conocidos para que pueda predecir la salida futura, y aprendizaje no supervisado, que ayuda a encontrar patrones estructurales ocultos o intrínsecos en los datos de entrada. (MathWorks, 2021a)

### **2.9.2 Reconocimiento de imágenes mediante técnicas tradicionales de procesamiento de imágenes**

Existen varias técnicas tradicionales para el procesamiento de imágenes que se basan en el reconocimiento de los píxeles proporcionados por las imágenes adquiridas.

**Reconocimiento de imágenes enfocado en color:** las imágenes tienen por generalidad o característica la saturación, el tono y el valor o también conocido como el rojo, verde y azul (RGB) que pueden proporcionar información necesaria de la imagen.

**Plantillas :** es una técnica pequeña que se basa en realizar un pequeño corte en la imagen para poder localizar regiones que coinciden con otras imágenes.

**Segmentación de imágenes :** esta técnica utiliza todas las propiedades que un objeto o imagen proporciona de manera simple o compuesta esto puede ser el tamaño, forma o color. (Mathworks, 2020)

### **2.9.3 Reconocimiento de imágenes mediante Deep Learning**

El aprendizaje profundo (Deep Learning) proporciona un marco para diseñar e implementar redes neuronales profundas utilizando algoritmos, modelos y aplicaciones previamente entrenados. Puede utilizar redes neuronales complejas (ConvNets y CNN) y redes de memoria a corto y largo plazo (LSTM) para realizar la clasificación y regresión de imágenes, series de tiempo y datos de texto. Puede crear arquitecturas de red como las redes adversas comunes (GAN) y las redes siamesas mediante discriminación automática, rondas de entrenamiento personalizadas y pesos comunes. Con Deep Network Designer, puede diseñar, analizar y entrenar su red. La aplicación Experiment Manager lo ayuda a administrar múltiples experimentos de aprendizaje profundo, realizar un seguimiento de los parámetros de entrenamiento, analizar los resultados y comparar el código de diferentes pruebas. Puede ver la activación de la clase y realizar un seguimiento gráfico del progreso del ejercicio. (Hudson et al., 2021).

El aprendizaje profundo es particularmente adecuado para aplicaciones de reconocimiento complejas como el reconocimiento facial, la traducción de texto y el reconocimiento de voz. También es una tecnología central utilizada en tareas y sistemas avanzados de asistencia al conductor, incluido el mapeo de carriles y el reconocimiento de señales de tráfico. (MATLAB & Simulink, 2021).

### **2.9.4 Transferencia de Aprendizaje**

La transferencia de aprendizaje o Transfer Learning, es un método de Deep Learning en el que un modelo entrenado para una tarea se utiliza como punto de partida para otro modelo que realiza otra tarea similar. Actualizar y reentrenar una red de transferencia de aprendizaje suele ser mucho más fácil y rápido que entrenarla desde cero. Esta metodología se usa comúnmente en aplicaciones de detección de objetos, reconocimiento de imágenes y reconocimiento de voz, entre otras. (Mathworks, 2021c)

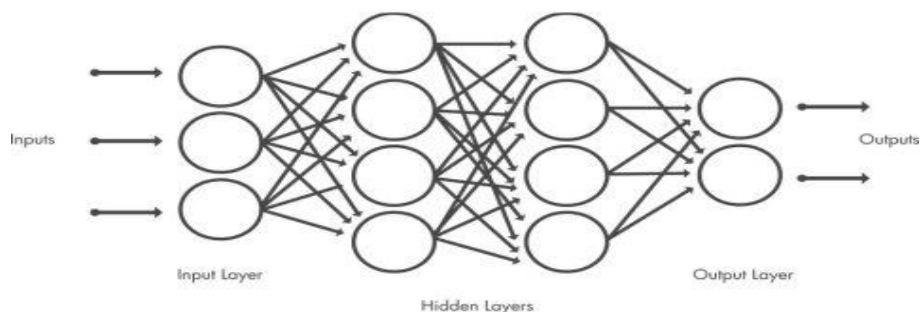
Transfer Learning es una técnica de uso común por las siguientes razones:

- Le permite entrenar modelos con menor cantidad de datos mediante la reutilización de modelos ya existentes que han sido entrenados anteriormente con un conjunto grande de datos.
- Ayuda a reducir el tiempo para su entrenamiento. Los pesos no se aprenden desde cero, porque el modelo previamente entrenado aprende los pesos de lecciones anteriores.
- Aprovecha las arquitecturas desarrolladas por la comunidad de investigación de aprendizaje profundo, incluidas las arquitecturas de uso común como GoogLeNet y ResNet.

### 2.9.5 Red neuronal Convolutacional.

Una red neuronal convolutacional CNN es un método de diseño de red para deep learning que realiza su aprendizaje directamente cuando se extraen los datos. Estas redes son muy utilizadas para determinar ciertos patrones en imágenes en objetos, caras y escenas. También resultan eficaces para archivar y archivar datos sin imágenes, tales como datos de audio, series temporales y señales. El aprendizaje se lleva a cabo en capa superior de neuronas donde la salida de neuronas similares es aplicada o incide en lugares específicos. Actualmente, los sistemas de detección de patrones basados en redes neuronales convolucionales son los que tienen mejor desempeño. Las redes convolucionales comúnmente están organizadas en 2 tipos de capas (las capas convolucionales y las capas de submuestreo). Como se muestra en la figura 2.4 cada capa tiene un ordenamiento topográfico y está asociada con una localización bidimensional respecto a una trayectoria receptiva. (Tamura, 2019).

Figura 2.4 Red Neuronal Convolutacional



Fuente: (MathWorks, 2021b)

### **2.9.6 Red GoogLeNet**

GoogLeNet es una red neuronal convolucional ya existente que cuenta con 22 capas de profundidad. Puede cargar una versión preentrenada de la red en los conjuntos de datos ImageNet o Places365. La red entrenada en ImageNet clasifica las imágenes en 1000 categorías de objetos, como teclado, mouse y muchos animales. La red entrenada en Places365 es similar a la red entrenada en ImageNet, pero clasifica las imágenes en 365 categorías de lugares diferentes, como campo, parque, pista y vestíbulo. Estas redes han aprendido diferentes representaciones de características para una amplia gama de imágenes. Las redes preentrenadas tienen un tamaño de entrada de imagen de 224 por 224.

Puede volver a entrenar la red neuronal convolucional GoogLeNet para realizar una nueva tarea mediante el aprendizaje por transferencia. En el aprendizaje por transferencia, el método más común es utilizar redes previamente entrenadas en el conjunto de datos de ImageNet. Si la nueva tarea es similar a la clasificación de escenas, el uso de la cuadrícula formada por Places365 puede proporcionar una mayor precisión.(Mathworks, 2021a)

### **2.9.7 Resnet\_18.**

Es una red neuronal convolucional que posee 18 capas de profundidad. Tiene la opción cargar una versión preentrenada de la red, dispone de un millón de imágenes de la base de datos. Con la red preentrenada, logra clasificar imágenes en 1000 categorías de objetos, como plantas y muchos animales. La red dispone un tamaño de entrada de imagen de 224x224.(Mathworks, 2021b)

## CAPITULO 3

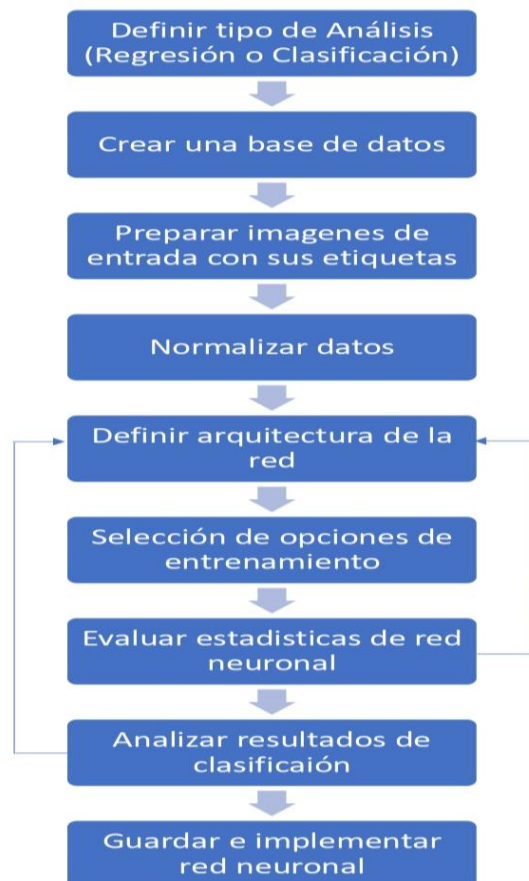
### DESARROLLO DEL ALGORITMO DE PROCESAMIENTO DE IMÁGENES.

#### 3.1 DESCRIPCIÓN DE LOS REQUERIMIENTOS EN LAS DISTINTAS ETAPAS PARA EL DE DESARROLLO DEL ALGORITMO.

El objetivo de este proyecto es el diseño de un detector basado en un algoritmo mediante el procesamiento de imágenes para la identificación del hongo Moniliasis el cual es desarrollado en el software Matlab 2021b.

A continuación, se presenta el cuadro de procedimiento para la implementación del algoritmo (Figura 3.1) que describe una comprensión a simple vista de la lógica de programación. Sin embargo, el diagrama no es lo suficientemente detallado pues es necesario describir la importancia de cada bloque en el código general.

Figura 3.1 Cuadro de procedimiento para la implementación del algoritmo



Elaborado por: Jefferson Faconda y Edison Suquillo

### 3.1.1 Definir tipo de análisis

En esta primera etapa se define el tipo de análisis que se utiliza para el desarrollo del algoritmo. Se utiliza el Análisis de Clasificación debido a que se necesita un resultado de etiqueta discreta, esto quiere decir que la respuesta está fundamentada en un conjunto finito de resultados que en este caso son las 3 fases del hongo Moniliasis; la figura 3.2 se puede observar la fase 1 cuando en la mazorca empieza una leve mancha negra con una pequeña porción de polvo blanco que se desarrolla hasta alcanzar el 30% del total de la mazorca, la figura 3.3 representa la fase 2 se caracteriza por el crecimiento de la mancha negra y con el aumento del polvo blanco debido al hongo, la cual representa una infección mayor al 30% y menor al 90% del total de la mazorca, y la figura 3.4 representa la fase 3 donde se visualiza que el hongo ha afectado un porcentaje mayor al 90% de su total coloración, haciéndolo de color negro con abundante presencia de polvo blanco.

Figura 3.2 Fase 1



Elaborado por: Jefferson Faconda y Edison Suquillo

Figura 3.3 Fase 2



Elaborado por: Jefferson Faconda y Edison Suquillo

Figura 3.4 Fase 3



Elaborado por: Jefferson Faconda y Edison Suquillo

### 3.1.2 Crear una base de datos

En esta etapa es necesario obtener una gran cantidad de fotografías del cacao para crear una base de datos para el entrenamiento del algoritmo. Dichas fotografías se obtienen manualmente en un ambiente no controlado: no se encuentran centradas, tienen diferentes fondos, la intensidad de luz varía en cada fotografía, etc. Las cuales son obtenidas con un dispositivo celular Xiaomi Redmi Note 8 Pro el cual cuenta con un modo Pro para fotografías que permite configurar la apertura del diafragma, tiempo de exposición y Sensibilidad del sensor (ISO), adicional se utiliza la configuración 64 MegaPixel que dispone el dispositivo.

Como se describe anteriormente el algoritmo debe ser capaz de reconocer las diferentes etapas del Hongo Moniliasis, por lo cual es necesario crear una base de datos con varias etiquetas como se las describe a continuación:

#### **Etiqueta 1**

Esta base de datos se llama “Bueno”, la cual está conformada por 331 fotografías de la mazorca de cacao en buen estado.

#### **Etiqueta 2**

Esta base de datos se llama “Fase\_1”, la cual está conformada por 433 fotografías de la mazorca con una infección de la Moniliasis menor al 30% de su total.

#### **Etiqueta 3**

Esta base de datos se llama “Fase\_2”, la cual está conformada por 578 fotografías de la mazorca con una infección de la Moniliasis mayor al 30% y menor al 90% de su total.



## Etiqueta 4

Esta base de datos se llama “Fase\_3”, la cual está conformada por 547 fotografías de la mazorca con una infección de la Moniliasis mayor al 90% de su total.

Figura 3.4 Almacenamiento de imágenes



Elaborado por: Jefferson Faconda y Edison Suquillo

### 3.1.3 Preparar imágenes de entrada con sus etiquetas

El modelo RGB es uno de los sistemas informáticos más utilizados para crear y reproducir colores en pantalla. Depende de la llamada "síntesis adicional", donde se agrega la intensidad de la luz relacionada con el rojo, el verde y el azul para ser sumadas entre sí y obtener diferentes colores, incluido el blanco y negro. (Aguirre Dobernack, 2018).

Una vez obtenido las imágenes RGB mediante el dispositivo móvil anteriormente mencionado es necesario un procesamiento de las fotografías con la finalidad de corregir problemas, como las variaciones en intensidad producidas por el ruido o por la deficiencia de la iluminación, además se puede utilizar la opción de Matlab “imageDataAugmenter” para aumentar la base datos de entrenamiento, esta opción permite configurar un conjunto de opciones procesamiento como es la escala, la rotación y la reflexión.

### 3.1.4 Normalizar Datos

Para el algoritmo en Matlab la normalización de datos se realiza como un proceso de reducción de rangos de datos de la entrada que tienen efecto sobre la red neuronal para evitar que tome decisiones incorrectas al momento del procesamiento. Este proceso de normalización permite también que las redes neuronales no instalables sean adaptables, lo que es crucial para los siguientes procesos. (Moreno, 2019)

### **3.1.5 Definir Arquitectura de la Red**

Una vez normalizado los datos lo siguiente es escoger el tipo de arquitectura de red con la que se desarrolla el algoritmo, en este caso se opta por las Redes Neuronales Convolucionales (CNN) debido a que es un tipo de red diseñado para Deep Learning que tiene la capacidad de aprender los datos directamente sin tener que extraer las características de las imágenes de manera manual, también permite reentrenar redes preexistentes.

La arquitectura de CNN puede variar según los tipos y la cantidad de capas involucradas. Los tipos y el número de capas incluidas dependen de la aplicación o los datos específicos. Eso quiere decir, si tiene respuestas categóricas, debe tener una capa y una función de clasificador, mientras que, si su respuesta es continua, debe tener una capa de regresión al final de la red. Para una red pequeña basta con solo una o dos capas de complejidad en cambio para red más grande es posible que necesite datos más complejos con millones de imágenes en color por lo cual se tiene que crear una red con múltiples capas convolucionales.

### **3.1.6 Selección de Opciones de Entrenamiento de la Red Neuronal**

Una vez definida la arquitectura de la red se procede a colocar las diferentes opciones de entrenamiento que tiene la red neural, con los siguientes ítems a considerar.

### **3.1.7 Evaluar Estadísticas de Red Neuronal**

Con la red neuronal convolucional entrenada se puede observar el porcentaje de precisión que tiene la red neuronal convolucional en caso de ser un porcentaje alto se procede con la siguiente etapa caso contrario se analiza nuevamente las etapas anteriores.

### **3.1.8 Analizar Resultados de Clasificación**

Para el desarrollo de esta etapa es de gran ayuda el análisis de la Matriz de Confusión, esta matriz presenta las predicciones realizadas y los resultados correctos en forma de tabla con estos datos se puede medir el desempeño del algoritmo para la clasificación de datos, en caso de tener un desempeño bajo se tiene que analizar las etapas anteriores para mejorar el desempeño de clasificación del algoritmo.

### 3.1.9 Guardar e implementar Red Neuronal

Una vez que el algoritmo cumple con los parámetros de clasificación de las distintas etapas del hongo Moniliasis se procede a guardar e implementar el algoritmo a través de la interfaz APP DESIGNER de Matlab.

## 3.2 ALGORITMO Y LÓGICA DE PROGRAMACIÓN

### 3.2.1 Selección de Imágenes

Una vez creada la base de datos en el computador se procede a llamar dicha base datos al algoritmo con la opción “path”, además con la ayuda del comando “pwd” en el Command Windows permite verificar la dirección la carpeta actual.

Para la creación de la base de datos en el algoritmo se utiliza ImageDatastore en el cual se puede especificar las siguientes propiedades: path (dirección de la carpeta de imagenes), IncludeSubfolders (para que incluya las subcarpetas), LabelSource y foldernames (el nombre de las carpetas es la etiqueta de las imágenes) como se observa en la figura 3.5.

Figura 3.5 Creación del imds

```
%Creación de base de datos en la que el nombre de las subcarpetas es la
%etiqueta de la imagen
imds = imageDatastore(path, 'IncludeSubfolders', true, 'LabelSource', 'foldernames')
```

Elaborado por: Jefferson Faconda y Edison Suquillo

Si se desea hacer un conteo de imágenes que se encuentran en la base de datos creada se puede utilizar la opción “numel” la cual permite observar el número de elementos en un vector y si se desea un conteo por cada etiqueta el comando “countEachLabel” muestra el valor de vector por etiqueta como se observar en la figura 3.6.

Figura 3.6 conteo de imágenes

```
%% Información general de la base de datos
%Numero total de imagenes
total = numel(imds.Files)
%Numero de imagenes por etiquetas
porClase = countEachLabel(imds)
```

Elaborado por: Jefferson Faconda y Edison Suquillo

Una vez creada la base de datos (imds) en Matlab se procede a dividir el archivo de imágenes en dos nuevos almacenes de datos. El nuevo almacén de datos imds\_Test (imágenes de las que va aprender la red) contiene los primeros archivos p de cada etiqueta e imds\_Validation

(imágenes auxiliares que sirven para validar la red) contiene los archivos restantes de cada etiqueta. El valor de p puede entre 0 y 1 que indica el porcentaje de archivos de cada etiqueta a asignar a imds\_Test como se puede observar en la figura 3.7.

Figura 3.7 Dividir Datos

```
%% Dividir datos|
[imds_Test,imds_Validation] = splitEachLabel(imds,0.75)
```

Elaborado por: Jefferson Faconda y Edison Suquillo

### 3.2.2 Aumento de datos de imágenes para entrenamiento de la red

Una vez creada la base de datos con las 4 etiquetas se procede aumentar imágenes para que la red neural tenga un mejor rendimiento mediante el comando “imageDataAugmenter” que permite un preprocesamiento para el aumento de imágenes como cambio de escala, rotación, reflexión y etc. Con los diferentes filtros que utiliza el comando mencionado, en la figura 3.8 se observa que el comando rota imágenes por ángulos aleatorios en el rango [0, 360] grados y cambia el tamaño de las imágenes por factores de escala aleatorios en el rango [0.5, 1].

Figura 3.8 Aumento de Imágenes

```
%Transformaciones a las imagenes (Rotación, Traslación, Escalado, etc)
augmenter = imageDataAugmenter( ...
    'RandRotation',[0 360], ...
    'RandScale',[0.5 1])
```

Elaborado por: Jefferson Faconda y Edison Suquillo

La red neural GoogleNet utiliza un tamaño de imágenes de 224x224x3 como se observa en la figura 3.9 donde “l=3” en la red neuronal ingresan imágenes a color.

Figura 3.9 Tamaño de las imágenes

```
%Cambio de tamaño de las imagenes para ingresar a la red
r=224;
c=224;
l=3;
```

Elaborado por: Jefferson Faconda y Edison Suquillo

Posteriormente con el comando “augmentedImageDatastore” almacena las imágenes aumentadas que transforma en datos que sirve para realizar la validación de la red neural, con el tamaño de la entrada que necesita la red, las imágenes de las que aprende la red neuronal, también las imágenes auxiliares que sirven para validar las imágenes almacenadas y el aumento

de imágenes preprocesadas como se puede visualizar en la figura 3.10, por último, se realiza una suma de todos archivos existentes.

Figura 3.10 Creación de la base de dato con el aumento de imágenes

```
%Crear base de datos con aumento de imagenes
auds_test = augmentedImageDatastore([r c l],imds_Test,'DataAugmentation',augmenter)
auds_validation = augmentedImageDatastore([r c l],imds_Validation,'DataAugmentation',augmenter)

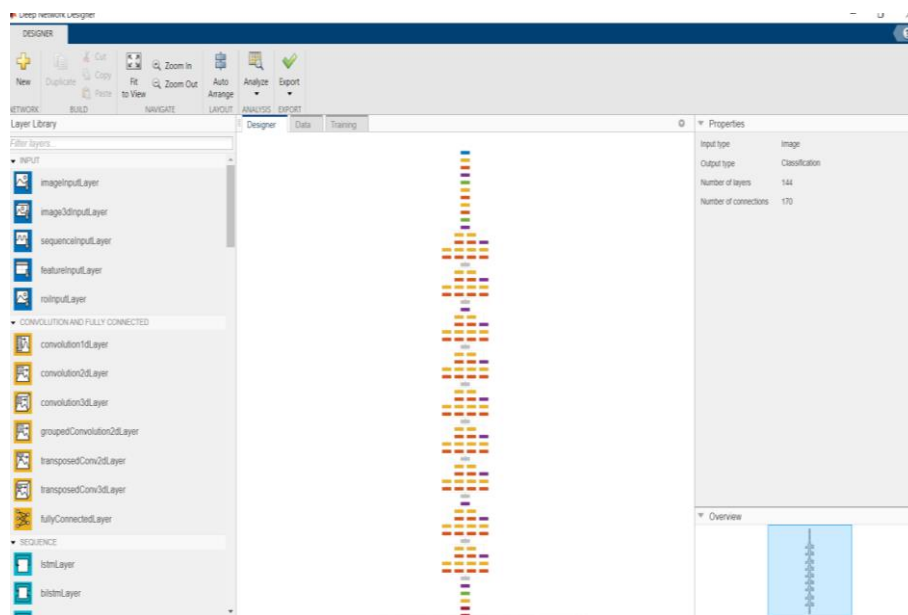
%Numero total de imagenes modificadas (No cambia el numero solo se aplican
%las transformaciones
total2 = auds_test.NumObservations+auds_validation.NumObservations
```

Elaborado por: Jefferson Faconda y Edison Suquillo

### 3.2.3 Creación de la Red Neuronal Convolutiva mediante transferencia de aprendizaje GoogLeNet

Para la creación de la red neuronal convolutiva en Matlab se usa la herramienta Deep Network Designer en la cual se selecciona la red preexistente para realizar las modificaciones necesarias, en este caso se ocupa la red ya existente GoogLeNet como se observa en la figura 3.11.

Figura 3.11 Red existente GoogLeNet



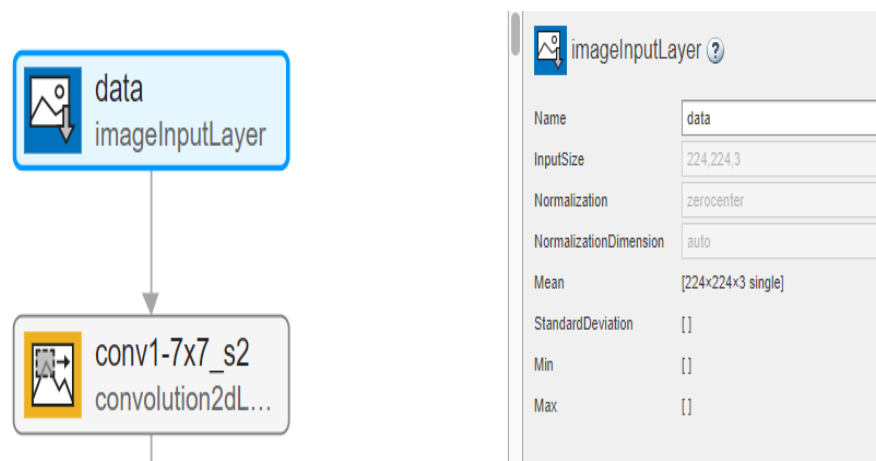
Elaborado por: Jefferson Faconda y Edison Suquillo

Una vez seleccionada la red GoogLeNet se procede a realizar los cambios de las capas necesarios de la red entrenada para que la red realice la transferencia de aprendizaje y tenga la capacidad de reconocer el hongo Moniliasis en sus diferentes etapas.

En la opción Designer se visualizan las 144 capas las cuales se describen las más importantes y las que deben ser modificadas a continuación:

La primera capa describe el tamaño en el cual deben ingresar las imágenes para su entrenamiento como se aprecia en la figura 3.12.

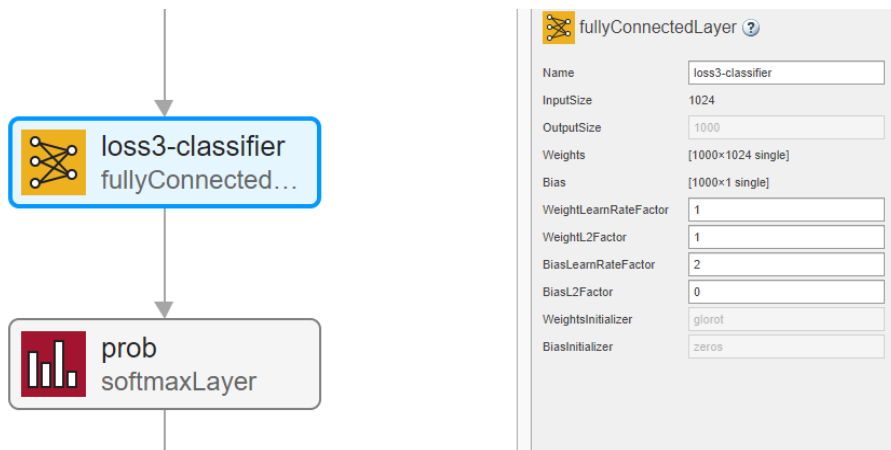
Figura 3.12 Primera Capa de la red GoogLeNet



Elaborado por: Jefferson Faconda y Edison Suquillo

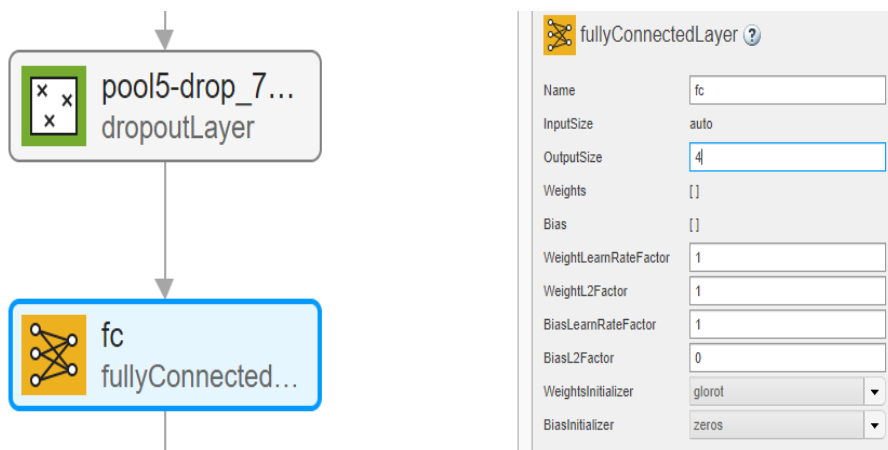
La capa “fullyConnectedLayer” es modificada debido a que esta es la última capa capaz de aprender las nuevas instrucciones de la red para realizar la transferencia de aprendizaje, además en esta capa se define el número de etiquetas con las cuales se trabaja, en la figura 3.13 es como GoogLeNet está entrenada a trabajar con 1000 etiquetas por defecto y en la figura 3.14 se puede observar que se reemplaza esa capa por una de solo 4 etiquetas (bueno, fase\_1, fase\_2 y fase\_3).

Figura 3.13 Capa “fullyConnectedLayer” por defecto de la red GoogLeNet



Elaborado por: Jefferson Faconda y Edison Suquillo

Figura 3.14 Capa “fullyConnectedLayer” modificada de la red GoogLeNet



Elaborado por: Jefferson Faconda y Edison Suquillo

La capa “classificationLayer” debe ser modificada porque esta capa es la encargada de la clasificación final del nuevo conjunto de datos ingresado después de ser procesados por las capas anteriores. La capa de clasificación viene por defecto con las 1000 etiquetas de GoogLeNet en cambio en la figura 3.15 se puede observar el cambio de la capa que viene por defecto por una capa nueva que clasifica las nuevas etiquetas.

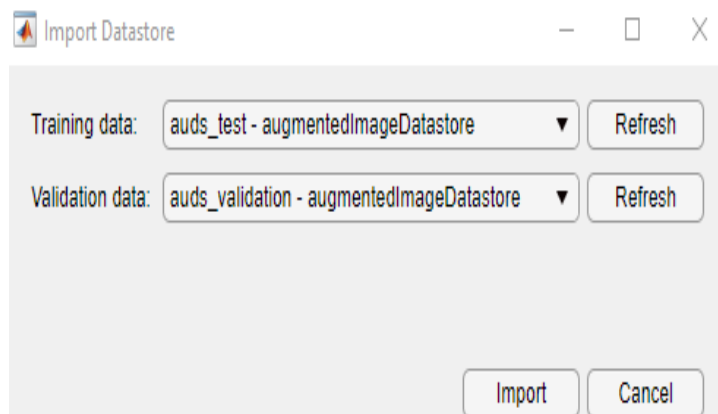
Figura 3.15 Capa “classificationLayer” modificada de la red GoogLeNet



Elaborado por: Jefferson Faconda y Edison Suquillo

Se procede a la configuración de los Datos de la red neuronal convolucional donde se llama desde el “workspace” al almacenamiento de datos aumentados anteriormente que son “auds\_test” y “auds\_validation” como se observa en la figura 3.16.

Figura 3.16 Importación de la base de datos “auds\_test” y “auds\_validation”

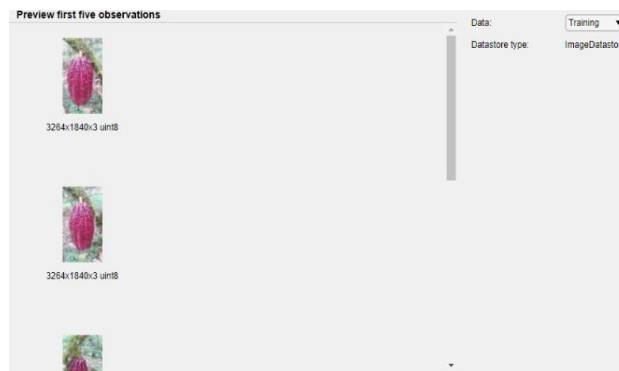


Elaborado por: Jefferson Faconda y Edison Suquillo



En la figura 3.17 se puede observar el almacenamiento de datos que son cargados a la CNN.

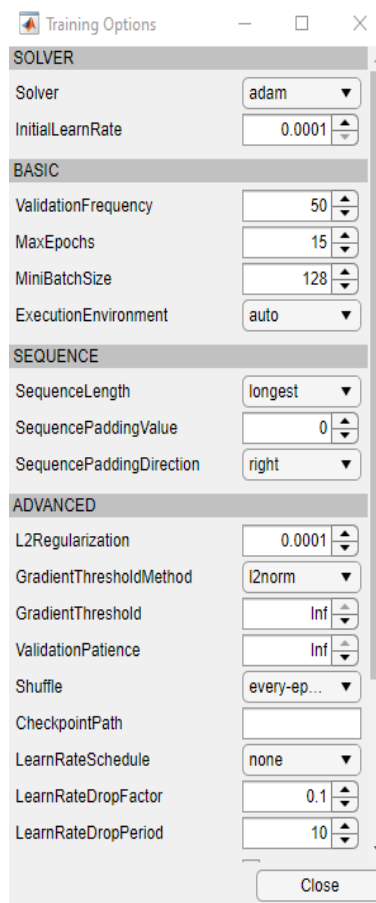
Figura 3.17 Visualización de la base de datos “auds\_test”



Elaborado por: Jefferson Faconda y Edison Suquillo

Al tener cargada la base de datos lo siguientes es configurar los parámetros de entrenamiento los cuales se los deja por defecto del Toolbox de redes neuronales de Matlab como se observa en la figura 3.18.

Figura 3.18 Parámetros de entrenamiento para la CNN



Elaborado por: Jefferson Faconda y Edison Suquillo

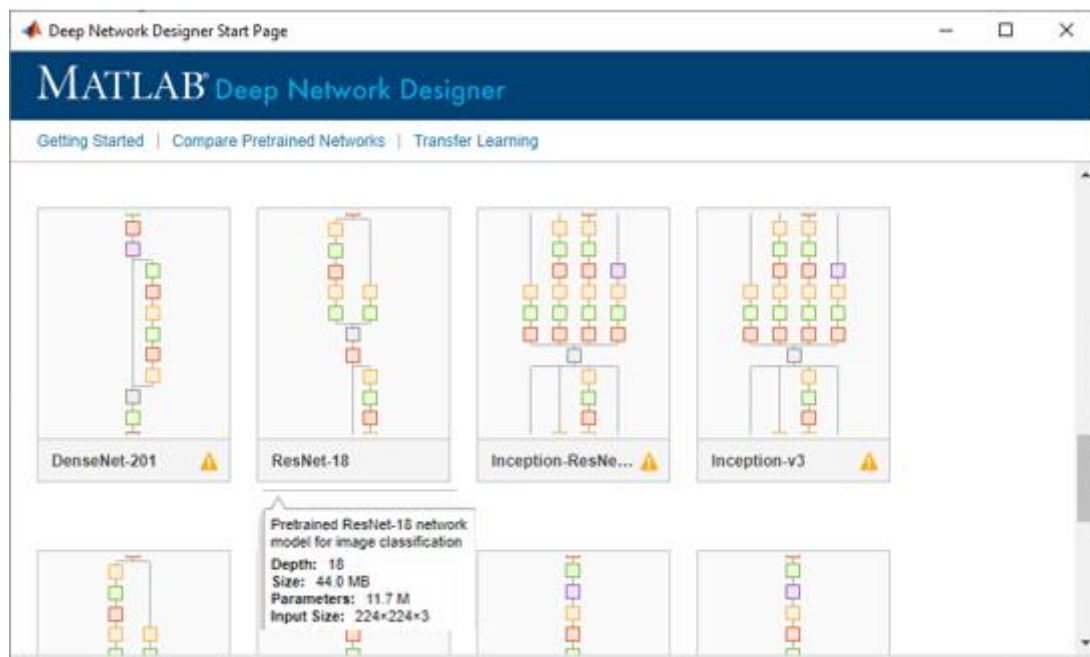
Después de configurar todos los parámetros anteriores se da clic en la opción “Train” el cual da inicio al entrenamiento de la CNN que varía el tiempo dependiendo de las características del computador que se utiliza.

Al finalizar el entrenamiento de la red se procede a exportar mediante la opción workspace y para guardar la red (.mat) se escribe el siguiente comando en el coman windows de Matlab “save(nombre\_de\_la\_red.mat)”.

### 3.2.4 Creación de la Red Neuronal Convolutacional mediante transferencia de aprendizaje ResNet-18

Para la creación de la red neuronal convolutacional con la red ya existente ResNet-18 se realiza el mismo procedimiento y se configura los mismos parámetros que la GoogLeNet solo se tiene que escoger la opción de ResNet-18 en vez de GoogLeNet en la opción Deep Network Designer como se visualiza en la figura 3.19.

Figura 3.19 CNN con ResNet\_18



Elaborado por: Jefferson Faconda y Edison Suquillo

### 3.2.5 Creación de la Red Neuronal Convolutacional Básica

Una vez ya obtenida la base datos se tiene que crear una CNN básica, para ello se inicia creando las capas de la red como observa en la figura 3.20.

A continuación, se describe capa creada para la CNN:

**Capa de Entrada de la Imagen (“imageInputLayer”).** – En esta capa se tiene que especificar el tamaño de entrada de a la red.

En las CNN no es necesario colocar una línea de comando para normalizar porque el software Matlab normaliza los datos de manera automática.

**Capa de Convolución (“Convolution2dLayer”).** – Este tipo de capa convolucional 2D aplica filtros convolucionales deslizantes a la entrada 2D. La capa hace girar la entrada moviendo los filtros a lo largo de la entrada vertical y horizontalmente. En esta línea de código se coloca el tamaño del filtro (5x5 pixeles que recorren por toda la imagen) y números de filtros (20 filtros) que se desean colocar.

**Capa de Activación o Rectificación (“reluLayer”).** – La capa de convolución por lo general es seguido de una capa de activación no lineal, Matlab al normalizar los datos presenta las características en valores enteros, esta capa lo que realiza es, al encontrar algún valor negativo lo establece como cero.

**Capa de Reducción (“maxPooling2dLayer”).** – Esta capa es la encargada de la agrupación máxima que realiza un muestreo descendente dividiendo la entrada en varias regiones de agrupación rectangular y calcula el máximo de cada región. De esta manera la agrupación reduce la complejidad de la red y crea una nueva red general.

Esta capa está configurada como “maxPooling2dLayer(2,'Stride',2)” donde se especifica Stride como escalar para poder usar el mismo valor tanto para vertical como horizontal, en este caso se especifica el valor de 2, este valor como un vector de dos enteros positivos [a b], donde “a” es el tamaño de paso vertical y “b” es el tamaño de paso horizontal.

**Capa totalmente conectada (“fullyConnectedLayer”).** – Esta capa realiza una multiplicación entre la entrada y una matriz de peso, después de dicha multiplicación agrega un vector de sesgo. El tamaño de salida para esta capa es el número de clases o etiquetas para la clasificación en este caso serian 4 (Bueno, fase\_1, fase\_2 y fase\_3).

**Capa de Activación (“softmaxLayer”).** - Esta capa ayuda a convertir los valores de cada clase de salida (Bueno, fase\_1, fase\_2 y fase\_3) en puntuaciones normalizadas por lo cual puede

interpretar cada valor como la probabilidad de que la imagen de entrada pertenezca a una clase de salida.

**Capa de clasificación (“classificationLayer”).** – La capa de clasificación de salida devuelve el nombre de la clase más probable de acuerdo con la capa anterior.

En las **opciones de entrenamiento** se aprecia el tipo que optimizador 'adam' que es una opción de entrenamiento de las redes neuronales de aprendizaje profundo.

'MaxEpochs'15' (número de épocas), donde una época es el paso completo del algoritmo de entrenamiento sobre todo el conjunto de entrenamiento.

'Shuffle','every-epoch' cambia los datos antes de época y los de validación.

'Plots','training-progress' visualiza las gráficas de resultado de entrenamiento.

'Verbose',false, se coloca para ocultar progresión de entrenamiento en comand window.

'InitialLearnRate' la tasa de aprendizaje inicial manejada para el entrenamiento, se utiliza un escalar positivo, se coloca 0.0001 para que el resultado sea más optimo.

'ValidationData' se utiliza este comando para usar las imágenes de validación.

Figura 3.20 CNN Básica

<b>%% Creación de red neuronal</b>	
layers = [ ...	
imageInputLayer([r c l])	%Capa de entrada de imagen
convolution2dLayer(5,20)	%Capa de convolución1
reluLayer	%Capa de activacion1
maxPooling2dLayer(2,'Stride',2)	%Capa de reducción
fullyConnectedLayer(3)	%Capa totalmente conectada
softmaxLayer	%Capa de activación 2
classificationLayer];	%Capa de clasificación
<b>%% Opciones de entrenamiento</b>	
opts = trainingOptions('adam', ...	%Tipo de optimizador
'MaxEpochs',15, ...	%Numero máximo de epocas
'Shuffle','every-epoch', ...	%Variar despues de cada epoca
'Plots','training-progress', ...	%Visualizar graficas de resultado de entrenamiento
'Verbose',false, ...	%Oclutar progresion de entrenamiento en comand window
'InitialLearnRate', 0.0001, ...	%Tasa de aprendizaje
'ValidationData',audsValidation);	%Usar imagenes de validación

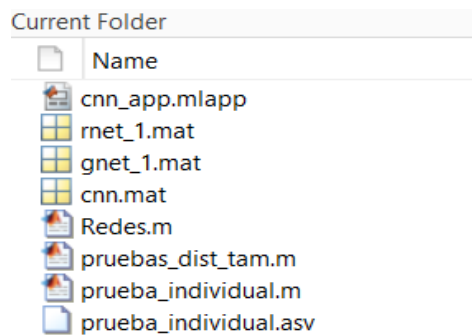
Elaborado por: Jefferson Faonda y Edison Suquillo

### 3.3 COMPILACIÓN Y DESARROLLO.

#### 3.3.1 Verificación de la red

La red al estar guardada en el workspace puede ser llamada en cualquier instante como se observa en la figura 3.21.

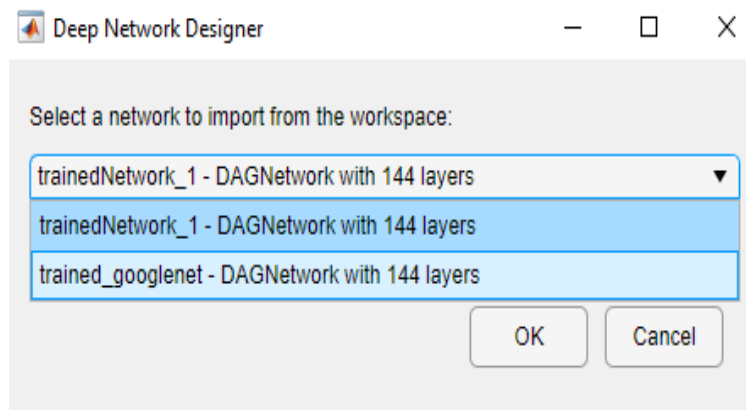
Figura 3.21 Redes guardadas



Elaborado por: Jefferson Faconda y Edison Suquillo

Para verificar que la transferencia de aprendizaje es la correcta se procede a importar el archivo “.mat” desde la red GoogLeNet como se muestra en la figura 3.22.

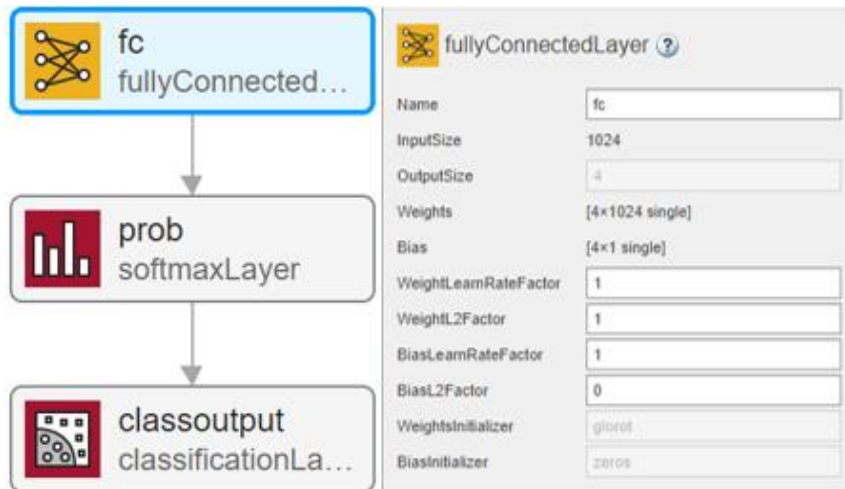
Figura 3.22 Importar red entrenada



Elaborado por: Jefferson Faconda y Edison Suquillo

Como se observar en la figura 3.23 el número de etiquetas son las 4 establecidas en el momento del entrenamiento.

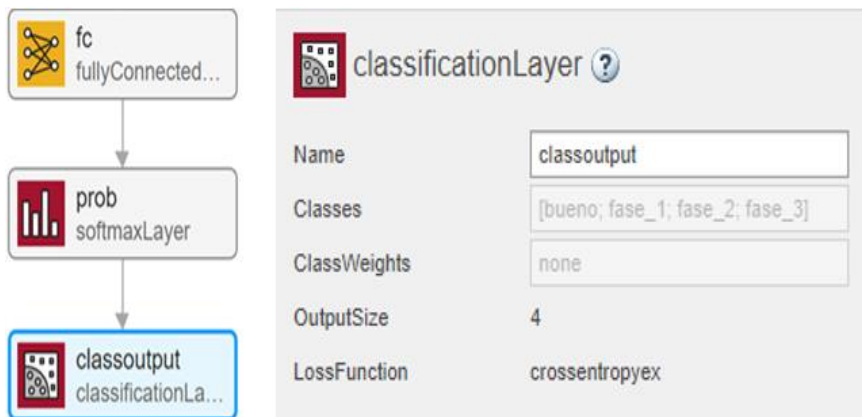
Figura 3.23 Número de etiquetas establecidas “4”



Elaborado por: Jefferson Faconda y Edison Suquillo

En la figura 3.24 se puede observar que el nombre de las etiquetas (bueno, fase\_1, fase\_2, fase\_3) se han detectado correctamente.

Figura 3.24 Nombre de las etiquetas establecidas



Elaborado por: Jefferson Faconda y Edison Suquillo

De igual manera para que la el sistema sea capaz de reconocer imágenes de cacao se crea una red neuronal para que cumpla esta condición, en caso de reconocer que es cacao ingresa al algoritmo de detección del hongo Moniliasis.

### 3.3.2 Tiempos de Procesamiento

El tiempo de procesamiento es el tiempo que tarda ejecutar el algoritmo. Matlab permite medir el tiempo transcurrido durante la ejecución de un determinado código, a través del comando “Tic” que funciona con el comando “Toc”. La función “Tic” registra el tiempo actual y la

función “Toc” usa el valor registrado para calcular el tiempo transcurrido. Estos comandos son útiles y resultan muy fáciles de implementar como se observa en la figura 3.25.

Figura 3.25 Tiempo de Procesamiento

```
tic  
predi = classify(net, Ima)  
tt=toc;  
t=[t, tt]
```

Elaborado por: Jefferson Faconda y Edison Suquillo

### 3.4 INTERFAZ GRÁFICA E IMPLEMENTACIÓN DE LA RED

Para implementar la red neuronal que clasifica los frutos de cacao, se desarrolla una aplicación con la herramienta “APP DESIGNER” de Matlab en la que se pueda cargar una imagen y se reconozca si posee el hongo.

En la sección 4.1, 4.2 y 4.3 se observa que la red más eficiente para trabajar con imágenes de alta calidad de entre las tres principales estudiadas es la arquitectura Googlenet. Por este motivo, en la interfaz se implementa el primer entrenamiento de esta red.

La interfaz gráfica consta de una pantalla en la que se encuentra un botón que al ser aplastado se abre una ventana que permite elegir una imagen del computador. Luego, mediante la red neuronal se clasifica la imagen y aparece en la pantalla si es cacao ingresa a la red neuronal de detección del hongo e indica la fase de desarrollo de moniliasis en la que se encuentra caso contrario aparece una etiqueta con el mensaje “NO ES CACAO”.

Para realizar el reconocimiento de un fruto se realiza siguiendo el procedimiento:

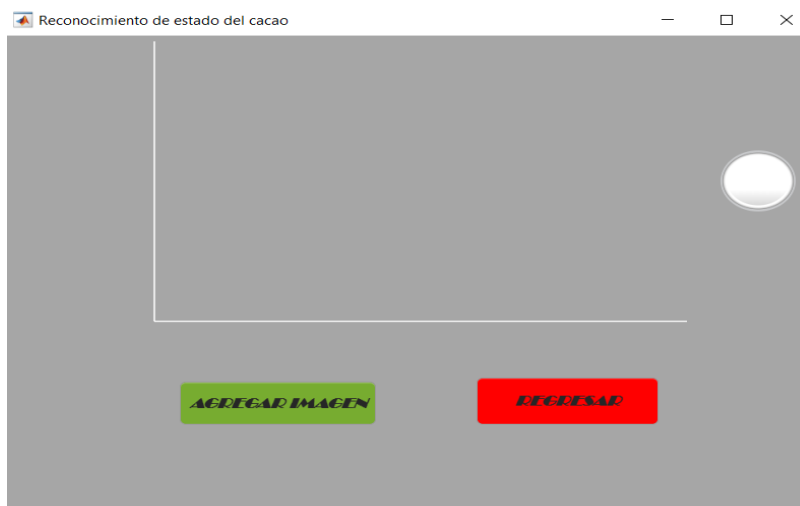
1. Ingresar a la Portada Presionar en Ingresar
2. Presionar el botón Agregar Imagen.
3. Navegar hasta el archivo que contenga el fruto a analizar y presione el botón abrir.
4. La imagen en cuestión se muestra en la ventana de la aplicación y debajo se indica la etapa de enfermedad que clasifica la red neuronal.

Figura 3.26 Portada



Elaborado por: Jefferson Faconda y Edison Suquillo

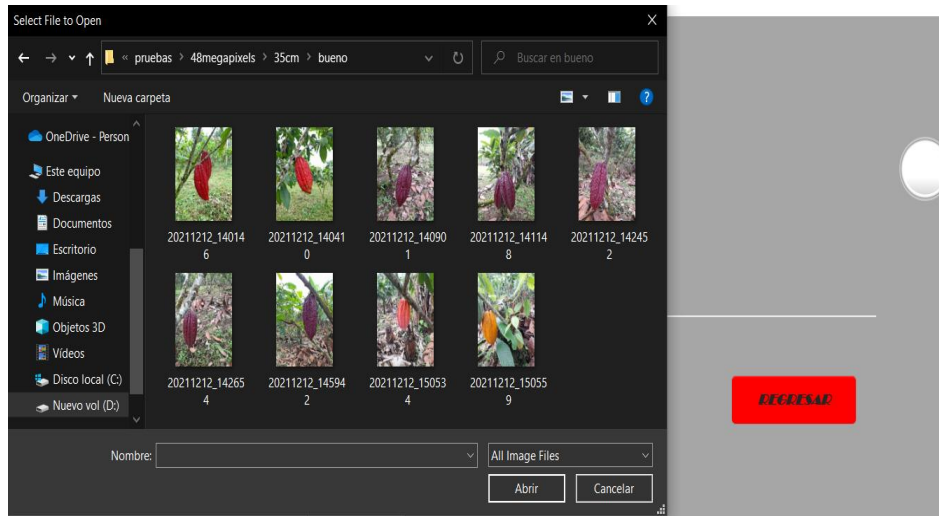
Figura 3.27 Ventana de selección.



Elaborado por: Jefferson Faconda y Edison Suquillo



Figura 3.28 Selección de la imagen.



Elaborado por: Jefferson Faconda y Edison Suquillo

Figura 3.29 Resultado obtenido.



Elaborado por: Jefferson Faconda y Edison Suquillo

## CAPÍTULO 4

### PRUEBAS Y RESULTADOS DE LOS ALGORITMOS PROPUESTOS

Para evaluar la eficacia de los algoritmos propuestos en la detección de las distintas etapas de la enfermedad en un conjunto de imágenes, se realizan varias pruebas en las que se modifican las variables más relevantes para este caso.

Además de evaluar el rendimiento de las redes neuronales, se mide el tiempo de procesamiento en las distintas pruebas que se pretenden realizar. Debido a que los tiempos que el algoritmo se demora en ejecutarse depende de la capacidad de procesamiento de la computadora. El análisis se realiza en computadoras con diferentes características de manera que se evalúa el efecto de la computadora en el tiempo de ejecución.

En concreto se utiliza tres máquinas diferentes. Las características de la primera máquina se pueden observar en la figura 4.1 la cual cuenta con una con tarjeta gráfica para los entrenamientos como se observa en la figura 4.2, esto debido a que, usando computadoras sin tarjetas gráficas, el tiempo es demasiado elevado y el proceso de entrenamiento se vuelve muy ineficiente, las características de la segunda y la tercera maquina se detallan en las figuras 4.3 y 4.4 correspondientemente, estas dos máquinas son utilizadas para evaluar las redes neuronales con las bases de datos disponibles.

Figura 4.1 Computadora usada para entrenamientos

Nombre del dispositivo	DESKTOP-EK4UJTN
Procesador	Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz 2.30 GHz
RAM instalada	16,0 GB (15,8 GB utilizable)
Id. del dispositivo	21DBA8C4-15C0-4B70-A928-A890E955FC2B
Id. del producto	00326-10000-00000-AA856
Tipo de sistema	Sistema operativo de 64 bits, procesador x64
Lápiz y entrada táctil	La entrada táctil o manuscrita no está disponible para esta pantalla

Elaborado por: Jefferson Faconda y Edison Suquillo

Figura 4.2 Tarjeta de video disponible en el PC

Elementos	Detalles
GeForce MX130	Versión del controla... 462.80
	Tipo de controlador: DCH
	Nivel de característi... 11_0
	Núcleos de CUDA: 384
	Reloj de gráficos: 1108 MHz
	Velocidad de datos ... 5.01 Gbps
	Interfaz de memoria: 64-bits
	Ancho de banda de ... 40.08 GB/s
	Memoria total dispo... 12189 MB
	Memoria de video d... 4096 MB GDDR5
	Memoria de video d... 0 MB

Elaborado por: Jefferson Faconda y Edison Suquillo

Figura 4.3 Maquina 1 utilizada para evaluar redes neuronales

Edición de Windows

Windows 10 Home  
© Microsoft Corporation. Todos los derechos reservados.



Sistema

Procesador: Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz  
Memoria instalada (RAM): 8,00 GB (7,89 GB utilizable)  
Tipo de sistema: Sistema operativo de 64 bits, procesador x64  
Lápiz y entrada táctil: Compatibilidad con entrada manuscrita

Configuración de nombre, dominio y grupo de trabajo del equipo

Nombre del equipo: DESKTOP-HM9TCJB [Cambiar configuración](#)  
Nombre completo de equipo: DESKTOP-HM9TCJB  
Descripción del equipo:  
Grupo de trabajo: WORKGROUP

Activación de Windows

Windows está activado [Lee los Términos de licencia del software de Microsoft](#)  
Id. del producto: 00356-02054-28078-AAOEM [Cambiar la clave de producto](#)

Elaborado por: Jefferson Faconda y Edison Suquillo

Figura 4.4 Maquina 2 utilizada para evaluar redes neuronales

[Ver información básica acerca del equipo](#)



Edición de Windows

Windows 10 Home  
© Microsoft Corporation. Todos los derechos reservados.

Sistema

Procesador: Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz 2.40 GHz  
Memoria instalada (RAM): 8,00 GB (7,89 GB utilizable)  
Tipo de sistema: Sistema operativo de 64 bits, procesador x64  
Lápiz y entrada táctil: La entrada táctil o manuscrita no está disponible para esta pantalla

Configuración de nombre, dominio y grupo de trabajo del equipo

Nombre de equipo: DESKTOP-E1N1ECE [Cambiar configuración](#)  
Nombre completo de equipo: DESKTOP-E1N1ECE  
Descripción del equipo:  
Grupo de trabajo: WORKGROUP

Activación de Windows

Windows está activado [Lea los Términos de licencia del software de Microsoft](#)

Elaborado por: Jefferson Faconda y Edison Suquillo

À continuación, se detalla cada prueba y se analizan los resultados obtenidos.

## 4.1 PRUEBAS REALIZADAS

### 4.1.1 Rendimiento de distintos tipos de redes

Esta prueba consiste en evaluar el desempeño de arquitecturas diferentes que han sido entrenadas y validadas con la base de datos disponible. La primera red fue creada y entrenada con la finalidad de simplificarla lo máximo posible. La segunda red preexistente, la cual es el modelo GoogLeNet, es entrenada mediante transferencia de aprendizaje. La tercera red, Resnet-18, se entrena de igual manera con transferencia de aprendizaje. Las métricas que se evalúan son la exactitud y la pérdida.

La exactitud se define como la relación entre los valores clasificados correctamente para el total de datos:

$$Exactitud = \frac{\text{Imágenes clasificadas correctamente}}{\text{Total de imágenes en la base de datos}}$$

La pérdida se cuantifica mediante la validación cruzada, que calcula la diferencia entre dos distribuciones de probabilidad.

Los datos de entrenamiento se componen de 1889 imágenes divididas en 4 etiquetas: Bueno, Fase 1, Fase 2 y Fase 3. El número de imágenes por etiqueta se muestran en la tabla 4.1.

Tabla 4.1. Número de imágenes de la base de datos de entrenamiento.

<b>Etiqueta</b>	<b>Número</b>
Bueno	331
Fase 1	433
Fase 2	578
Fase 3	547
Total	1889

Tabla de número de imágenes usado para la base de datos de entrenamiento. Elaborado por: Jefferson Faconda y Edison Suquillo

Un montaje de un grupo de imágenes de entrenamiento se indica en la figura 4.5.

Figura 4.5 Imágenes de frutos de cacao para entrenamiento.



Elaborado por: Jefferson Faconda y Edison Suquillo

Esta base de datos se subdividió de manera aleatoria en porcentajes de 75% de imágenes para entrenamiento y 25% de imágenes de validación.

Las imágenes de prueba se componen de tres grupos de imágenes tomadas en tres calidades de imágenes: 6 megapíxeles, 20 megapíxeles y 48 megapíxeles. En cada grupo de imágenes se subdividen en fotos tomadas a 3 distancias diferentes: 35 centímetros, 70 centímetros y 110 centímetros. En la tabla 4.2 se detalla el número de imágenes de prueba a 6 megapíxeles, en la tabla 4.3 se detalla el número de imágenes de prueba a 20 megapíxeles y en la tabla 4.4 se detalla el número de imágenes de prueba a 48 megapíxeles.

Tabla 4.2. Base de datos de imágenes en 6 Megapíxeles.

<b>Calidad [MPx]</b>	<b>Distancia [cm]</b>		
	<b>35</b>	<b>70</b>	<b>110</b>
Etiqueta	Número	Número	Número
Bueno	12	12	12
Fase 1	13	13	13
Fase 2	13	13	13
Fase 3	10	10	10
Total	48	48	48

Tabla de la base de datos de imágenes en 6 Megapíxeles. Elaborado por: Jefferson Faconda y Edison Suquillo

Tabla 4.3. Base de datos de imágenes en 20 Megapíxeles.

<b>Calidad [MPx]</b>	<b>Distancia [cm]</b>		
	<b>35</b>	<b>70</b>	<b>110</b>
Etiqueta	Número	Número	Número
Bueno	12	12	12
Fase 1	10	10	10
Fase 2	14	14	14
Fase 3	10	10	10
Total	46	46	46

Tabla de la base de datos de imágenes en 20 Megapíxeles. Elaborado por: Jefferson Faconda y Edison Suquillo

Tabla 4.4. Base de datos de imágenes en 48 Megapíxeles.

<b>Calidad [MPx]</b>	<b>Distancia [cm]</b>		
	<b>35</b>	<b>70</b>	<b>110</b>
Etiqueta	Número	Número	Número
Bueno	9	10	6
Fase 1	8	9	9
Fase 2	14	13	11
Fase 3	3	8	5
Total	34	40	31

Tabla de la base de datos de imágenes en 48 Megapíxeles. Elaborado por: Jefferson Faconda y Edison Suquillo

En la figura 4.6 se muestra un montaje de algunas imágenes para pruebas.

Figura 4.6 Imágenes de cacaos para pruebas.



Elaborado por: Jefferson Faconda y Edison Suquillo

Para todos los casos de entrenamiento se aumenta la base de información con la aplicación de transformaciones de giro entre  $0[^\circ]$  y  $360[^\circ]$  y escalamiento de tamaño entre  $50[\%]$  y  $100[\%]$ .

A continuación, se describe los parámetros básicos que se utilizan para entrenar las redes neuronales convolucionales:

#### **Parámetros de entrenamiento de las redes:**

**Optimizador.** – “Adam” es un optimizador estocástico de primer orden, se utiliza como un optimizador predeterminado en aplicaciones de aprendizaje profundo debido a que al ser de primer orden su derivada tiene menos requerimientos computacionales en comparación con otros optimizadores de segundo orden.

**Épocas.** - 15 (número de épocas), donde una época es el paso completo del algoritmo de entrenamiento sobre todo el conjunto de entrenamiento.

**Tasa de aprendizaje.** - se utiliza un escalar positivo, se coloca 0.0001 para que el resultado sea más óptimo.

**Validación.** – La validación cada 50 iteraciones es por defecto del programa. La validación es una forma de comprobar que tan bien está generalizando las características de cada clase la red. Mediante la validación se comprueba que tan bien está entrenada la red porque son imágenes que solo se usan internamente para comprobar la exactitud, con eso evitan el sobreentrenamiento.

**Mezcla de épocas.** - La mezcla de datos ayuda a que la información que entra a la red sea lo más variada posible, de tal manera que la red encuentre las características reales que distinguen cada clase.

**Exactitud de entrenamiento.** - La exactitud de entrenamiento se mide durante cada época y se espera un 100%.

**Exactitud de la validación.** – La exactitud de validación se mide después de cada época y se espera un 100%.

**Perdida del entrenamiento.** - La pérdida de entrenamiento se mide durante cada época y se espera el menor porcentaje.

**Perdida de la validación.** - La pérdida de validación se mide después de cada época y se espera el menor porcentaje.

**Tiempo de entrenamiento.** – El tiempo que tarda el computador en procesar el entrenamiento de la red con los parámetros establecidos.

#### **4.1.2 Exactitud con distintos tamaños de imágenes y distancia variable**

En esta fase se evalúa el desempeño de las redes neuronales para el conjunto de imágenes de prueba. La métrica que se utiliza para evaluar el desempeño es la exactitud y también se usan las matrices de confusión para examinar de manera gráfica como se clasifican las imágenes.

El análisis se realiza con todas las redes entrenadas en la sección anterior. Para realizar esta evaluación, se pretende evaluar la exactitud de cada red tres veces, una por cada distancia establecida en 6 megapíxeles, y sucesivamente con cada calidad de imagen creciente.

#### **4.1.3 Comparación del comportamiento de las redes**

Para esta evaluación se analiza el comportamiento individual de las tres redes cuyo entrenamiento da el mejor desempeño de cada clase: Básica #2, Googlenet #1 y Resnet\_18 #1. Se estudia una imagen al azar por estado de la fruta de las imágenes tomadas a 35 centímetros de distancia en 6 megapíxeles de calidad, tabla 4.2, este grupo de imágenes de las que se obtiene mejores rendimientos para cada red.

Se examina la clase predicha por la red, la clase real y se utilizan herramientas visuales como: matriz de confusión, sensibilidad a la oclusión, mapeo de activación de clases ponderado por gradiente (Grad-CAM) y explicaciones locales interpretables agnósticas del modelo (LIME), que ayuda a tener una idea de los rasgos de la imagen que contribuyen en mayor medida a que el clasificador tome una decisión e identifique el estado del cacao.



A continuación, se describe las características de la matriz de confusión:

### **Matriz de confusión**

La matriz de confusión se representa por una tabla que sirve para describir el rendimiento de un modelo de clasificación comparando valores previstos y reales.

La matriz de confusión tiene los siguientes componentes básicos:

**Verdaderos Positivos.** - La predicción fue correcta y el valor verdadero es correcto.

**Verdaderos Negativos.** - La predicción fue incorrecta y el valor verdadero es incorrecto.

**Falsos Positivos.** – La predicción fue correcta, pero el valor verdadero fue incorrecto.

**Falsos Negativos.** - La predicción fue incorrecta, pero el valor verdadero es correcto.

Las imágenes para esta prueba se indican en la figura 4.7 donde se encuentra el cacao bueno, cacao con moniliasis fase 1, cacao con moniliasis fase 2, cacao con moniliasis fase 3.:

Figura 4.7 Imágenes para pruebas.



Elaborado por: Jefferson Faconda y Edison Suquillo

## 4.2 RESULTADOS

### 4.2.1 Rendimiento de distintos tipos de redes

A continuación, se detalla los resultados de los entrenamientos de las tres redes entrenadas:

#### Red básica

La red básica se crea de manera simplificada para comprender el funcionamiento de las redes neuronales convolucionales. Esta red consta únicamente de una capa de entrada de 224x224x3 pixeles; una capa convolucional; una capa de activación del tipo ‘ReLU’; Una capa de reducción; una capa totalmente conectada con 4 neuronas; una capa de activación de tipo ‘softmax’ y la capa de clasificación. Las características principales de la red se muestran en la imagen 4.8:

Figura 4.8 Características de la red básica.

ANALYSIS RESULT				
	Name	Type	Activations	Learnables
1	imageinput 224x224x3 images with 'zerocenter' normalization	Image Input	224x224x3	-
2	conv 20 5x5 convolutions with stride [1 1] and padding [0 0 0 0]	Convolution	220x220x20	Weights 5x5x3x20 Bias 1x1x20
3	relu ReLU	ReLU	220x220x20	-
4	maxpool 2x2 max pooling with stride [2 2] and padding [0 0 0 0]	Max Pooling	110x110x20	-
5	fc 3 fully connected layer	Fully Connected	1x1x3	Weights 3x242000 Bias 3x1
6	softmax softmax	Softmax	1x1x3	-
7	classoutput crossentropyex	Classification Output	1x1x3	-

Elaborado por: Jefferson Faconda y Edison Suquillo

Para la fase de entramiento se usan los parámetros indicados en la tabla 4.5:

Tabla 4.5. Parámetros de entrenamiento de la red básica

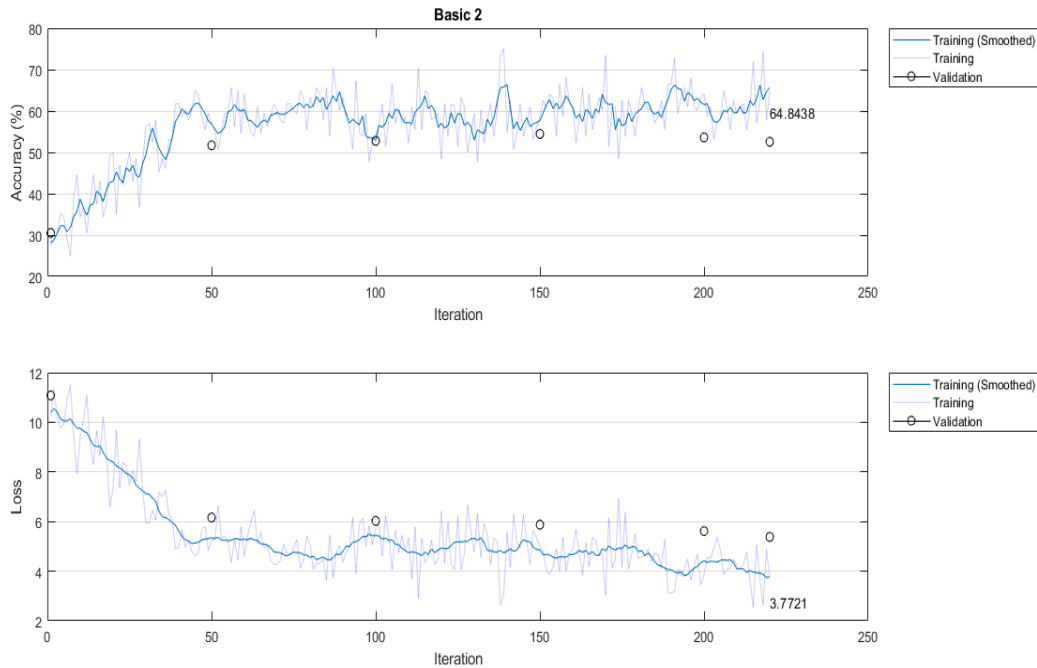
Parámetro	Valor
Optimizador	Adam
Épocas	15
Tasa de aprendizaje	0.0001
Validación	Cada 50 iteraciones
Mezcla de datos	Cada época

Tabla de parámetros de entrenamiento de la red básica analizar. Elaborado por: Jefferson Faconda y Edison Suquillo

Todos los demás parámetros de entrenamiento se los deja por defecto del Toolbox de redes neuronales de Matlab.

El proceso de entrenamiento de una corrida se muestra en la figura 4.9, el resto de las curvas de entrenamiento se muestran en el anexo 1 donde los valores “X” son los números de las iteraciones y “Y” es el porcentaje de exactitud del entrenamiento.

Figura 4.9 Proceso de entrenamiento de la red básica.



Elaborado por: Jefferson Faconda y Edison Suquillo

Se observa un aumento de exactitud de entrenamiento al inicio. Sin embargo, al llegar a un valor cercano a 55% el aumento de exactitud se detiene y estabiliza, lo cual significa que esta red no puede encontrar más patrones en la base de datos. Se observa como la exactitud de validación se mantiene por debajo y cerca de la exactitud del entrenamiento, esto es una señal de sobre ajuste, fenómeno no deseado que significa que la red comienza a generalizar en exceso los datos de entrenamiento y no los datos de prueba. Por lo que su rendimiento solo es bueno en los datos de prueba.

La pérdida de entrenamiento disminuye al inicio, y del mismo modo que con la exactitud, se detiene la reducción y se mantiene estable en un valor cercano a 4%. Lo que significa que los errores que encuentra la red ya no pueden ser disminuidos. La pérdida de validación se mantiene por encima y cerca de la curva de entrenamiento, esto también es señal de sobre ajuste.

En ambas graficas se puede observar un comportamiento irregular de las curvas de exactitud y perdida de entrenamiento.

En la tabla 4.6 se observan los resultados obtenidos en todos los entrenamientos realizados en esta red:

Tabla 4.6 Resultados de entrenamiento de la red básica.

N° de prueba	Exactitud del entrenamiento	Exactitud de la validación	Pérdida del entrenamiento	Pérdida de la validación	Tiempo de entrenamiento [min]
1	54.6875	56.9915	6.1434	5.6986	21
2	64.8438	52.5424	3.7721	5.381	26
3	66.4062	54.661	3.6976	6.0265	23

Tabla de resultados del entrenamiento de la red básica. Elaborado por: Jefferson Faconda y Edison Suquillo

## GoogLeNet

Esta arquitectura de red neuronal es diseñada y optimizada para clasificar 1000 etiquetas diferentes. Para la aplicación del trabajo se la entrena de nuevo mediante transferencia de aprendizaje. De esta manera, se adaptan los pesos de cada capa con la finalidad de adaptarse a la base de datos de este estudio.

Los parámetros de entrenamiento de esta red son los mismos que para la red básica como se muestran en la tabla 4.7.

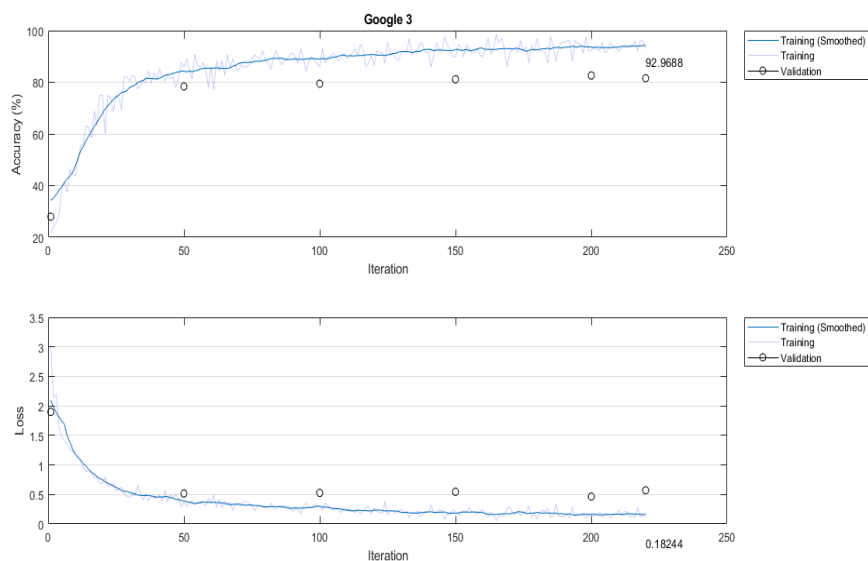
Tabla 4.7 Parámetros de entrenamiento de la red GoogLeNet.

Parámetro	Valor
Optimizador	Adam
Épocas	15
Tasa de aprendizaje	0.0001
Validación	Cada 50 iteraciones
Mezcla de datos	Cada época

Tabla de parámetros de entrenamiento de la red GoogLeNet analizar. Elaborado por: Jefferson Faconda y Edison Suquillo

La fase de entrenamiento de una corrida se puede observar la figura 4.10. El resto de los entrenamientos se encuentran en el anexo 1:

Figura 4.10 Proceso de entrenamiento de la red GoogLeNet



Elaborado por: Jefferson Faconda y Edison Suquillo

En esta gráfica se tiene que la exactitud de entrenamiento tiene un incremento más pronunciado que en el caso de la red básica, y se estabiliza en valores cercanos a 90%. Se tiene también que la validación de exactitud se mantiene en valores cercanos y por debajo de la curva de entrenamiento, la diferencia con los valores de entrenamiento muestra signos de sobreajuste.

En cuanto a la curva de pérdida de entrenamiento, su comportamiento indica una disminución hasta valores muy cercanos a cero y se mantiene estable luego de esto. La pérdida de validación también disminuye y tiene valores mayores a aquellos de la curva de entrenamiento.

El comportamiento de las curvas de exactitud y pérdida de entrenamiento es más suave que en el caso de la red básica.

En la tabla 4.8 se puede observar los resultados obtenidos por esta red:

Tabla 4.8 Resultados de la red GoogLeNet

N° de prueba	Exactitud del entrenamiento	Exactitud de la validación	Perdida del entrenamiento	Perdida de la validación	Tiempo de entrenamiento [min]
1	96.875	85.5932	0.0757	0.4538	144
2	92.9688	81.5678	0.1824	0.5706	135
3	93.75	83.0508	0.1914	0.5078	138

Tabla de resultados del entrenamiento de la red GoogLeNet. Elaborado por: Jefferson Faconda y Edison Suquillo

En base a los resultados obtenidos se observa que la red reentrenada GoogLeNet tiene un desempeño superior a la red básica. También, de entre las redes GoogLeNet se tiene que la primera corrida de entrenamiento se desempeña mejor que las otras.

## Resnet-18

Esta red neuronal también es capaz de reconocer imágenes entre 1000 clases. De igual manera se entrena esta red usando transferencia de aprendizaje. Los parámetros de entrenamiento se indican en la tabla 4.9.

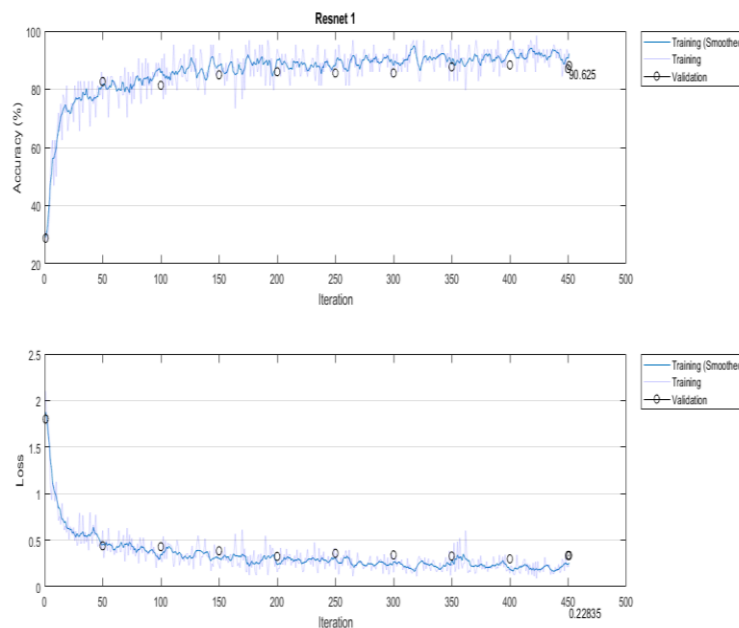
Tabla 4.9 Parámetros de entrenamiento de la red Resnet\_18.

Parámetro	Valor
Optimizador	Adam
Épocas	15
Tasa de aprendizaje	0.0001
Validación	Cada 50 iteraciones
Mezcla de datos	Cada época

Tabla de parámetros de entrenamiento de la red Resnet\_18 analizar. Elaborado por: Jefferson Faconda y Edison Suquillo

Las curvas de exactitud y pérdida de una corrida de entrenamiento se muestran en la figura 4.11, el resto de las curvas de entrenamiento se encuentran en el anexo 1:

Figura 4.11 Entrenamiento de la red Resnet-18.



Elaborado por: Jefferson Faconda y Edison Suquillo

Se observa en la curva de exactitud como los valores de entrenamiento se incrementan de manera constante hasta llegar a estabilizarse en un valor cercano a 90%. Los valores de exactitud en validación se encuentran muy cercanos a la curva de entrenamiento. Esta es buena señal, debido a que la red está generalizando las características reales para diferenciar los frutos. En la curva de pérdida en entrenamiento el comportamiento es descendente hasta estabilizarse en un valor cercano a 0.2. La comprobación de pérdida en validación da valores muy cercanos a los del entrenamiento. De manera similar con los valores de precisión, esto es muestra de un entrenamiento idóneo sin sobreajuste.

En la tabla 4.10 se muestran los valores obtenidos para los entrenamientos con esta red.

Tabla 4.10 Resultados de entrenamiento de la red Resnet\_18.

N° de prueba	Exactitud del entrenamiento	Exactitud de la validación	Pérdida del entrenamiento	Pérdida de la validación	Tiempo de entrenamiento [min]
1	90.625	88.1356	0.2284	0.3361	156
2	95.3125	88.5593	0.1247	0.3108	159
3	93.75	86.0169	0.211	0.4291	153

Tabla de resultados del entrenamiento de la red Resnet\_18. Elaborado por: Jefferson Faconda y Edison Suquillo

En base a la tabla anterior se puede ver que tiene un desempeño similar a la red GoogLeNet y entre las redes de esta clase el segundo entrenamiento se desempeña de mejor manera.

#### 4.2.2 Exactitud con distintos tamaños de imágenes y distancia variable

A continuación, se detalla los resultados de las pruebas realizadas a distintas distancia y diferente calidad de imagen:

##### Resultados para imágenes de 6 megapíxeles

En la tabla 4.11 se tiene que todas las redes por lo general se desempeñan mejor a una distancia de 35 [cm], luego en una distancia de 70 [cm] y el peor desempeño se obtiene con una distancia de 110 [cm].

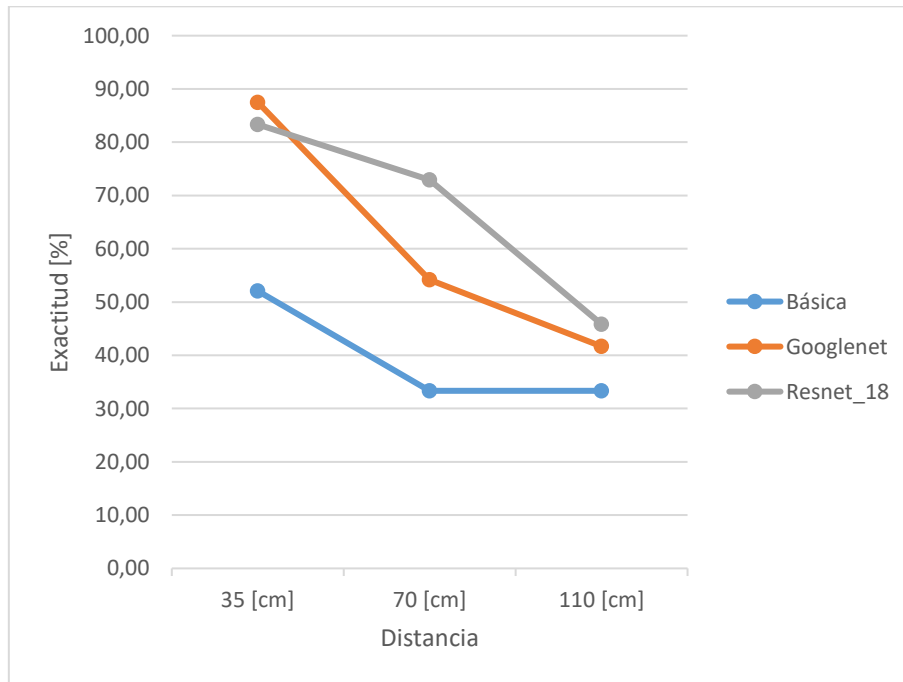
Tabla 4.11 Exactitud de las redes en imágenes de 6[MPx] a tres distancias.

N° de entrenamiento	Red	Distancia [cm]		
		35	70	110
1	Básica	27.08	27.08	29.17
2	Básica	52.08	33.33	33.33
3	Básica	47.92	29.17	29.17
1	Googlenet	87.50	54.17	41.67
2	Googlenet	85.42	62.50	47.92
3	Googlenet	81.25	68.75	47.92
1	Resnet_18	83.33	72.92	45.83
2	Resnet_18	83.33	50.00	29.17
3	Resnet_18	72.92	50.00	33.33

Tabla de Exactitud de las redes en imágenes a tres distancias. Calidad 6 Megapíxeles. Elaborado por: Jefferson Faconda y Edison Suquillo

Además, de cada clase de red se desempeñan mejor las redes: Básica #2, Googlenet #1 y Resnet\_18 #1. El comportamiento de estas 3 redes se observa en la figura 4.11:

Figura 4.12 Exactitud de las redes propuestas con imágenes a 6[MPx]



Elaborado por: Jefferson Faconda y Edison Suquillo



La red Resnet\_18 tiene un mejor despeño en las distancias de 70 y 110 [cm], en tanto que la red Googlenet tiene una mayor exactitud a una distancia de 35 [cm]. La red básica tiene el peor desempeño de todas.

En la tabla 4.12 se observa que el tiempo de ejecución del análisis de precisión de cada red es muy similar en todos los casos evaluados, con tiempos que rondan para la PC1 los 8 [s] y para la PC2 los 15 [s].

Tabla 4.6. Tiempo de ejecución en PC1. Calidad 6 [MPx].

Red	N° de entrenamiento	PC	Tiempo de ejecución [s]		
			35 [cm]	70 [cm]	110 [cm]
Básica	1	1	8,49	5,30	6,77
		2	13,73	12,46	12,21
	2	1	6,04	6,72	6,69
		2	12,61	11,74	12,38
	3	1	6,44	6,34	6,40
		2	12,74	12,47	12,49
Googlenet	1	1	9,35	7,57	7,54
		2	16,27	14,27	14,27
	2	1	8,56	7,63	7,98
		2	15,65	14,23	15,12
	3	1	8,31	7,39	7,60
		2	19,37	16,99	14,98
Resnet	1	1	7,68	7,56	7,68
		2	15,59	13,55	14,61
	2	1	7,71	7,43	7,77
		2	13,67	13,99	13,04
	3	1	7,43	7,45	8,07
		2	15,30	14,94	14,26

Tabla del tiempo de ejecución. Calidad 6 Megapíxeles. Elaborado por: Jefferson Faconda y Edison Suquillo

### Pruebas en imágenes de 20 Megapíxeles

En la tabla 4.13 se puede observar que todas las redes se comportan mejor para distancias de 35[cm], en especial las redes Googlenet y Resnet\_18. El desempeño para 70[cm] y 100[cm] es similar en las redes Básica y Googlenet. Sin embargo, la red Resnet\_18 se comporta mejor en distancias de 70[cm] que en 110[cm].

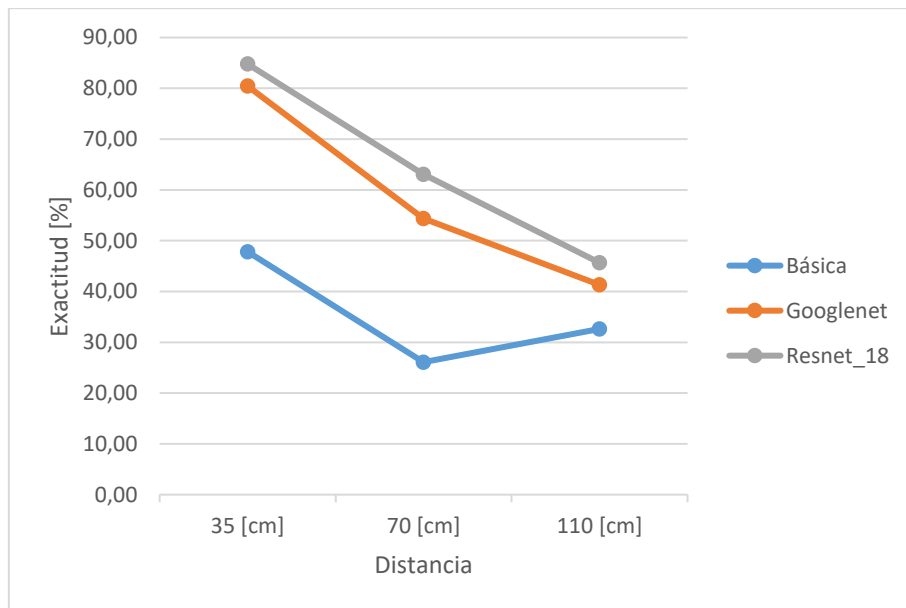
Tabla 4.13 Exactitud de las redes en imágenes de 20[MPx] a tres distancias.

N° de entrenamiento	Red	Distancia [cm]		
		35	70	110
1	Básica	39.13	41.30	34.78
2	Básica	47.83	26.09	32.61
3	Básica	36.96	34.78	34.78
1	Googlenet	80.43	54.35	41.30
2	Googlenet	80.43	52.17	50.00
3	Googlenet	84.78	45.65	47.83
1	Resnet_18	84.78	63.04	45.65
2	Resnet_18	63.04	52.17	30.43
3	Resnet_18	76.09	54.35	41.30

Tabla de exactitud de las redes en imágenes a tres distancias. Calidad 20 Megapíxeles. Elaborado por: Jefferson Faconda y Edison Suquillo

Del mismo modo que para 6 [MPx] las redes Básica #2, Googlenet #1 y Resnet\_18 #1 tienen el mejor rendimiento de su clase.

Figura 4.13 Exactitud de las redes propuestas con imágenes a 20[MPx].



Elaborado por: Jefferson Faconda y Edison Suquillo

Se observa que Resnet\_18 tiene un desempeño superior al de las demás redes en todas las distancias, Seguidamente esta la red Googlenet que tiene un desempeño similar pero inferior. La red Básica se desempeña de manera inferior para todas las distancias.

Se observa en la tabla 4.14 que los tiempos de evaluación de precisión en este caso son muy similares en torno a 20 [s] en la PC1 y para la PC2 los 40 [s] en todas redes.

Tabla 4.7. Tiempo de ejecución en PC1. Calidad 20 [MPx].

Red	N° de entrenamiento	PC	Tiempo de ejecución [s]		
			35 [cm]	70 [cm]	110 [cm]
Básica	1	1	24,38	21,12	20,74
		2	36,35	37,31	38,34
	2	1	20,69	21,31	21,76
		2	38,62	42,61	43,61
	3	1	20,72	21,01	21,65
		2	37,53	39,93	40,43
Googlenet	1	1	22,64	22,62	22,77
		2	41,31	42,54	44,45
	2	1	22,57	22,60	23,01
		2	43,71	43,90	43,78
	3	1	22,50	22,28	22,44
		2	44,69	49,81	45,73
Resnet	1	1	22,15	21,97	22,39
		2	39,30	37,72	37,09
	2	1	21,96	21,81	22,81
		2	41,08	51,85	42,19
	3	1	21,84	22,30	22,54
		2	39,61	37,98	38,84

Tabla del tiempo de ejecución. Calidad 20 Megapíxeles. Elaborado por: Jefferson Faonda y Edison Suquillo

### Pruebas en imágenes de 48 Megapíxeles:

Al observar la tabla 4.15 y la figura 4.13 se puede decir que la totalidad de las redes se desempeñan mejor a 35[cm]. Además, la diferencia en exactitud para 70[cm] y 110[cm] es similar en todas las redes.

Tabla 4.8. Exactitud de las redes en imágenes de 48[MPx] a tres distancias.

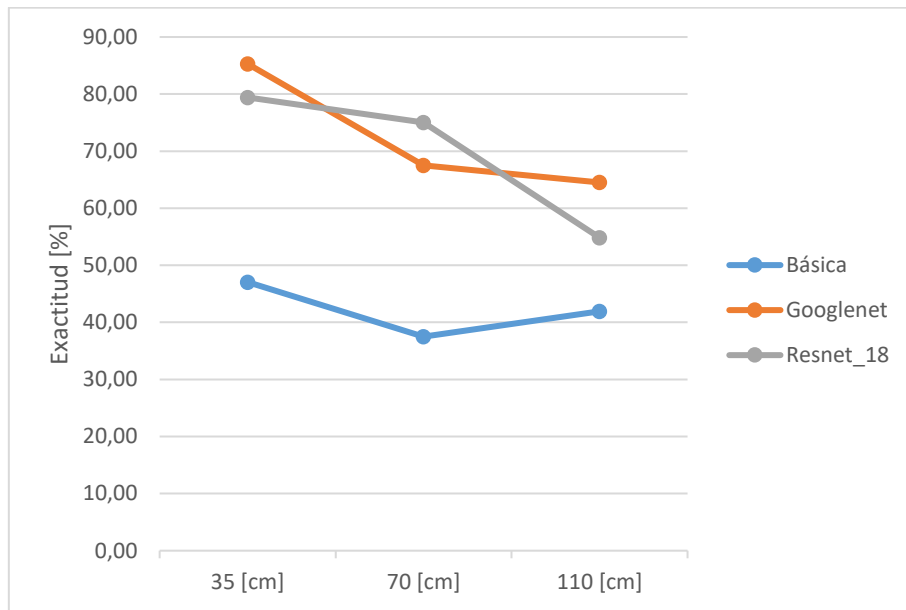
N° de entrenamiento	Red	Distancia [cm]		
		35	70	110
1	Básica	44.12	35.00	38.71
2	Básica	47.06	37.50	41.94
3	Básica	44.12	32.50	29.03
1	Googlenet	85.29	67.50	64.52
2	Googlenet	76.47	52.50	48.39
3	Googlenet	79.41	55.00	54.84
1	Resnet_18	79.41	75.00	54.84
2	Resnet_18	64.71	52.17	45.16
3	Resnet_18	58.82	54.35	29.03

Tabla de exactitud de las redes en imágenes a tres distancias. Calidad 48 Megapíxeles. Elaborado por: Jefferson

Faonda y Edison Suquillo

Las redes que mejor se desempeñan son: Básica #2, Googlenet #1 y Resnet\_18 #1.

Figura 4.14 Exactitud de las redes propuestas con imágenes a 48[MPx].



Elaborado por: Jefferson Faconda y Edison Suquillo

En esta prueba se obtiene un mejor comportamiento de la red Googlenet para 35 y 110 [cm], en tanto que Resnet\_18 únicamente sobresale al clasificar imágenes de 70[cm] de distancia. Al igual que en las pruebas anteriores, la red básica no se desempeña bien en ninguna distancia.

Observando la tabla 4.16 se puede ver que los tiempos son similares para todas las redes en cada distancia de estudio. En todas las imágenes a 70 [cm] la red se tarda alrededor de 60[s] para la PC1 y para la PC2 alrededor de 120[s] en evaluar cada grupo de imágenes. En cambio, En las imágenes a 35 y 100 [cm] se tarda el algoritmo alrededor de 50 [s] en la PC1 y en la PC2 alrededor de 85[s] en ejecutarse. El motivo de la diferencia en los tiempos se debe a la calidad de imágenes que se encuentran en ese grupo de imágenes.

Tabla 4.9. Tiempo de ejecución en PC1. Calidad 48 [MPx].

Red	N° de entrenamiento	PC	Tiempo de ejecución [s]		
			35 [cm]	70 [cm]	110 [cm]
Básica	1	1	62,44	62,65	43,87
		2	84,57	138,88	89,16
	2	1	43,51	56,66	42,57
		2	87,13	135,17	82,42
	3	1	44,21	57,52	42,67
		2	84,72	123,96	82,00
Googlenet	1	1	49,11	63,19	46,87
		2	91,53	143,87	85,95
	2	1	48,23	62,53	44,61
		2	86,99	119,37	90,80
	3	1	48,33	59,37	44,78
		2	106,27	133,28	84,39
Resnet	1	1	44,28	63,14	44,45
		2	84,12	115,33	79,96
	2	1	46,14	58,48	44,23
		2	90,60	129,02	86,81
	3	1	47,85	64,05	44,07
		2	85,42	118,41	75,64

Tabla del tiempo de ejecución. Calidad 48 Megapíxeles. Elaborado por: Jefferson Faconda y Edison Suquillo

### 4.2.3 Comparación del comportamiento de las redes

Los resultados obtenidos de las distintas redes entrenadas se los describe a continuación:

#### Red básica

La matriz de confusión para esta red se muestra en la figura 4.15, de la tabla 4.15, para esta red en la distancia y calidad estudiada se tiene una exactitud de 47.06%:

Figura 4.15 Matriz de confusión para la red básica.

		48[MP]-35[cm]			
		bueno	fase_1	fase_2	fase_3
True Class	bueno		5	2	2
	fase_1		3	4	1
	fase_2		1	11	2
	fase_3			1	2
		Predicted Class			

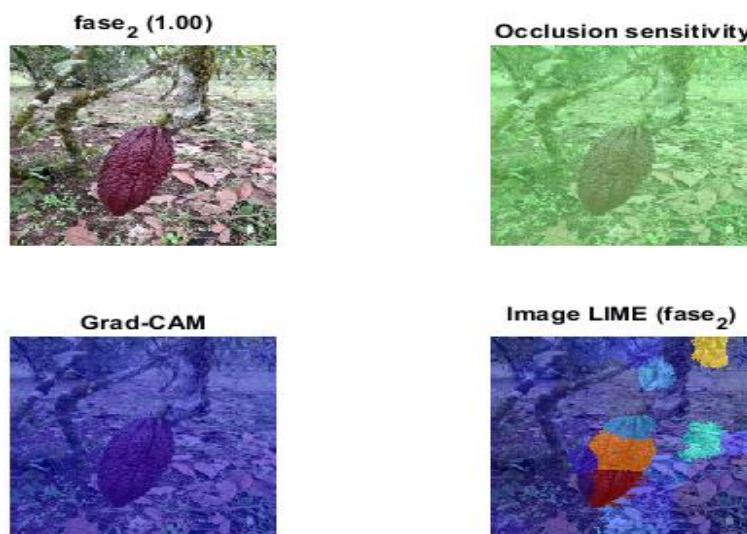
Elaborado por: Jefferson Faconda y Edison Suquillo

En esta matriz se observa un comportamiento irregular de esta red, no fue posible clasificar ningún cacao bueno correctamente y menos de la mitad de los cacaos con moniliasis fase 1 fueron reconocidos correctamente. También se observa que la red clasifica bien la mayoría de los cacaos con moniliasis fase 2 y se tiene la tendencia a clasificar muchas frutas con esta etapa del hongo. Los cacaos con moniliasis fase 3 se clasifican bien y también la red reconoce algunos cacaos buenos con esta fase.

A continuación, en las técnicas de visualización se analiza con más detalle el comportamiento de la red.

### Técnicas de visualización en cacaos buenos:

Figura 4.16 Técnicas de visualización para cacaos buenos. Red básica.



Elaborado por: Jefferson Faconda y Edison Suquillo

La figura 4.16 muestra un cacao bueno, sin embargo, este fue clasificado como cacao con moniliasis fase 2 con probabilidad del 100%. Se observa que para este estado del cacao no existe activación en la sensibilidad a la oclusión ni con Grad-CAM. Con la técnica LIME se observa que hay mayor activación de la red en la parte inferior derecha del cacao y la parte central para clasificarlo como en etapa 2 de la enfermedad.

### Técnicas de visualización en cacaos con moniliasis fase 1:

Figura 4.17 Técnicas de visualización para cacaos fase 1. Red básica.

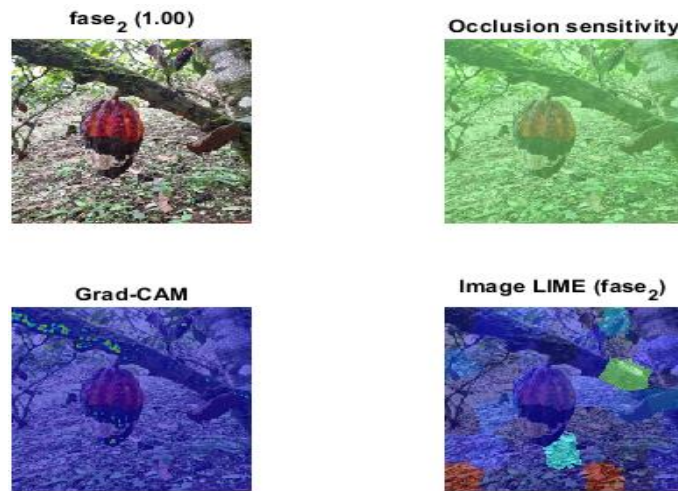


Elaborado por: Jefferson Faonda y Edison Suquillo

La figura 4.17 muestra un cacao en etapa 1 de moniliasis y se clasifica incorrectamente como cacao fase 2 con una probabilidad del 100%. Para la técnica de sensibilidad a la oclusión existe una gran importancia en un área fuera del fruto, y con Grad-CAM no existe activación de la red. La técnica LIME muestra una mayor activación de la red para la zona inferior de la foto que corresponde a una a hojas en el suelo.

### Técnicas de visualización en cacaos con moniliasis fase 2:

Figura 4.18 Técnicas de visualización para cacaos fase 2. Red básica.

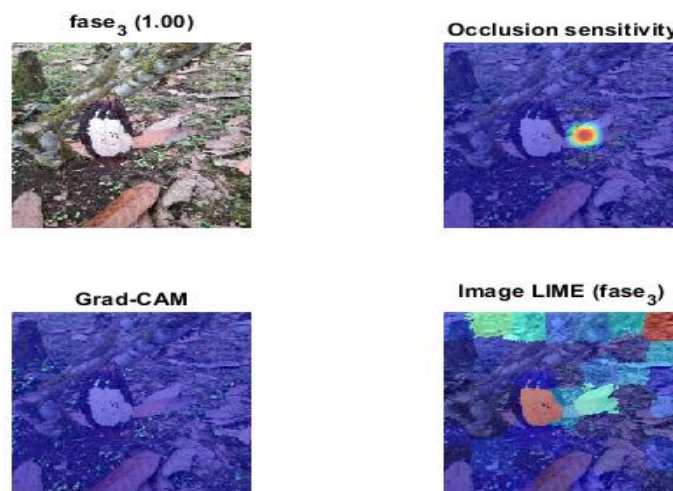


Elaborado por: Jefferson Faconda y Edison Suquillo

La figura 4.18 corresponde a un cacao con moniliasis en fase 2, y se clasifica correctamente con una probabilidad del 100%. Con la técnica de sensibilidad a la oclusión no se observan zonas de activación de la red. Con la técnica Grad-CAM se observan zonas iluminadas en amarillo de activación media en los alrededores del fruto y en la rama del árbol que no son relevantes al fruto. Mediante la técnica LIME se observa una activación positiva de la red en el suelo para clasificar al fruto, esto es algo no deseado.

### Técnicas de visualización en cacaos con moniliasis fase 3:

Figura 4.19 Técnicas de visualización para cacaos fase 3. Red básica.



Elaborado por: Jefferson Faconda y Edison Suquillo



En la figura 4.19 tanto la clasificación determinada por la red (fase 3), como la etapa de enfermedad real son iguales. La técnica de sensibilidad de oclusión muestra una zona roja de contribución positiva a la clasificación en una zona fuera del cacao, en una hoja que está en el suelo. La técnica Grad-CAM no muestra zonas de activación para evaluar el fruto. Por último, con la técnica LIME la zona de mayor contribución a la red esta fuera del fruto en la esquina superior de la figura, existe una activación media en la zona blanca del hongo.

En base a los resultados de las técnicas de visualización y la matriz de confusión, se concluye que la red básica durante su entrenamiento no ha sido capaz de generalizar las características que diferencian las etapas de enfermedad del cacao.

### Red GoogLeNet

La matriz de confusión para esta red se observa en la figura 4.20, de la tabla 4.15, para esta red en la distancia y calidad se tiene una exactitud de 85.29%:

Figura 4.20 Matriz de confusión para la red Googlenet.

		48[MP]-35[cm]			
		bueno	fase_1	fase_2	fase_3
True Class	bueno	8	1		
	fase_1		5	3	
	fase_2		1	13	
	fase_3				3
		Predicted Class			

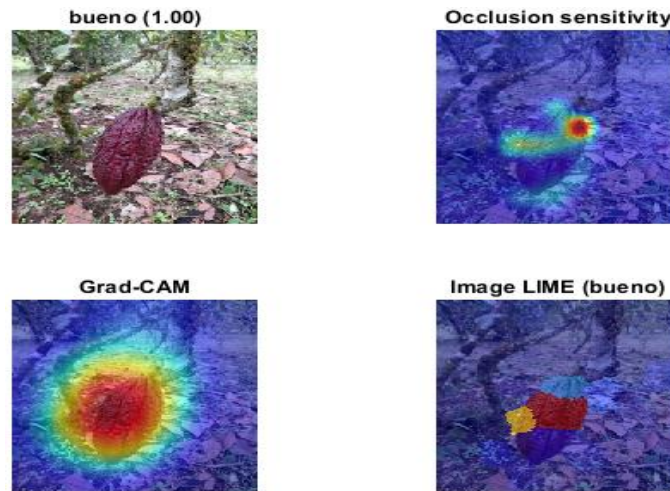
Elaborado por: Jefferson Faconda y Edison Suquillo

Se observa en esta gráfica una clasificación correcta en la totalidad de los casos de cacaos con moniliasis 3, y una exactitud casi perfecta para cacaos buenos y en fase 2. En el caso de cacaos en etapa 1 de moniliasis se obtiene que se clasifican tres imágenes de ocho como cacaos en etapa 2 de moniliasis. Al examinar los conjuntos de datos de estas dos clases de frutos se observa que la diferenciación entre ambas etapas no es intuitiva incluso para los humanos.

Seguidamente, con las técnicas de visualización se analiza con más detalle el comportamiento de esta red.

### Técnicas de visualización en cacaos buenos:

Figura 4.21 Técnicas de visualización para cacaos buenos. Red Googlenet.

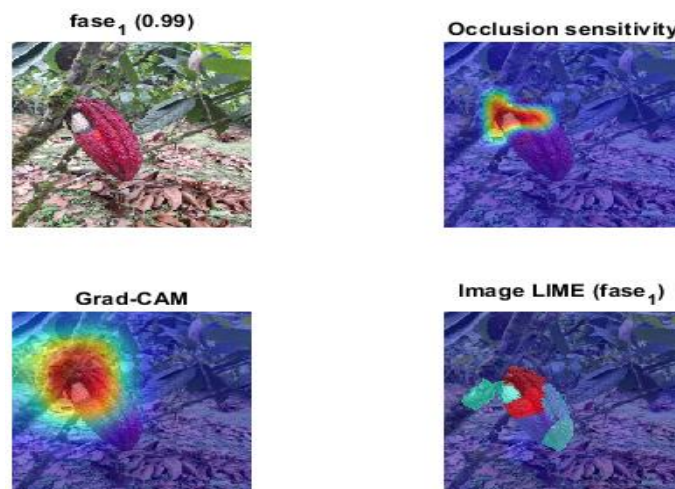


Elaborado por: Jefferson Faconda y Edison Suquillo

La figura 4.21 corresponde a un cacao sano y se clasifica de manera correcta con una probabilidad del 100%. En la técnica de sensibilidad a la oclusión se activa con mayor intensidad la parte superior del cacao para establecer la clasificación. Usando la técnica Grad-CAM se produce una mayor activación en la totalidad del fruto. Por último, con la técnica LIME se observa zonas de mayor importancia para la red en la parte media del cacao.

### Técnicas de visualización en cacaos con moniliasis fase 1:

Figura 4.22 Técnicas de visualización para cacaos fase 1. Red Googlenet.

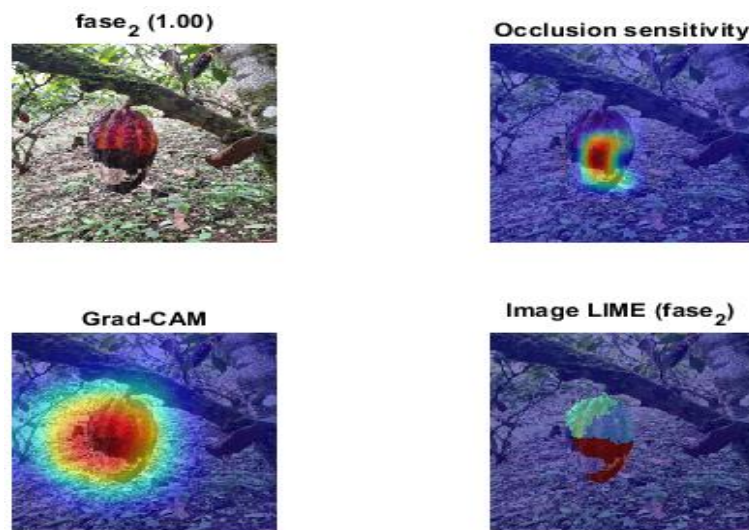


Elaborado por: Jefferson Faconda y Edison Suquillo

Para la figura 4.22, la clase predicha por la red es cacao con moniliasis fase 1 y probabilidad del 99%, es igual a la etapa real de la enfermedad. El método de sensibilidad a la oclusión muestra una contribución más positiva para esta clasificación en la zona blanca donde se ubica el hongo. En tanto que, con la técnica Grad-CAM la parte superior correspondiente a la región del hongo activa la red. La técnica LIME muestra que la zonas derecha e inferior alrededor del hongo las más importante para que la red clasifique la imagen.

### **Técnicas de visualización en cacaos con moniliasis fase 2:**

Figura 4.23 Técnicas de visualización para cacaos fase 2. Red Googlenet.

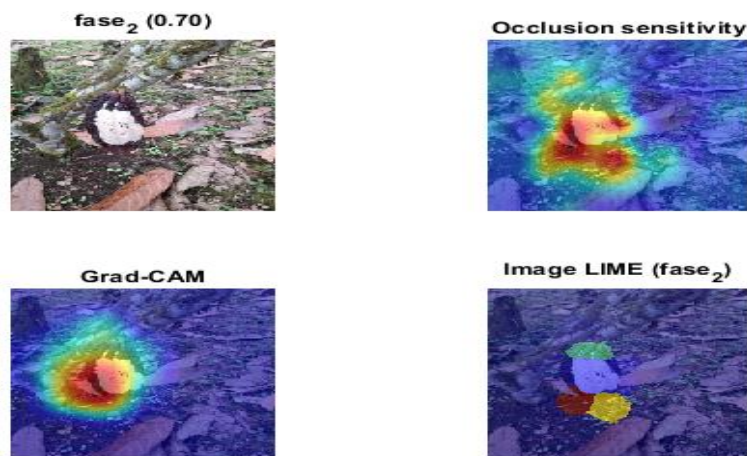


Elaborado por: Jefferson Faconda y Edison Suquillo

En la figura 4.23 se puede examinar que tanto la fase 2 de moniliasis, como la clasificación dada por esta red son iguales. La técnica de sensibilidad de oclusión muestra que la zona negra encima de la parte blanca del hongo posee una importancia mayor para la red. La técnica Grad-CAM, indica que la zona encima del hongo activa la red para realizar la clasificación. Usando LIME, la zona de mayor influencia para la red esta alrededor de la parte blanca del hongo.

### Técnicas de visualización en cacaos con moniliasis fase 3:

Figura 4.24 Técnicas de visualización para cacaos fase 3. Red Googlenet.



Elaborado por: Jefferson Faconda y Edison Suquillo

En este caso la figura 4.24 muestra un cacao en etapa 3 del hongo, sin embargo, red clasifica erróneamente el fruto como cacao con moniliasis etapa 2 con probabilidad del 70%. La técnica de sensibilidad a la oclusión muestra que la región que produce la mayor activación de la red se encuentra directamente debajo del hongo. Con la herramienta Grad-CAM se tiene una influencia mayor a la izquierda del hongo. Finalmente, con la técnica LIME se observa que la región más importante para la red para clasificar la imagen está ubicada en la parte inferior de la fruta y también en el suelo.

Con base en los resultados anteriores se puede establecer que, para clasificar el estado de enfermedad del cacao, esta red se enfoca en la coloración y el tamaño del hongo en caso de existir. También, la red da más importancia a los alrededores del hongo para todas las etapas de moniliasis.

### Red Resnet\_18

La matriz de confusión para esta arquitectura de red neuronal se indica a continuación. Esta red en este grupo de imágenes tiene una exactitud del 79.41%:

Figura 4.25 Matriz de confusión para la red Resnet\_18.

**48[MP]-35[cm]**

	bueno	fase_1	fase_2	fase_3	
True Class	bueno	7	2		
	fase_1		4	4	
	fase_2		1	13	
	fase_3			3	
		bueno	fase_1	fase_2	fase_3
		Predicted Class			

Elaborado por: Jefferson Faconda y Edison Suquillo

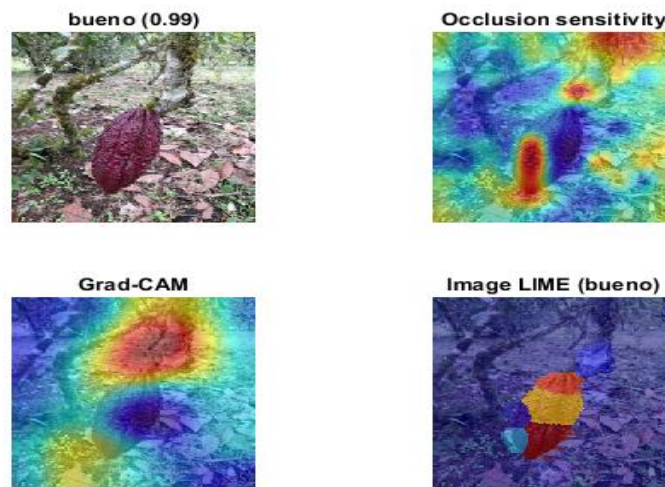
La matriz de confusión de Resnet\_18 indica que, únicamente las imágenes de cacao con moniliasis fase 3 se clasifican correctamente en todos los casos, los cacaos con moniliasis fase 2 y cacaos buenos se clasifican bien y solo erra la red en uno y dos casos respectivamente. Los cacaos con fase 1 de la enfermedad son los frutos con los que tiene más problemas la red, puesto que se confunde la mitad de estos con cacaos en fase 2 de moniliasis.

Al igual que con la red Googlenet, al examinar las imágenes, la diferencia entre las etapas 1 y 2 de moniliasis no resulta muy evidente para los humanos.

A continuación, en las técnicas de visualización se analiza con más detalle el comportamiento de la red.

### Técnicas de visualización en cacaos buenos:

Figura 4.26 Técnicas de visualización para cacaos sanos. Red Resnet\_18.

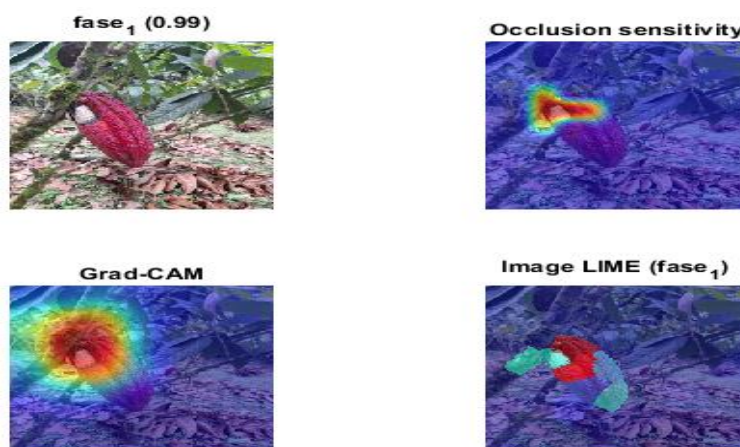


Elaborado por: Jefferson Faconda y Edison Suquillo

Se observa en la figura 4.26 que la red clasifica correctamente el cacao como bueno con una probabilidad del 99%. La sensibilidad a la oclusión indica que las regiones más importantes para la red están la parte inferior del cacao y las hojas de los alrededores. Usando Grad-CAM se observa que la parte encima del fruto es de mayor importancia para realizar la clasificación, que es algo no esperado. Finalmente, mediante la técnica LIME se tiene que, en especial la parte inferior del cacao contribuye a que la red reconozca que el fruto está sano. Igualmente, todo el fruto se reconoce para clasificarlo como bueno.

### Técnicas de visualización en cacaos con moniliasis fase 1:

Figura 4.27 Técnicas de visualización para cacaos fase 1. Red Googlenet.

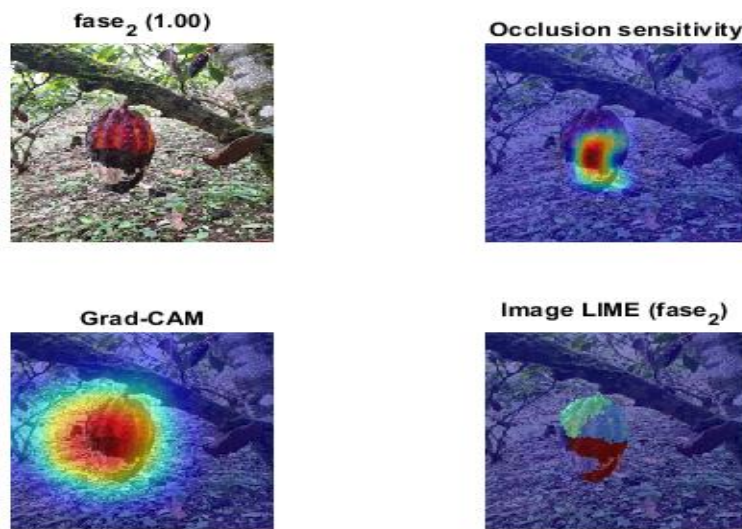


Elaborado por: Jefferson Faconda y Edison Suquillo

En la figura 4.27 se observa que la imagen está clasificada correctamente como cacao con moniliasis fase 1 con una probabilidad del 99%. También, con la técnica de sensibilidad a la oclusión, la red se activa positivamente para buscar cacaos con esta fase de la enfermedad tanto en la parte blanca del hongo, como en la zona superior del fruto. De la misma manera, Con Grad-CAM se obtiene que la red se activa mayormente tanto dentro y fuera del área del cacao. Mediante la técnica LIME, la red se enfoca en los alrededores del hongo con especial énfasis en la región encima del hongo.

### Técnicas de visualización en cacaos con moniliasis fase 2:7

Figura 4.28 Técnicas de visualización para cacaos fase 2. Red Googlenet.

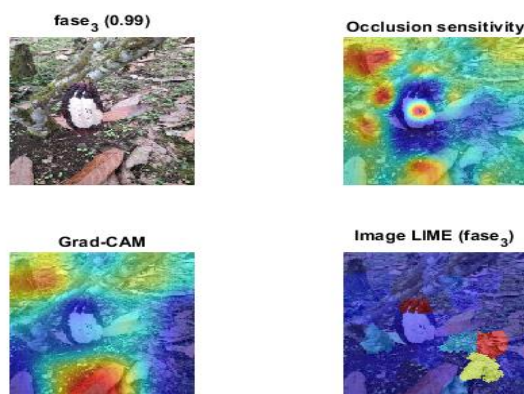


Elaborado por: Jefferson Faconda y Edison Suquillo

En la figura 4.28 se puede notar que la red clasifica el fruto con el hongo en fase 2 con una probabilidad del 100%, lo cual es correcto. La técnica de sensibilidad a la oclusión indica que, para esta etapa de la enfermedad, la red se activa con mayor intensidad en la zona negra sobre el hongo. Usando Grad-CAM las zonas de mayor activación están igualmente encima del hongo de la fruta. Con el método LIME se tiene que la red se enfoca en los alrededores del hongo, nuevamente con la parte superior derecha como la de mayor interés.

### Técnicas de visualización en cacaos con moniliasis fase 3:

Figura 4.29 Técnicas de visualización para cacaos fase 3. Red Googlenet.



Elaborado por: Jefferson Faconda y Edison Suquillo

En la figura 4.29 se observa que la red a clasificado la imagen correctamente como cacao con moniliasis en fase 3 con una probabilidad del 99%. La sensibilidad a la oclusión muestra que para esta fase de desarrollo del hongo la red se activa en la ubicación de la parte blanca del hongo. Con la técnica Grad-CAM la zona de activación para predecir la respuesta se enfoca más que nada en una hoja del suelo con tonalidad rojiza, algo no deseado. Usando LIME, las zonas más importantes de búsqueda para la red son la parte superior del fruto fuera del hongo y a la derecha del fruto en hojas que están en el suelo.

En la siguiente tabla se observa el tiempo que tarda la red en reconocer cada cacao clasificado:

Tabla 4.10. Tiempo para reconocimiento de cada cacao. PC1.

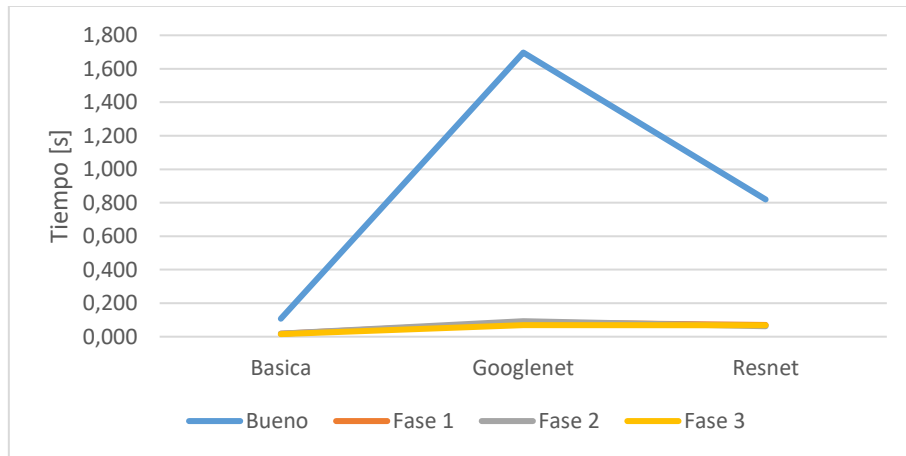
Red	PC	Tiempo de reconocimiento [s]			
		Bueno	Fase 1	Fase 2	Fase 3
Básica	1	0.325	0.012	0.029	0.010
Básica	2	0.233	0.016	0.016	0.016
Googlenet	1	1.635	0.035	0.061	0.054
Googlenet	2	1.818	0.069	0.045	0.051
Resnet	1	1.810	0.091	0.038	0.033
Resnet	2	0.892	0.069	0.044	0.046

Tabla de de tiempo para el reconocimiento del cacao con el hongo en sus diferentes etapas. Elaborado por:

Jefferson Faconda y Edison Suquillo



Figura 4.30 Tiempo para reconocimiento de cada cacao. PC1.



Elaborado por: Jefferson Faconda y Edison Suquillo

Se observa que las redes tienen respuestas de centésimas de segundo al evaluar los cacaos contaminados con el virus. No obstante, cuando se trata de cacaos sanos el tiempo de procesamiento de la imagen se eleva considerablemente en todas las redes. Los casos de la arquitectura Googlenet es la que se tarda más con un tiempo mayor al segundo y medio. Una de las causas es la complejidad de la red y el tamaño de la imagen. En caso de trabajar con redes más simples e imágenes más pequeñas el tiempo mejora.

#### 4.2.4 Resultados Obtenidos

Al realizar el análisis correspondiente de las diferentes redes neuronales se opta por utilizar la red neuronal de GoogLeNet.

Al cargar una imagen de un cacao a la interfaz diseñada, esta interfaz detecta en que estado se encuentra el cacao; si no tiene síntomas del hongo la interfaz refleja la etiqueta “Buena” y el led es de color “Verde” como se observa en la figura 4.31, si se encuentra en fase 1 la interfaz refleja la etiqueta “Fase 1” y el led es de color “Amarillo” como se observa en la figura 4.32, si se encuentra en fase 2 la interfaz refleja la etiqueta “Fase 2” y el led es de color “Magenta” como se observa en la figura 4.33, si se encuentra en fase 3 la interfaz refleja la etiqueta “Fase 3” y el led es de color “Rojo” como se observa en la figura 4.34 y en caso de cargar alguna otra imagen diferente a un cacao la interfaz refleja el mensaje “NO ES UN CACAO” y el led es de color blanco como se puede observar en la figura 4.35.

Figura 4.31 Resultado del Cacao sin el hongo Moniliasis.



Elaborado por: Jefferson Faconda y Edison Suquillo

Figura 4.32 Resultado del Cacao con el hongo Moniliasis en Fase 1.



Elaborado por: Jefferson Faconda y Edison Suquillo

Figura 4.33 Resultado del Cacao con el hongo Moniliasis en Fase 2.



Elaborado por: Jefferson Faconda y Edison Suquillo

Figura 4.34 Resultado del Cacao con el hongo Moniliasis en Fase 3.



Elaborado por: Jefferson Faconda y Edison Suquillo

Figura 4.35 Resultado al cargar una imagen diferente al cacao.



Elaborado por: Jefferson Faconda y Edison Suquillo

## CAPÍTULO 5

### CONCLUSIONES Y RECOMENDACIONES

#### 5.1 CONCLUSIONES

Se desarrolló un detector basado en un algoritmo mediante redes neuronales convolucionales, obteniendo en un entrenamiento de la red GoogLeNet un 96,87% de exactitud para la identificación del hongo Moniliasis, siendo este el mejor porcentaje de exactitud de todas las pruebas realizadas.

Las etapas del hongo Moniliasis fueron determinadas por el crecimiento de la mancha negro y el aumento del polvo blanco en la mazorca de cacao, definiendo así a tres etapas: Fase 1 (menor al 30% de su total), Fase 2 (mayor al 30% y menor al 90% de su total), Fase 3 (mayor al 90% de su total) las cuales fueron registradas mediante fotografías que fueron etiquetadas dependiendo la fase en la que se encontraba el hongo y se aumentó una etiqueta para el cacao bueno.

En el caso de este trabajo se simuló el detector utilizando redes neuronales convolucionales, la red básica creada desde cero obtuvo un porcentaje de 66,4% de exactitud de entrenamiento, la red GoogLeNet obtuvo un 96,87% de exactitud de entrenamiento y por último la red ResNet-18 obtuvo 95,31% de exactitud de entrenamiento, definiendo así que la red GoogLeNet es la mejor opción para este trabajo.

Se realizaron distintas pruebas con fotografías a diferentes distancias y de calidad de imagen, obteniendo como resultado que las imágenes de 48 Megapíxel tienen un mejor promedio de exactitud para las diferentes distancias.

Para este trabajo se probó con dos Laptop de distintas características, en la primera laptop la red GoogLeNet se demoró 49,11 segundos y en la segunda laptop tardó 91,53 segundos a una distancia de 35cm, definiendo así que el tiempo de procesamiento varía dependiendo del computador con el que se esté trabajando.

El algoritmo diseñado tuvo la capacidad de distinguir las tres etapas mediante la aplicación de filtros y procesamiento de imágenes que fueron establecidas al momento de entrenar la red.

## **5.2 RECOMENDACIONES**

Se recomienda para una posible implementación del sistema detector disponga la obtención de fotografías de manera automática con la ayuda de diferentes mecanismos como drones o robots, se debe considerar que las fotografías deben ser tomadas a 35cm para que el sistema sea óptimo.

Al utilizar redes neuronales convolucionales la base de datos debe contener el mayor número de fotografías o imágenes para que el sistema de detección sea más preciso.

## REFERENCIAS

- Aguirre Dobernack, N. (2018). Proyecto fin de carrera-Nicolás Aguirre Dobernack\_v2.pdf.
- Balón González, H. R., & Donoso Bruque, M. E. (2017). Con Mención en Gestión Empresarial Agropecuaria. 67. <http://repositorio.ucsg.edu.ec/bitstream/3317/5498/1/T-UCSG-PRE-TEC-AGRO-75.pdf>
- Celi, S., & Figueroa, D. (2017). Departamento de ciencias de la vida y de la agricultura. Diagnóstico del estado de la moniliasis del cacao (*Moniliophthora Roreri*) y mazorca negra (*Phytophthora Palmivora*), en el proyecto “Santo Cacao” del GAD provincial de Santo Domingo de los Tsáchilas, 68.
- de Ciencias, F., Dirección, A., Posgrado, D. E., Theobroma, (, Rendimiento, E. L., Mismo, D., La, E. N., Kallari, A., Onofre, I. M., & Solís, P. (2016). Universidad técnica de Ambato "El manejo fitosanitario del cultivo de cacao nacional.
- De, D., & Tsáchilas, L. (2017). Universidad Central del Ecuador facultad de ciencias agrícolas carrera de Ingeniería Agronómica propuesta de manejo integrado de la moniliasis (*Moniliophthora Roreri*) del cacao (*Theobroma Cacao*) en Santo.
- Diana, Estefanía, C. H. (2017). Universidad Central Del Ecuador. Universidad Central Del Ecuador, 91, 399–404.
- Edwin, P. V. (2019). Drones a tool for efficient agriculture: A high-tech future. *Idesia*, 37(1), 75–84. <https://doi.org/10.4067/S0718-34292019005000402>
- Enríquez, G. (2019). La moniliasis del cacao. 72, 1–98.
- Figueroa Yovanny. (2017). Universidad Estatal del Sur de Manabí facultad de ciencias Naturales y de la agricultura carrera de Ingeniería Agropecuaria trabajo de titulación modalidad proyecto de investigación previa la obtención del título de Ingeniero Agropecuario tema.
- Guillermo Romero-Macas, H. I. (2020). Algoritmo de reconocimiento de personas mediante procesamiento digital de imágenes usando MATLAB Algorithm of recognition of people by digital image processing using MATLAB Algoritmo de reconhecimento de pessoas por processamento de imagem digital usando . 5(08), 1241–1252. <https://doi.org/10.23857/pc.v5i8.1656>
- Hudson, M., Martin, B., Hagan, T., & Demuth, H. B. (2021). Deep Learning Toolbox™ Getting Started Guide. [www.mathworks.com](http://www.mathworks.com)
- INEC. (2021). Encuesta de superficie y producción agropecuaria continua. Instituto Nacional de Estadísticas y Censos, 23.
- M, S. I. U. C. (2006). Manual Básico De Matlab.
- Mathworks. (2020). ¿Qué es el reconocimiento de objetos? MathWorks. <https://es.mathworks.com/discovery/image-recognition-matlab.html>

- Mathworks. (2021a). GoogLeNet convolutional neural network - MATLAB googlenet - MathWorks España. <https://es.mathworks.com/help/deeplearning/ref/googlenet.html>
- MathWorks. (2021a). Machine Learning en MATLAB - MATLAB & Simulink - MathWorks España. <https://es.mathworks.com/help/stats/machine-learning-in-matlab.html>
- Mathworks. (2021b). Red neuronal convolucional ResNet-18 - MATLAB resnet18 - MathWorks América Latina. <https://la.mathworks.com/help/deeplearning/ref/resnet18.html>
- MathWorks. (2021b). Redes neuronales convolucionales - MATLAB & Simulink. <https://es.mathworks.com/discovery/convolutional-neural-network-matlab.html>
- Mathworks. (2021c). Transfer Learning - MATLAB & Simulink. [https://es.mathworks.com/discovery/transfer-learning.html?s\\_tid=srchtitle\\_learning%20transfer\\_1](https://es.mathworks.com/discovery/transfer-learning.html?s_tid=srchtitle_learning%20transfer_1)
- MATLAB & Simulink. (2021). ¿Qué es una red neuronal? - MATLAB & Simulink. <https://es.mathworks.com/discovery/neural-network.html>
- Montoya, E. A. Q., Colorado, S. F. J., Muñoz, W. Y. C., & Golondrino, G. E. C. (2017). Propuesta de una Arquitectura para Agricultura de Precisión Soportada en IoT. RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao, 24, 39–56. <https://doi.org/10.17013/risti.24.39-56>
- Moreno, A. (2019). Clasificación de imágenes usando redes neuronales convolucionales en Python. Universidad de Sevilla, 80.
- Planas de Martí, S. (2018). Agricultura De Precisión Y Protección De Cultivos. Revista de Ingeniería, 47, 10–19. <https://doi.org/10.16924/revinge.47.3>
- Sánchez, A., Vayas, T., Mayorga, F., & Freire, C. (2020). Sector Cacaotero Ecuatoriano. Observatorio Económico y Social de Tungurahua, 9–12.
- SENASA. (2017). CACAOTAL GUÍA. <http://www.iica.int>
- Tamura, H. (2019). Desarrollo de una aplicación de procesamiento de imágenes basada en técnicas de deep learning. Journal of Chemical Information and Modeling, 53(9), 287.

## ANEXO 1: RESULTADOS DE LOS ENTRENAMIENTOS DE LAS DISTINTAS REDES

Resultados de los entrenamientos de las diferentes redes con la herramienta de Matlab Deep Network Designer arroja automática las gráficas donde los valores en “X” son los números de iteraciones y “Y” es el porcentaje de exactitud del entrenamiento.

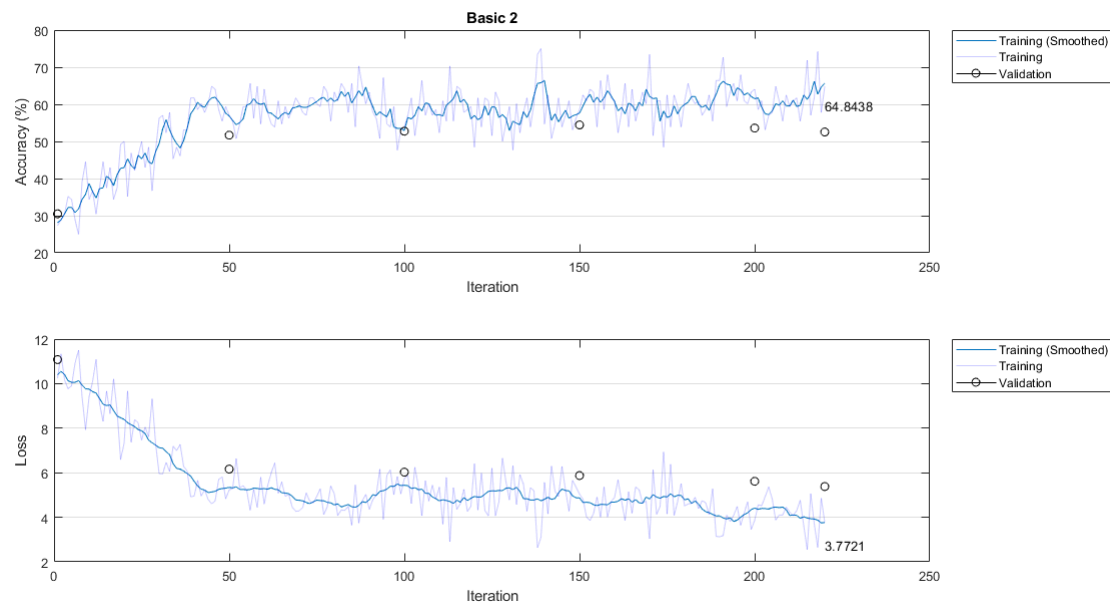
Entrenamiento de las diferentes redes con los siguientes parámetros:

Parámetro	Valor
Optimizador	Adam
Épocas	15
Tasa de aprendizaje	0.0001
Validación	Cada 50 iteraciones
Mezcla de datos	Cada época

Elaborado por: Jefferson Faconda y Edison Suquillo

### A. Básica

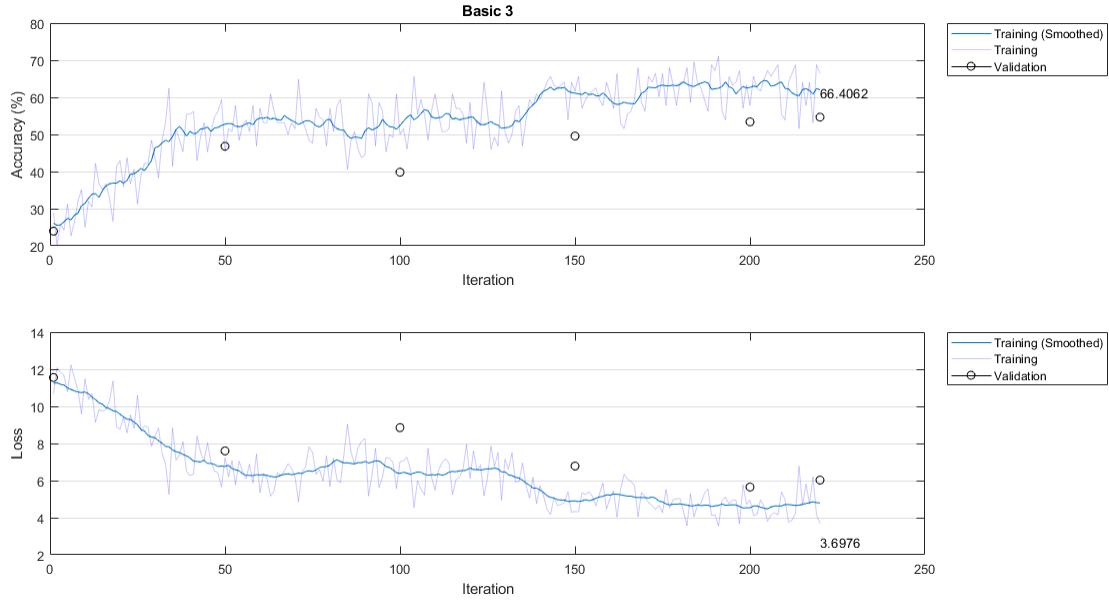
#### Prueba 2



Elaborado por: Jefferson Faconda y Edison Suquillo



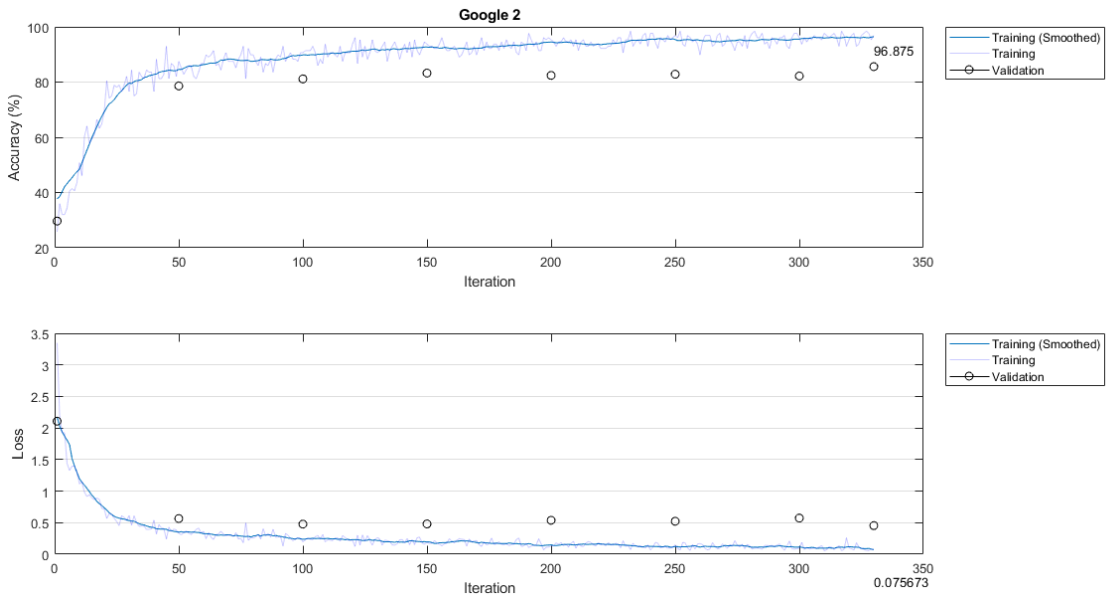
### Prueba 3



Elaborado por: Jefferson Faconda y Edison Suquillo

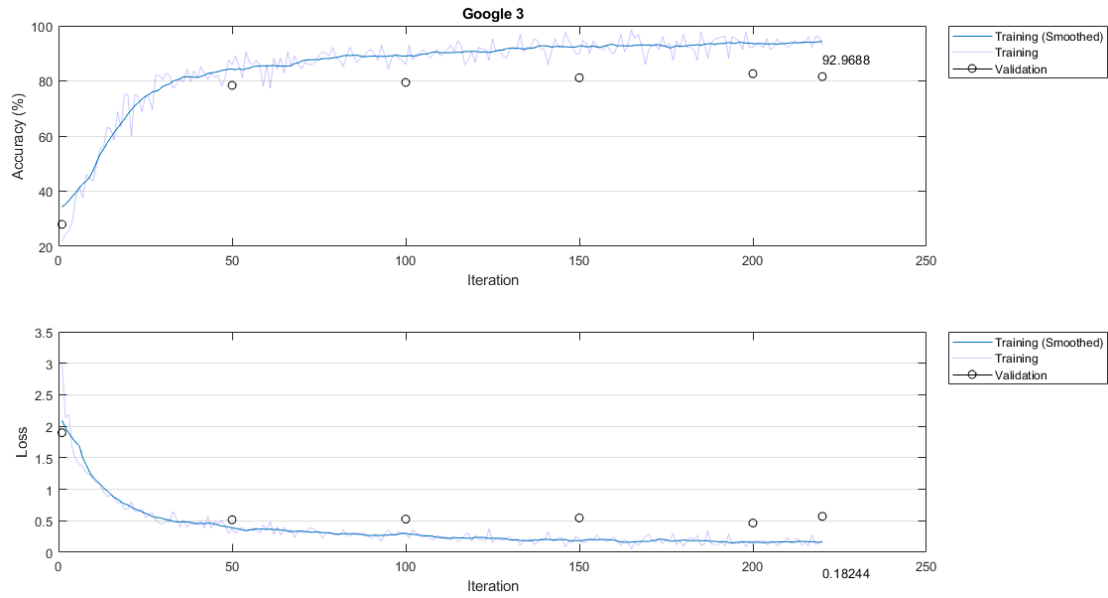
## B. GoogLeNet

### Prueba 2



Elaborado por: Jefferson Faconda y Edison Suquillo

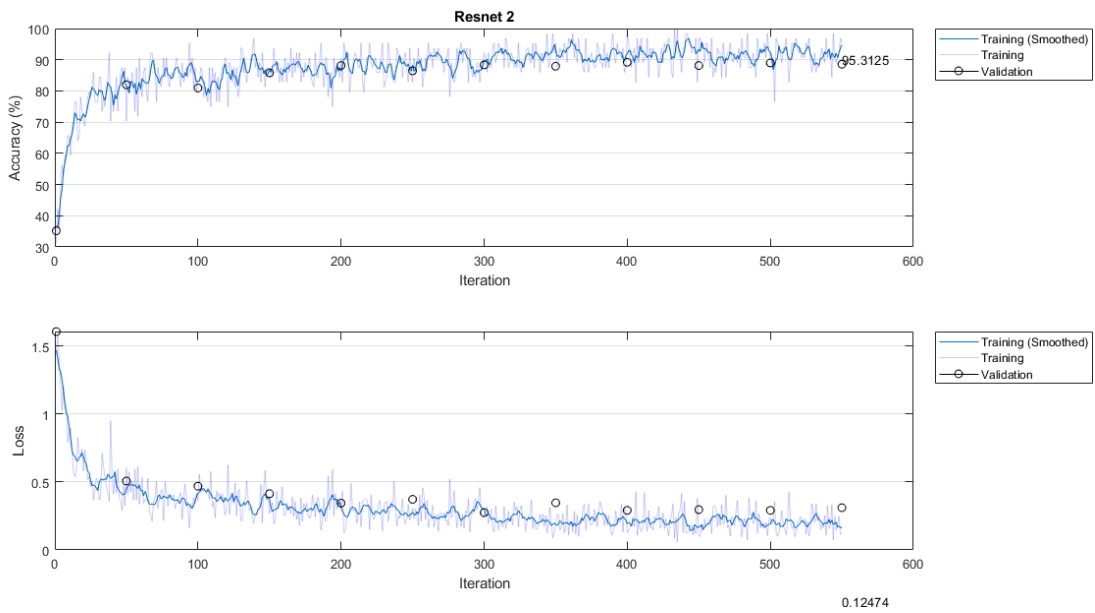
### Prueba 3



Elaborado por: Jefferson Faconda y Edison Suquillo

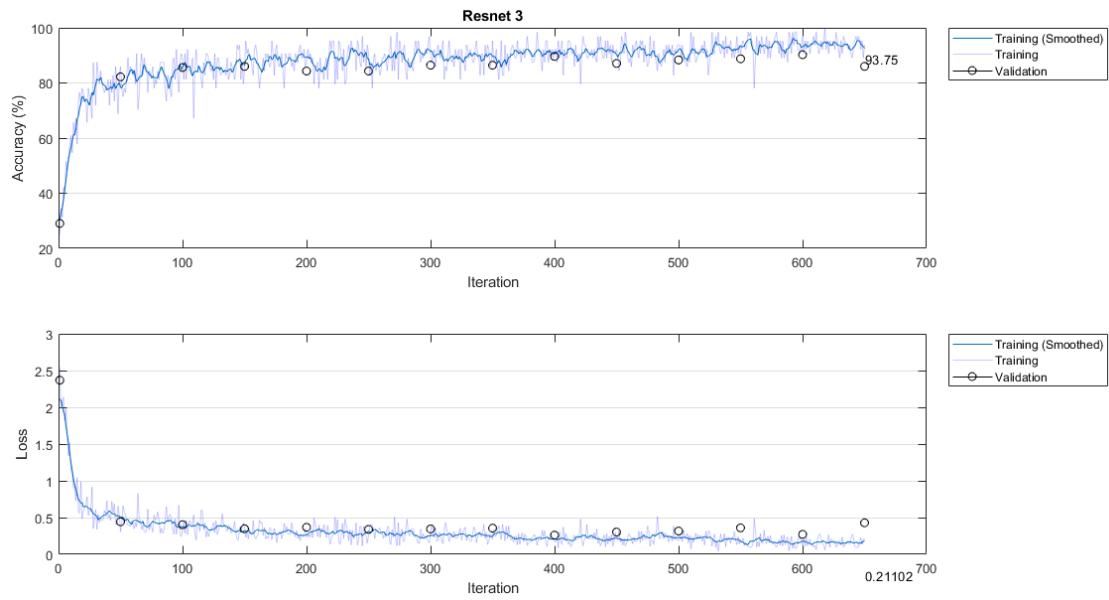
### C. ResNet-18

#### Prueba 2



Elaborado por: Jefferson Faconda y Edison Suquillo

### Prueba 3



Elaborado por: Jefferson Faconda y Edison Suquillo