

UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO  
FACULTAD DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA, INFORMÁTICA Y MECÁNICA  
ESCUELA PROFESIONAL DE INGENIERÍA INFORMÁTICA Y DE SISTEMAS



TESIS

---

**“CONSTRUCCIÓN DE UN PROTOTIPO DE SISTEMA  
PARA CLASIFICAR ENFERMEDADES EN LAS  
HOJAS DE CAFETO BASADO EN VISIÓN  
COMPUTACIONAL”**

---

**Tesis para optar al Título Profesional de:**  
INGENIERO INFORMÁTICO Y DE SISTEMAS

**Presentado por:**  
Br. ERWIN JHARIN PUCLLA ZEGARRA

**Asesor:**  
Dr. LAURO ENCISO RODAS

CUSCO - PERÚ  
2020

# Dedicatoria

*Dedicado principalmente a mis padres, familiares, compañeros y docentes quienes fueron parte de mi formación académica y vida profesional, especialmente a mi madre Elsa Zegarra Yanqui, que siempre estuvo guiandome en todo momento.*

# Agradecimientos

Se agradece por el apoyo brindado para cumplir mis objetivos a largo y corto plazo a:

Mis **padres y familiares** que siempre estuvieron motivando a cumplir mis metas y ser una persona de bien.

A mi asesor Lauro Enciso Rodas, por el apoyo brindado durante la elaboración de esta investigación.

# Resumen

Existen enfermedades y plagas que afectan al crecimiento del cafeto que son clasificadas haciendo uso de métodos tradicionales, manuales y visuales generando un margen de error en los resultados y tiempos de respuesta prolongados en los diagnósticos, en consecuencia, se tiene mayor índice de expansión de enfermedades en los cultivos de cafeto, mala calidad de granos y disminución en la producción del café. Sin embargo carecemos de una herramienta tecnológica eficaz y automática para la clasificación de enfermedades, por ello la necesidad de construir un prototipo de sistema de clasificación de enfermedades en las hojas del cafeto basado en visión computacional, y mostrar información técnica de las mismas para plantear un mejor control; las enfermedades afectan al fruto como a las hojas, en algunos casos se puede apreciar en la raíz de la planta, entre las enfermedades que dañan las hojas se tiene; *Leucoptera Coffeella*, *Mycena Citricolor*, *Hemileia Vastatrix*. Para contrarrestar las enfermedades de manera eficiente en tiempo prudente se construye un prototipo de sistema de clasificación de enfermedades constituido en 3 partes, primero, la construcción de un conjunto de datos (dataset de imágenes) de validación y entrenamiento formado por 1000 imágenes de enfermedades de interés, en segundo lugar, se diseña una arquitectura de red neuronal convolucional para la fase de entrenamiento y clasificación de la enfermedad, así mismo, se propone modelos de redes neuronales convolucionales ya construidos como el modelo VGG-16, AlexNet e InceptionV3 los cuales fueron entrenados en un entorno virtual denominado Google Colaboratory; la imagen de entrada a ser clasificada sufre transformaciones de preprocesamiento como técnicas de segmentación de imágenes, ecualización de imágenes y filtros de suavizado; finalmente se construye el prototipo del sistema haciendo uso del modelo ya entrenado y los módulos de segmentación de imágenes basado en color y umbralización para poder separar la región sana y enferma solo en caso de *Hemileia Vastatrix*, así mismo, se complementa con la información técnica de las enfermedades a tratar tales como biología, agente causal, daño y control. Desarrollado en entorno de escritorio, así el proceso de clasificación se realiza en tiempo real, sin conocimiento previo adquirido a través de la experiencia, menor margen de error y es diagnosticado inmediatamente por los especialistas o caficultores evitando la propagación masiva de la enfermedad.

## Palabras Clave

Redes neuronales convolucionales, clasificación de imágenes, segmentación de imágenes, *Hemileia Vastatrix*, *Mycena Citricolor*, *Leucoptera Coffeella*.

# Abstract

There are diseases and pests that affect the growth of the coffee tree that are classified using traditional, manual and visual methods, generating a margin of error in the results and prolonged response times in diagnoses, consequently, there is a higher rate of disease expansion in coffee crops, poor quality of grains and a decrease in coffee production. However, we lack an effective and automatic technological tool for the classification of diseases, for this reason the need to build a prototype of a system for classifying diseases in coffee leaves based on computer vision, and to show technical information about them to propose a better control; Diseases affect both fruit and leaves, in some cases it can be seen in the root of the plant, among the diseases that damage the leaves is; *Leucoptera Coffeella*, *Mycena Citricolor*, *Hemileia Vastatrix*. To counter diseases efficiently in a prudent time, a prototype of a disease classification system is built, consisting of 3 parts, first, the construction of a dataset (image dataset), devalidation and training, formed by 1000 images of diseases of Of interest, secondly, a convolutional neural network architecture is designed for the training and classification phase of the disease, likewise, convolutional neural network models already built are proposed, such as the VGG-16, AlexNet and InceptionV3 models, which were trained in a virtual environment called Google Colaboratory; the input image to be classified undergoes preprocessing transformations such as image segmentation techniques, image equalization and smoothing filters; Finally, the prototype of the system is built using the convolutional neural network model already trained and the image segmentation modules based on color and thresholding to be able to separate the healthy and diseased region only in the case of *Hemileia Vastatrix*, likewise, it is complemented with the technical information on the diseases to be treated such as biology, causal agent, damage and control. Developed in a desktop environment, so the classification process is carried out in real time, without prior knowledge acquired through experience, less margin for error and is immediately diagnosed by specialists or coffee growers, avoiding the massive spread of the disease.

## Keywords

Convolutional neural networks, Image classification, image segmentation, *Leucoptera Coffeella*, *Mycena Citricolor* and *Hemileia Vastratrix*.

# Índice general

Índice de figuras	VII
Índice de Tablas	IX
Términos y abreviaciones	X
<b>1. Aspectos generales</b>	<b>1</b>
1.1. Problema de investigación . . . . .	1
1.1.1. Descripción del problema . . . . .	1
1.1.2. Formulación del problema . . . . .	2
1.2. Antecedentes . . . . .	2
1.3. Justificación . . . . .	7
1.4. Objetivos . . . . .	7
1.4.1. Objetivo general . . . . .	7
1.4.2. Objetivo específico . . . . .	7
1.5. Alcances y limitaciones . . . . .	8
1.5.1. Alcances . . . . .	8
1.5.2. Limitaciones . . . . .	8
1.6. Metodología . . . . .	9
1.7. Contribuciones . . . . .	9
1.8. Cronograma . . . . .	10
<b>2. Marco conceptual</b>	<b>11</b>
2.1. El Café . . . . .	11
2.1.1. Hojas del café . . . . .	12
2.1.2. Botánica del café . . . . .	12
2.1.3. Clasificación taxonómica de las enfermedades de café . . . . .	13
2.1.4. Enfermedades en las hojas de café . . . . .	13
2.2. Visión computacional . . . . .	19
2.2.1. Procesamiento de imágenes . . . . .	19
2.2.2. Técnicas de segmentación . . . . .	24
2.3. Aprendizaje máquina . . . . .	26
2.3.1. Aprendizaje supervisado . . . . .	27
2.3.2. Redes neuronales convolucionales . . . . .	28
2.3.3. Evaluación . . . . .	33
2.4. Propuesta de arquitecturas de redes neuronales convolucionales . . . . .	35
2.4.1. InceptionV3 . . . . .	35
2.4.2. AlexNet . . . . .	36

2.4.3. VGG-16 . . . . .	38
<b>3. Construcción del prototipo</b>	<b>40</b>
3.1. Fases de la construcción . . . . .	42
3.1.1. Preprocesamiento de la imagen . . . . .	43
3.1.2. Clasificación de la imagen . . . . .	48
3.1.3. Información técnica . . . . .	63
3.1.4. Segmentación de la imagen y determinación del grado de severidad caso Hemileia Vastatrix . . . . .	64
<b>4. Análisis de resultados y discusiones</b>	<b>68</b>
4.1. Análisis de resultados . . . . .	68
4.1.1. Comparativa de arquitecturas de redes neuronales convo- lucionales en clasificación de enfermedades en las hojas de cafeto . . . . .	68
4.1.2. Resultados de segmentación, caso Hemileia Vastatrix . . . . .	71
4.2. Discusión de resultados . . . . .	72
4.3. Detalles técnicos . . . . .	73
4.3.1. Recursos hardware . . . . .	73
4.3.2. Recursos software . . . . .	74
<b>Conclusiones</b>	<b>75</b>
<b>Recomendaciones</b>	<b>76</b>
<b>ANEXOS</b>	<b>77</b>
<b>A. Informe del sistema antiplagio Turnitin</b>	<b>78</b>
<b>B. Set de datos de entrenamiento y validación</b>	<b>82</b>

# Índice de figuras

1.1. Clasificación manual de la enfermedad en hojas del cafeto . . . . .	2
1.2. Enfermedades que dañan las hojas del cafeto. . . . .	2
1.3. Modelo de red neuronal convolucional VGG-16 . . . . .	4
1.4. Modelo de red neuronal convolucional VGG-19 . . . . .	5
2.1. Estructura de un grano de café . . . . .	11
2.2. Estructura de una hoja de cafeto . . . . .	12
2.3. Fases Hemileia Vastatrix. . . . .	14
2.4. Grado de severidad en la hoja de café . . . . .	14
2.5. Estado biológico de la Leucoptera Coffeella . . . . .	16
2.6. Mycena Citricolor . . . . .	17
2.7. Ondas de color . . . . .	20
2.8. Histograma, Minador de café. . . . .	21
2.9. Filtro con promedio de vecinos. . . . .	23
2.10. Aplicación de la mediana. . . . .	23
2.11. Filtro de la mediana. . . . .	24
2.12. Representación gráfica del filtro gaussiano. . . . .	25
2.13. Ejemplo de umbralización . . . . .	26
2.14. Esquema de machine learning . . . . .	27
2.15. Neurona artificial . . . . .	29
2.16. Red neuronal bajo el esquema Feedforward . . . . .	30
2.17. Arquitectura referencial del una red neuronal convolucional . . . . .	30
2.18. Transformación de convolución en una neurona. . . . .	31
2.19. Aplicación de una convolución y Relu. . . . .	31
2.20. Max pooling y average pooling . . . . .	32
2.21. Aplicación de Softmax . . . . .	33
2.22. Matriz de confusión en un entorno multiclase . . . . .	34
2.23. Pérdida de entropía cruzada categórica . . . . .	35
2.24. Módulo de la arquitectura Inception original. . . . .	36
2.25. Módulo de Inception donde el filtro de la convolución 5x5 es reemplazado por dos filtros de convolución de 3x3 . . . . .	36
2.26. Modelo Alexnet . . . . .	37
2.27. Modelo VGG-16 . . . . .	38
2.28. Arquitectura VGG-16 . . . . .	39
3.1. Diagrama de flujo, fases de la construcción . . . . .	42
3.2. Preprocesamiento de la imagen . . . . .	43
3.3. Aplicación de ecualización del histograma. . . . .	44
3.4. Filtro de la mediana . . . . .	45



3.5.	Aplicación del filtro de la mediana . . . . .	46
3.6.	Aplicación del filtro de Gauss con desviación de 9 . . . . .	47
3.7.	Aplicación del filtro de Gauss con desviación de 27 . . . . .	47
3.8.	Clasificación y predicción . . . . .	48
3.9.	Fases de la clasificación . . . . .	49
3.10.	Metodología general para construir el modelo de red neuronal . . . . .	52
3.11.	Modelo de red neuronal convolucional . . . . .	55
3.12.	Modelo de red neuronal convolucional . . . . .	56
3.13.	Visualización de mapa de características convolución 1 . . . . .	57
3.14.	Visualización de mapa de características convolución 2 . . . . .	57
3.15.	Visualización de mapas de funciones o mapas de activación generados por la CNN . . . . .	58
3.16.	Visualización de los mapas de características VGG-16 . . . . .	59
3.17.	Visualización de los mapas de características VGG-16 . . . . .	60
3.18.	Visualización de los mapas de características InceptionV3 . . . . .	62
3.19.	Información técnica correspondiente a la enfermedad clasificada . . . . .	63
3.20.	Región enferma . . . . .	64
3.21.	Región sana . . . . .	65
3.22.	Resultado del prototipo . . . . .	67
4.1.	Entrenamiento del modelo CNN diseñado . . . . .	69
4.2.	Entrenamiento del modelo AlexNet . . . . .	69
4.3.	Entrenamiento del modelo VGG-16 . . . . .	70
4.4.	Entrenamiento del modelo Inception V3 . . . . .	70
4.5.	Comparación de resultados . . . . .	71
4.6.	Resultados, cálculo del grado de severidad en la hoja . . . . .	72
A.1.	Reporte de similitud parte I . . . . .	79
A.2.	Reporte de similitud parte II . . . . .	80
A.3.	Reporte de similitud parte III . . . . .	81
B.1.	Dataset de imágenes de entrenamiento . . . . .	83
B.2.	Dataset de entrenamiento Leucoptera Cofeella y Mycena Citricolor . . . . .	84
B.3.	Dataset de validación Hemileia Vastatrix . . . . .	85
B.4.	Dataset de validación Leucoptera Cofeella y Mycena Citricolor . . . . .	86

# Índice de Tablas

2.1. Clasificación de enfermedades en el café . . . . .	11
2.2. Grado de severidad de la roya de café . . . . .	14
2.3. Información técnica de enfermedades en hojas de cafeto. . . . .	18
2.4. Filtro gaussiano con $s=1$ y matriz de filtrado identidad. . . . .	24
2.5. Características de una red neuronal artificial . . . . .	29
2.6. Ejemplo de probabilidades por clase para la capa Softmax . . . . .	33
3.1. Procedimiento para la construcción del prototipo . . . . .	41
3.2. Descripción del modelo de CNN . . . . .	54
4.1. Resultados de evaluar arquitecturas de redes neuronales convolu- cionales . . . . .	69

# Términos y abreviaciones

**Binarización.-** Reducción de información donde los únicos valores posibles son verdadero o falso, o valores equivalentes a dos parámetros.

**Botánica.-** Área de la biología que estudia las plantas especialmente las características de clasificación, descripción, distribución, identificación así como fisiología y relaciones con otros seres vivos.

**Cafeto.-** Arbusto donde se alojan los granos de café.

**Caficultor.-** Persona que se dedica al cultivo de café.

**Clasificación.-** Búsqueda en un conjunto de datos de un determinado elemento que guarde una semejanza al dato de interés.

**CNN** ( Convolutional Neuronal Network.)

**Coffea Arábica.-** Nombre científico del café.

**Convolución.-** Proceso de transformar dos funciones en una resultante a través de operadores matemáticos.

**CPU** (Unidad de central de procesamiento)

**Densidad de probabilidad.-** Función que caracteriza el comportamiento probable de una población, matemáticamente como la derivada de la función de distribución.

**Dominio espacial.-** Conjunto de puntos que componen una imagen.

**Ecuilización.-** Transformación que mejora el contraste de una imagen.

**Endospermo.-** Tejido de las plantas que sirve de alimento para otros organismos.

**Enfermedad.-** Proceso de alterar el funcionamiento de un ser vivo u organismo.

**Epidermis.-** Membrana epitelial del las hojas formada por una capa de células de clorofila.

**Esporas.-** Célula vegetal reproductora que no necesita ser fecundada.

**Feedforward.-** Esquema de conexión donde la información se dirige de la entrada hacia la salida.

**Función de activación.-** Es la salida de un nodo, dada una entrada o un conjunto de entradas.

**Google Colaboratory.-** Herramienta de Google que nos ofrece máquinas virtuales con CPU y GPU.

**Hemileia Vastatrix.-** Hongo perteneciente al orden Pucciniaceae, causante de la roya en las hojas de café.

**Histograma.-** Representación gráfica de un conjunto de variables que indican frecuencia de los valores representados.

**ICO** ( International Organization Coffee.)

**IID** ( Índice de intensidad de daño.)

**III** ( Índice de intensidad de infestación.)

**Incidencia.-** Interferencia que repercute al correcto funcionamiento alterándolo o interrumpiéndolo.

**Infestación.-** Invadir o llenar por completo un lugar, en este caso la región de la hoja, acción de infestarse.

**Kernel.-** Es una matriz reducida que se utiliza para desenfoque, enfoque, realce, detección de bordes, etc.

**Leucoptera Coffella.-** Llamado minador de café, es un microlépidoptero bisexual que daña los cultivos de café.

**Máquina de vector de soporte.-** Técnica de aprendizaje automático utilizado para clasificación o regresión.

**Monotónico.-** Función o cantidad que varía de tal forma que o bien nunca aumenta o bien nunca disminuye.

**Mycena Citricolor.-** Hongo constituido por hospederos desde malezas hasta plantas productivas perteneciente a la familia Tricholomataceae.

**NET.-** Conjunto de fuerzas sobre una neurona.

**Neurona artificial.-** Conjunto de unidades conectadas entre sí para transmitir señales.

**Plaga.-** Conjunto de organismos animales o vegetales que destruyen cultivos y las plantas.

**Pooling.-** Reducción de muestreo que ocurre en una red neuronal.

**Prototipo.-** Modelo del comportamiento que tendrá un sistema informático, usado para entender ciertos aspectos.

**Puccineaceae.-** Especie de Roya que ataca la hoja de plantas, con mayor número de hongos.

**RELU** ( Unidad Neuronal Rectificada.)

**Rubiáceo.-** Familia de plantas dicotiledóneas que consta de árboles, arbustos, hiervas con hojas, flores y frutos.

**Segmentación.-** Partición de un todo en regiones de interés.

**Severidad.-** Exactitud y rigor para el cumplimiento de reglas.

**Umbral.-** Límite al que se atribuye ciertas reglas.

# Capítulo 1

## Aspectos generales

### 1.1. Problema de investigación

#### 1.1.1. Descripción del problema

Las enfermedades en las plantas del cafeto generalmente dañan las hojas que son parte fundamental y que contiene el pigmento llamado clorofila, que a través de ella absorbe energía y realiza el proceso de fotosíntesis para garantizar la salubridad y el fortalecimiento de la planta<sup>1</sup>. En los cultivos de café los asistentes, especialistas y caficultores con conocimientos empíricos en temas relacionados al manejo agronómico del café aplican procesos de clasificación manual, visual y tradicional de las enfermedades en base a experiencias obtenidas con el pasar de los tiempos, ocasionando inconvenientes durante el proceso de diagnóstico de la enfermedad y a su vez incrementar el índice de expansión de las enfermedades. Por ejemplo en caso de *Hemileia Vastatrix*, una vez clasificada la enfermedad se requiere determinar el grado de severidad en la hoja de cafeto lo que conlleva a cálculos matemáticos para determinar el porcentaje de área infectada con respecto al área foliar de la hoja el cual no puede realizarse en tiempo real. A consecuencia de usar métodos tradicionales de clasificación se puede obtener diagnósticos ineficientes e inoportunos de las enfermedades, que da origen a la defoliación de hojas y muerte de las plantas afectadas disminuyendo la cantidad y calidad de producción ocasionando la reducción de precios en el mercado local y nacional afectando la economía de los caficultores. Para la clasificación automática de la enfermedad se precisa de una imagen que puede presentar características como; bajo contraste, ruido en la imagen, así como semejanza de enfermedades caso *Leucoptera Coffeella* y *Hemileia Vastatrix* en su fase avanzada; lo que genera cierta problemática en la fase de clasificación. Revisando trabajos de investigación relacionados se puede identificar que no se cuenta con la disponibilidad de una herramienta no convencional que clasifique las enfermedades en las hojas del cafeto de manera automática sin tener un conocimiento previo, y a su vez facilitar información técnica de la enfermedad para un diagnóstico pertinente por los especialistas.

---

<sup>1</sup>ver en: <http://lasplantasricardo.blogspot.com/2008/12/yes.html>

Figura 1.1: Clasificación manual de la enfermedad en hojas del cafeto



Fuente: Propia

Figura 1.2: Enfermedades que dañan las hojas del cafeto.



(a) *Hemileia Vastatrix*

(b) *Mycena Citricolor*

(c) *Leucoptera Coffella*

Fuente: Propia

### 1.1.2. Formulación del problema

¿Cómo construir un prototipo de sistema basado en visión computacional, para automatizar el proceso de clasificación de enfermedades en hojas de cafeto?

## 1.2. Antecedentes

1. “Diagnóstico de incidencia y severidad de Roya Amarilla en cultivares de café en el Sector de Mandor, Distrito de Maranura, Provincia La Convención” (Mantupa Wilbert, 2016), publicado en el repositorio de la Universidad Nacional de San Antonio Abad de Cusco, Perú; que consiste en la evaluación de incidencia en la Roya de Café y el cálculo del grado de severidad como pilar fundamental para obtener datos relevantes tanto para el diagnóstico, así como datos de la parcela, ubicación geográfica y otros. El proceso de evaluación se obtuvo acerca de algunas variedades de cafetos.

- Para la evaluación se seleccionó 10 plantas, para el cálculo de severidad a nivel de hoja, proceso realizado visualmente.
- Posterior a ello se tomaron datos para su análisis.

- Se concluye que la variedad Typica tiene un 96.8% de incidencia, Caturra 77.1%, Catimor un 20.5%. Con respecto a la severidad Typica tiene un grado de severidad de 55.18% equivalente al nivel 4, Caturra se encuentra con incidencia de 24.64% perteneciente al grado 3, Catimor tiene una incidencia de 4.2% perteneciente al grado 1.
- Por lo tanto se concluye que las dos primeras variedades tales como Typica y Caturra son más susceptibles a la Roya Amarilla y en caso no se realice un diagnóstico temprano incrementa el índice de expansión en los cultivos.

Este trabajo de investigación está más ligado a determinar los grados de incidencia y severidad en las hojas de café, más no a proporcionar información pertinente de acuerdo a los cálculos obtenidos para un diagnóstico adecuado; además el cálculo del grado de severidad en las hojas del café se realizó de manera visual, manual y tradicional, lo que genera tiempos de respuesta prolongados y mientras tanto incrementa el índice de expansión de la enfermedad y para evitar podrían usar métodos automatizados para determinar el grado de severidad en la hoja.

2. “Evaluación de técnicas de aprendizaje de máquina para la identificación de imágenes de edificios históricos de la ciudad del Cusco basado en bag-of-words y redes neuronales convolucionales” (David Escobedo, 2017), publicado en el repositorio institucional de la Universidad Nacional de San Antonio Abad del Cusco, Perú; haciendo uso de descriptors como SIFT (Scale Invariant Features Transform) y SURF (Speeded Up Robust Features) para la extracción de características y para clasificación se aplican técnicas como Support Vector Machine (SVM), K-Vecinos próximos (KNN) y redes neuronales convolucionales con función de activación Softmax en la capa de salida para generar la probabilidad de correspondencia de la imagen de entrada, este último utiliza optimizadores de ADAM para reducir el costo y función de costo de entropía cruzada ligados a los resultados de la métrica exactitud. Para el entrenamiento de la Red Neuronal Convolucional se usó el modelo de Red VGG-16 y VGG-19, constituido por los parámetros que se muestran a continuación, **figura 1.3** y **figura 1.4**.

Se concluye que los mejores resultados obtenidos del proceso de construcción del modelo y entrenamiento corresponden a la técnica de aprendizaje Support Vector Machine obteniendo una exactitud de 91.83%, las redes neuronales convolucionales basada en Bag-of-words obtiene una exactitud de 66.25%; por lo tanto la técnica de aprendizaje Support Vector Machine basada en redes neuronales convolucionales (InceptionV3) es la más óptima.

Tomamos como referencia la distribución del dataset de imágenes en datos de entrenamiento y validación en relación a porcentajes usados 76.6% y 23.3% respectivamente, parece ser la correcta ya que se obtienen resultados óptimos durante el entrenamiento del modelo de aprendizaje máquina en problemas de clasificación de imágenes y a su vez el modelo InceptionV3 se adecúa pese a la complejidad de tener ramas paralelas y mejora los resultados a mayor entrenamiento.



Figura 1.3: Modelo de red neuronal convolucional VGG-16

<b>Dimensión de la matriz</b>	<b>Operación</b>	<b>Detalles</b>
[224x224x3]	Imagen de Entrada	Image resize
[224x224x64]	<b>Convolución 1</b>	64 FM, 3x3 FCONV, Stride 1
[224x224x64]	<b>Convolución 2</b>	64 FM, 3x3 FCONV, Stride 1
[112x112x64]	<b>Max Pooling 1</b>	2x2 FPOL, Stride 2
[112x112x128]	<b>Convolución 3</b>	128 FM, 3x3 FCONV, Stride 1
[112x112x128]	<b>Convolución 4</b>	128 FM, 3x3 FCONV, Stride 1
[56x56x128]	<b>Max Pooling 2</b>	2x2 FPOL, Stride 2
[56x56x256]	<b>Convolución 5</b>	256 FM, 3x3 FCONV, Stride 1
[56x56x256]	<b>Convolución 6</b>	256 FM, 3x3 FCONV, Stride 1
[56x56x256]	<b>Convolución 7</b>	256 FM, 3x3 FCONV, Stride 1
[28x28x256]	<b>Max Pooling 3</b>	2x2 FPOL, Stride 2
[28x28x512]	<b>Convolución 8</b>	512 FM, 3x3 FCONV, Stride 1
[28x28x512]	<b>Convolución 9</b>	512 FM, 3x3 FCONV, Stride 1
[28x28x512]	<b>Convolución 10</b>	512 FM, 3x3 FCONV, Stride 1
[14x14x512]	<b>Max Pooling 4</b>	2x2 FPOL, Stride 2
[14x14x512]	<b>Convolución 11</b>	512 FM, 5x5 FCONV, Stride 1
[14x14x512]	<b>Convolución 12</b>	512 FM, 5x5 FCONV, Stride 1
[14x14x512]	<b>Convolución 13</b>	512 FM, 5x5 FCONV, Stride 1
[7x7x512]	<b>Max Pooling 5</b>	2x2 FPOL, Stride 2
[1x1x4096]	<b>Fully-Connected 1</b>	4096 Neuronas

Fuente:David Escobedo (2017)

Figura 1.4: Modelo de red neuronal convolucional VGG-19

Dimensión de la matriz	Operación	Detalles
[224x224x3]	Imagen de Entrada	Image resize
[224x224x64]	<b>Convolución 1</b>	64 FM, 3x3 FCONV, Stride 1
[224x224x64]	<b>Convolución 2</b>	64 FM, 3x3 FCONV, Stride 1
[112x112x64]	<b>Max Pooling 1</b>	2x2 FPOL, Stride 2
[112x112x128]	<b>Convolución 3</b>	128 FM, 3x3 FCONV, Stride 1
[112x112x128]	<b>Convolución 4</b>	128 FM, 3x3 FCONV, Stride 1
[56x56x128]	<b>Max Pooling 2</b>	2x2 FPOL, Stride 2
[56x56x256]	<b>Convolución 5</b>	256 FM, 3x3 FCONV, Stride 1
[56x56x256]	<b>Convolución 6</b>	256 FM, 3x3 FCONV, Stride 1
[56x56x256]	<b>Convolución 7</b>	256 FM, 3x3 FCONV, Stride 1
[56x56x256]	<b>Convolución 8</b>	256 FM, 3x3 FCONV, Stride 1
[28x28x256]	<b>Max Pooling 3</b>	2x2 FPOL, Stride 2
[28x28x512]	<b>Convolución 9</b>	512 FM, 3x3 FCONV, Stride 1
[28x28x512]	<b>Convolución 10</b>	512 FM, 3x3 FCONV, Stride 1
[28x28x512]	<b>Convolución 11</b>	512 FM, 3x3 FCONV, Stride 1
[28x28x512]	<b>Convolución 12</b>	512 FM, 3x3 FCONV, Stride 1
[14x14x512]	<b>Max Pooling 4</b>	2x2 FPOL, Stride 2
[14x14x512]	<b>Convolución 13</b>	512 FM, 5x5 FCONV, Stride 1
[14x14x512]	<b>Convolución 14</b>	512 FM, 5x5 FCONV, Stride 1
[14x14x512]	<b>Convolución 15</b>	512 FM, 5x5 FCONV, Stride 1
[14x14x512]	<b>Convolución 16</b>	512 FM, 5x5 FCONV, Stride 1
[7x7x512]	<b>Max Pooling 5</b>	2x2 FPOL, Stride 2
[1x1x4096]	<b>Fully-Connected 1</b>	4096 Neuronas

Fuente: David Escobedo (2017)

3. “Diagnóstico automático de la Roya de Café Amarilla aplicando técnicas de procesamiento de imágenes y aprendizaje máquina” (Arrasco Ordoñez, 2018), trabajo de investigación desarrollado y publicado en la Pontificia Universidad Católica del Perú; consiste en identificar la enfermedad de la roya de café que daña exclusivamente la hoja, el proceso de investigación consta en obtener una imagen de la hoja ya sea con la enfermedad o sin la enfermedad, posterior a ello se hace la extracción de características utilizando patrones binarios locales y descriptores de textura Haralick, para así determinar la región afectada por la enfermedad, de esta manera aislar tanto el área sin enfermedad y el área con enfermedad, para su fase de clasificación o identificación de la enfermedad se utiliza Support Vector Machine. Una vez que se logra clasificar la enfermedad, se procede con la determinación de la severidad en las hojas del café, por ejemplo, si se desea conocer el grado de severidad en una hoja esta vendría a ser el área de afección de la enfermedad

sobre el área total de la hoja sin enfermedad, este último proceso fue basado en segmentación de imágenes y conteo de píxeles, cabe destacar que este proceso se realiza en varias hojas de la planta para realizar un diagnóstico correcto.

Se realizó el proceso de entrenamiento haciendo uso de 10 modelos de aprendizaje máquina basados en Support Vector Machine, de los cuales 4 de los 10 modelos obtienen exactitud superior al 80 %, los 2 primeros modelos obtienen 87.50 % y 95.53 % cuyo resultado puede ser engañoso porque solo considera la variación en la intensidad de los píxeles en los canales de color RGB.

Se enfoca en la Roya Amarilla y su diagnóstico optando una clasificación binaria limitando el proceso de identificación de más enfermedades, cuando existen varias enfermedades que afectan las hojas y son perjudiciales para los cultivos; se pudo haber mejorado los resultados obtenidos haciendo uso de una red neuronal convolucional híbrida donde la capa clasificación estaría compuesta por Support Vector Machine de tipo multiclase de esta manera evitar los resultados engañosos debido a la variación de intensidad de píxeles.

4. “Clasificador de hojas mediante Deep Learning” (Galarraga, 2017), desarrollado y publicado en el repositorio de la Universidad Politécnica de Madrid, España; considerando que las plantas son importantes para el ser humano, a consecuencia de plagas son afectadas por muchas enfermedades, el siguiente trabajo de investigación consiste en diseñar, desarrollar y evaluar un clasificador de enfermedades basado en aprendizaje profundo, trabajando con enfermedades en las uvas y enfermedades en los tomates. Este proceso tiene 3 etapas, la primera consiste en segmentar la imagen extrayendo y aislando regiones de interés en la imagen, la segunda etapa consiste en buscar los mejores parámetros para entrenar la arquitectura de red neuronal.

Las tareas que realiza son:

- Proporciona información de la clase de hoja, si este es perteneciente a una hoja de uva o tomate.
- Proporciona información del estado de la hoja, la enfermedad, la causa, síntomas y sugerencias.
- **Enfermedades en la uva.-** Podredumbre Negra (Black rot), Yesca, Mancha.
- **Enfermedades en el tomate.-** Mancha Bacteriana, Tizón temprano, Tizón tardío, mancha negra, mancha de la hoja, Araña Roja, Rizado amarillo de tomate.

Se obtiene un rendimiento de clasificación de 98.37 % concluyendo que los modelos de aprendizaje AlexNet son buenos para la extracción de características ahorrando el tiempo de ejecución durante el entrenamiento, obteniendo un modelo capaz de clasificar correctamente 2021 imágenes de un total de 2053 ejemplos preclasificados por un experto.

El clasificador define si la enfermedad corresponde a una uva o un tomate y nos muestra las enfermedades de esa clase, más no específicamente la enfermedad en particular para su posterior tratamiento; la herramienta de

simulación no muestra el control asociado a cada enfermedad clasificada, sería más completo en caso fuese así, el uso de gran cantidad de datos de entrenamiento y validación (más de 10000 imágenes) hace más eficiente al modelo AlexNet evitando complicaciones relacionados a la complejidad de la arquitectura durante el entrenamiento.

### 1.3. Justificación

El café es un recurso natural con mayor índice de consumo a nivel mundial, la producción de este recurso en nuestro país es alto ocupando el undécimo lugar produciendo 192000 kilogramos de café equivalente al 2.23 % de la producción mundial según International Organization Coffee (ICO) <sup>2</sup>, información recopilada a abril del 2019. Reemplazar los métodos tradicionales de clasificación de enfermedades por un prototipo de sistema de clasificación automática sirve para realizar un diagnóstico oportuno y rápido evitando la propagación masiva de plagas y enfermedades con el apoyo de una herramienta tecnológica que consiste en clasificar la enfermedad sin conocimiento previo y brindar información técnica para su posterior control y manejo agronómico sin contar con la presencia obligatoria de un especialista; la optimización de los tiempos de clasificación y diagnóstico de las enfermedades sin tener conocimientos avanzados obtenidos a través de la experiencia ofrece facilidades a toda la comunidad de caficultores para que puedan controlar las enfermedades y plagas que afectan las hojas de los cultivos evitando pérdidas económicas garantizando la calidad y cantidad de granos durante la producción del café, de esta manera resolvemos el problema de la clasificación manual y tradicional de las enfermedades que conllevan a diagnósticos tardíos con un margen de error y limitados a ser aplicados solo por especialistas; así mismo el prototipo de sistema de clasificación de enfermedades en las hojas podría ser aplicado a otro rubro como enfermedades en hojas de la palta, tomates, rocotos o todo relacionado a cultivos agrícolas, es factible ampliar el dataset de imágenes con los datos requeridos (imágenes de hojas enfermas de palta, tomates, rocotos y otros) e iniciar nuevamente el proceso de entrenamiento haciendo uso de las arquitecturas propuestas y a su vez mostrar la funcionalidad en el prototipo construido.

### 1.4. Objetivos

#### 1.4.1. Objetivo general

Construir un prototipo de sistema basado en visión computacional para automatizar el proceso de clasificación de enfermedades en hojas de cafeto.

#### 1.4.2. Objetivo específico

1. Crear el dataset de imágenes de enfermedades en hojas del cafeto para entrenar los modelos de aprendizaje automático.

---

<sup>2</sup>Ver en:<https://www.infocafe.es/cafe/principales-productores-cafe.php>

2. Diseñar el modelo de aprendizaje automático y proponer otras arquitecturas basadas en redes neuronales convolucionales.
3. Entrenar y evaluar los modelos de redes neuronales y verificar la tasa de exactitud.
4. Aplicar técnicas de segmentación de imágenes y determinar el grado de severidad en la hoja, caso *Hemileia Vastatrix*.

## 1.5. Alcances y limitaciones

### 1.5.1. Alcances

- El cálculo de área de una región de la imagen se está realizando en base al conteo de píxeles, por lo que la captura de imágenes debe realizarse en un entorno plano.
- En la actualidad no se cuenta con un conjunto de imágenes etiquetadas de las enfermedades de café que dañan las hojas, se realiza la captura en un entorno semicontrolado, donde las imágenes de cada hoja enferma no tiene objetos al rededor de ella, y de igual manera el proceso de predicción se realiza con una hoja sin objetos al rededor; al considerar un entorno semicontrolado comprendemos que las imágenes son tomadas con un determinado fondo, evitando obstáculos que conllevan a un preprocesamiento más complejo.

### 1.5.2. Limitaciones

Se aprecia la complejidad del problema debido a que existen muchas enfermedades que dañan la producción del café, cabe señalar que hay enfermedades que se encuentran dentro del fruto y son difíciles de identificar tales como la broca de café, de igual manera con las enfermedades que afectan directamente a la raíz de la planta, así mismo suele suceder con los tallos, por tal motivo se investiga enfermedades que dañan las hojas del café porque son perceptibles a la visión humana, existen condiciones medio ambientales como oclusión, sombras, superposición, viento, que no permite una colección eficiente de datos, a continuación se define algunas limitaciones específicas del trabajo de investigación.

- Se limita las enfermedades a investigar, las que dañan las hojas con mayor incidencia y son perceptibles a la visión humana son *Leucoptera Coffeella*, *Mycena Citricolor*, *Hemileia Vastatrix*, **se trabaja con un total aproximado de 1000 hojas por enfermedad tanto para entrenamiento y validación, que son recolectados de cafetos de la Provincia de La Convención, Departamento del Cusco.**
- El cálculo de área de una región de la imagen se está realizando en base al conteo de píxeles, por lo que la captura de imágenes debe realizarse en un entorno plano.

- No se cuenta con recursos hardware autosuficientes tales como una cámara profesional para tomar buenas fotos, una buena computadora para realizar el proceso de entrenamiento, considerando que se usa Google Colaboratory este se reinicia cada cierto tiempo generando inconvenientes durante el entrenamiento.

## 1.6. Metodología

El trabajo de investigación tiene un enfoque exploratorio, porque pretende investigar las enfermedades en las hojas del cafeto y a su vez plantear una solución de automatizar el proceso de clasificación basado en visión computacional, creando un vínculo de estudio entre la biología del café y visión computacional el cual da origen a los resultados de un tema poco estudiado para familiarizarnos y se puede considerar aplicable a la sociedad. (Ernandez Sampieri, 2014). A continuación, se presentan los siguientes pasos a seguir:

- **Construcción del dataset de imágenes.** En esta etapa se lleva a cabo la toma de fotografías de las hojas del cafeto con enfermedades que son de interés, dichas imágenes suelen variar por muchos factores, estas deben ser tomadas de diferentes ángulos, al finalizar esta etapa se debe contar con un conjunto de imágenes.
- **Diseñar una arquitectura con redes neuronales convolucionales.** Esta etapa es importante, consiste en investigar técnicas de aprendizaje automático relacionados al aprendizaje supervisado como redes neuronales convolucionales, y diseñar una arquitectura de clasificación de enfermedades.
- **Desarrollo.** Consiste en la implementación del prototipo del sistema de clasificación de enfermedades basado en la arquitectura de red neuronal convolucional de la fase anterior, así como aplicar la pérdida de la entropía cruzada categórica para medir el rendimiento basado en la métrica precisión.

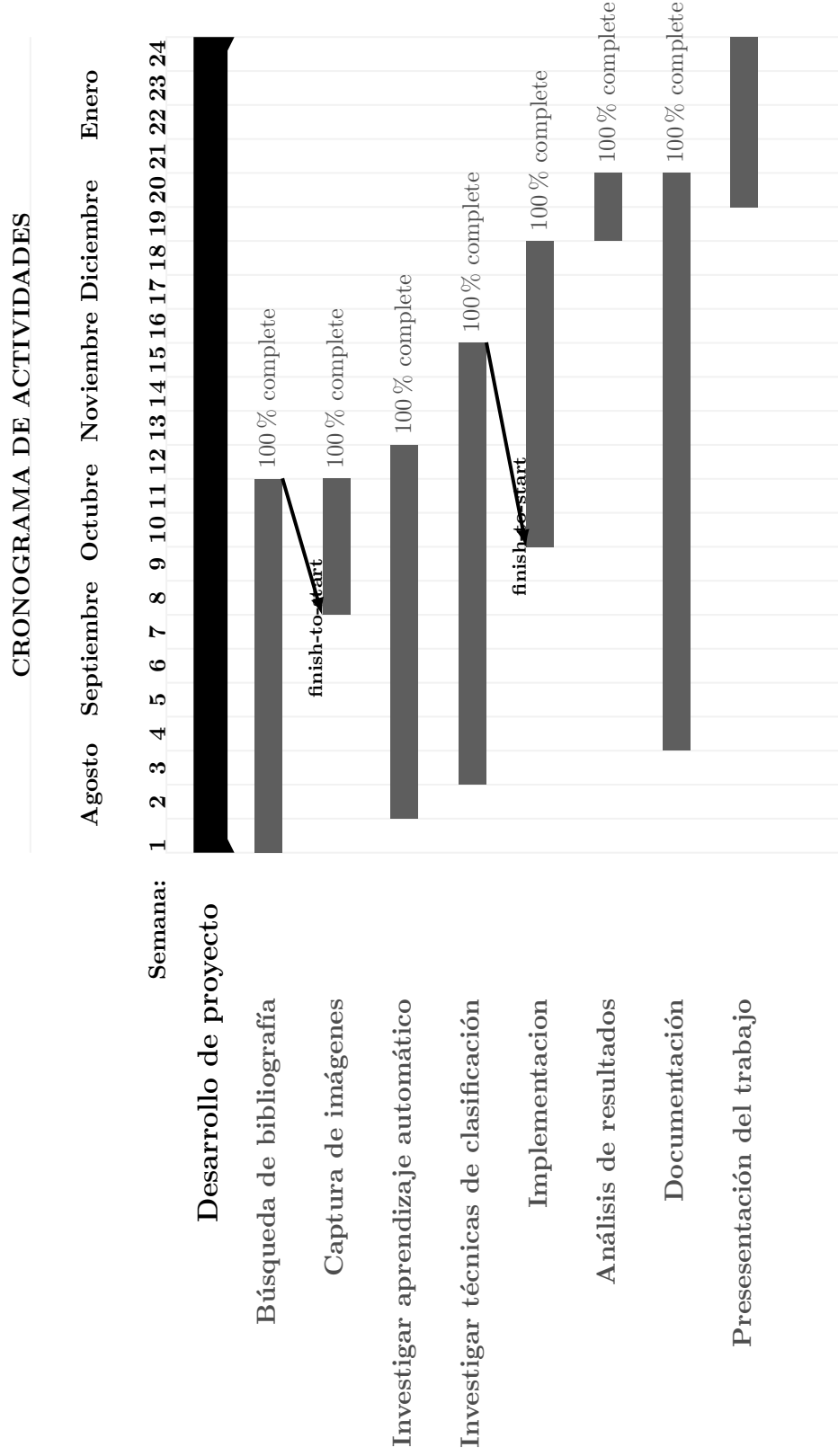
## 1.7. Contribuciones

Este trabajo de investigación cuenta principalmente con dos contribuciones para la comunidad de investigadores, agrónomos y caficultores:

- Contar con un diseño adecuado al contexto de investigación que sea capaz de clasificar las enfermedades que dañan las hojas del cafeto.
- Contar con un conjunto de imágenes de enfermedades que dañan las hojas del cafeto que pueden ser usados en trabajos de investigación futuros.

El siguiente trabajo de investigación espera tener el siguiente impacto social: Fomentar el desarrollo de trabajos de investigación del ámbito tecnológico aplicado al campo de la agricultura específicamente cultivo de café, para mejorar el manejo agronómico y el proceso de producción del café y así garantizar la calidad de granos.

## 1.8. Cronograma



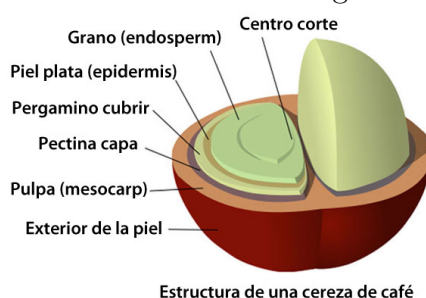
# Capítulo 2

## Marco conceptual

### 2.1. El Café

El café es un grano formado por dos semillas cubiertas por una sedosa envoltura, es el producto de interés que sufre muchas transformaciones para llegar a ser apto para el consumo humano y a su vez garantizar la calidad de este fruto. Los arbustos donde se alojan los granos se denominan cafeto, crecen en regiones tropicales y pertenecen a la familia de los rubiáceos del género *Coffea*. A continuación se muestra la estructura de un grano de café<sup>1</sup>, en la **figura 2.1**.

Figura 2.1: Estructura de un grano de café



Estructura de una cereza de café

fuelle: [www.tumundodelcafe.com/wp-content/uploads/2018/05/fruto-de-cafe.jpg](http://www.tumundodelcafe.com/wp-content/uploads/2018/05/fruto-de-cafe.jpg)

Tabla 2.1: Clasificación de enfermedades en el café

Área afectada	Nombre Común	Nombre Científico
Hojas	Minador de Café Ojo de Pollo Roya de Café	Leucoptera Coffeella Mycena Citricolor Hemileia Vastatrix
Frutos	Broca de Café Cercospora	Hypothenemus hampei Cercospora Coffeicola
Ramas/Tallos	Arañero	Pellicularia Koleroga

Fuente: SENASA (2003)

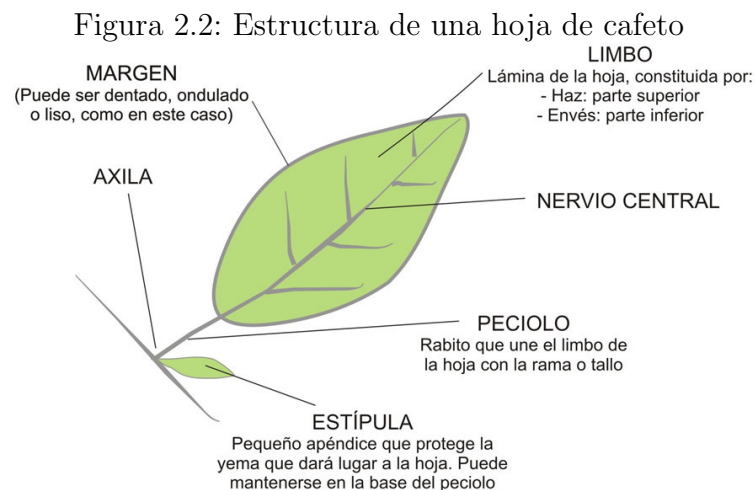
<sup>1</sup>ver en: [www.tumundodelcafe.com/morfologia-del-cafe/](http://www.tumundodelcafe.com/morfologia-del-cafe/)



### 2.1.1. Hojas del café

La hoja del cafeto es un componente muy fundamental para el proceso de crecimiento y maduración, al igual que en otras plantas la circulación de nutrientes así como el proceso de fotosíntesis es realizado a través de las hojas. Generalmente tienen un color verde oscuro y verde claro en la etapa temprana de maduración, las hojas del cafeto tienen seis partes como se muestra en la **Figura 2.2**.

- Estípula, son pequeñas protuberancias encargadas de la protección de la hoja.
- Pecíolo, tallo muy delgado que une las hojas con las ramas.
- Nervio Central, pequeñas venas que se encuentran sobre la hoja, son las encargadas del transporte de nutrientes.
- Margen, borde de la hoja.
- Limbo, toda la hoja donde se realiza la fotosíntesis, en esta región se encuentra la clorofila.



fuentes:<https://es-static.z-dn.net/files/d8b/f5181f5eee947bad1392be4f5cc4a45b.jpg>

### 2.1.2. Botánica del café

Las plantas de café también conocido con el nombre científico de *Coffea arabica*, tiene una peculiaridad natural de realizar paralelamente el proceso de florecimiento, generación de granos y maduración (granos con tonalidad carmesí), estos son recolectados varias veces al año de acuerdo al estado de sanidad que tengan las plantas pueden producir aproximadamente dos mil cuatrocientos gramos de granos verdes al año (Nini Maria, 2016).

### 2.1.3. Clasificación taxonómica de las enfermedades de café

#### Hemileia Vastatrix

Clasificado como un patógeno biotrófico que crece y se fortalece en el interior de los cultivos hospederos tales como el café, estudios morfológicos y moleculares demostraron la taxonomía de Hemileia Vastatrix. Perteneciente al Reino Fungi, Phylum Basidiomycota, orden Pucciniales, familia Chaconiaceae, género Hemileia, especie Hemileia Vastatrix y nombre común Roya de Café(Cáceda, 2018).

#### Leucoptera Coffeella

También conocido como micro lepidóptero que daña y se alimenta de las hojas del cultivo de café clasificado taxonómicamente como; nombre común Minador, nombre científico Leucoptera coffeella, Phylum Arthropoda, clase Insecto, orden Lepidoptera, familia Lyonetiidea("minador de hoja")(CUBIDES, 2019).

#### Mycena Citricolor

Enfermedad también conocido como Ojo de pollo cuyo objetivo es mermar la producción del café; perteneciente a Phylum Basidiomycota, Subphylum Agaricomycotina, Clase Basidiomycetes, orden Agaricales, familia Agaricaceae, género Mycena, especie Mycena citricolor(TÚCHEZ, 2019).

### 2.1.4. Enfermedades en las hojas de café

Las enfermedades que dañan las hojas del cafeto básicamente ocasionan un déficit en la producción, entre las enfermedades con mayor recurrencia en los cultivos se tiene Hemileia Vastatrix, Mycena Citricolor, Leucoptera Coffeella.

#### Hemileia Vastatrix(Roya Amarilla)

La roya de café es una de las enfermedades con mayor incidencia y más temida por los caficultores peruanos(Agarwal, 2018), esta enfermedad es causada por un hongo llamado Hemileia Vastatrix, perteneciente a la familia Puccineaceae, estos se alojan especialmente en las hojas, aparecen en forma de manchas amarillas y se extienden en el contorno de la hoja hasta tornarse un color naranja débil con el objetivo de matar las hojas y producir la caída, las etapas en el proceso de la enfermedad son(Barquero Miranda, 2013).

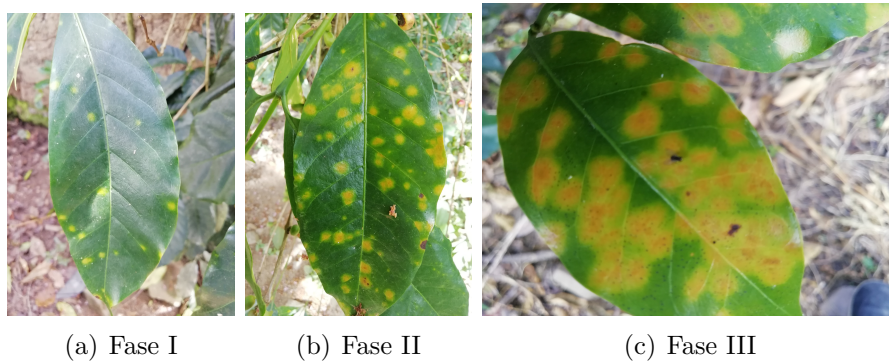
- **Crecimiento lento.-** Consiste en el inicio de producción de esporas, hongos en las hojas, se presentan pequeños puntos amarillos que se propagan con dificultad.
- **Crecimiento acelerado.-** En esta etapa la incidencia de la roya incrementa considerablemente de un mes a otro, dependiendo de las condiciones climatológicas, si la altitud sobre el nivel del mar es mayor tarda en reproducirse esta enfermedad.
- **Crecimiento máximo y final.-**En esta etapa la incidencia de la enfermedad en las hojas es alta, a consecuencia de ellos las hojas tienden a caer(Barquero Miranda, 2013).

Tabla 2.2: Grado de severidad de la roya de café

Grado	Descripción
0	No tiene síntomas visibles, hoja completamente sana.
1	Área afectada del 1 a 5 % del área foliar.
2	Área afectada del 6 al 20 % del área foliar.
3	Área afectada del 21 al 50 % del área foliar.
4	Mayor al 50 del área foliar se encuentra afectada.

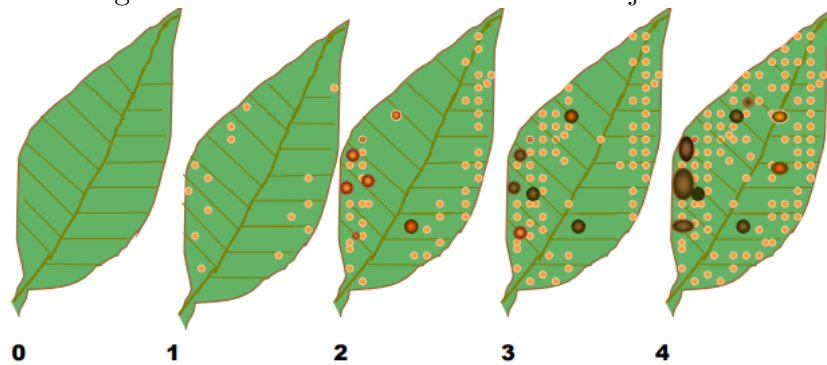
Fuente: SENASA (2003).

Figura 2.3: Fases Hemileia Vastatrix.



Fuente: Propia

Figura 2.4: Grado de severidad en la hoja de café



Fuente: [www.cesvver.org.mx/wp-content/uploads/Roya07.jpg](http://www.cesvver.org.mx/wp-content/uploads/Roya07.jpg)

Para el cálculo del índice de intensidad de daño (IID) de la enfermedad es necesario determinar el grado de severidad como se muestra en la tabla 2.2 (SENASA, 2003)

$$IID = \frac{\sum \#(GRADO) \times N^{\circ} \text{Hojas por cada grado}}{N^{\circ} \text{ GRADOS MAYOR} \times N^{\circ} \text{HOJAS EVALUADAS}} \quad (2.1)$$

- Proceso de diseminación.
  - Diseminación por el viento ocasionando caída de hojas enfermas.
  - Las precipitaciones fluviales ocasionan el origen de esporas y diseminación.
  - Ventajas por exceso de sombras y alta densidad de plantas.
- Proceso de control.
  - Aplicar oxiclورو de cobre posterior a la floración de los cultivos.
  - Aplicar caldo bordalés, posterior a la floración de los cultivos.
  - Aplicar azufre en épocas de sequía.
  - Aplicar Benomil, Folicur y Strobry.
  - Fertilizar las plantas con calcio regularmente y balanceadamente.

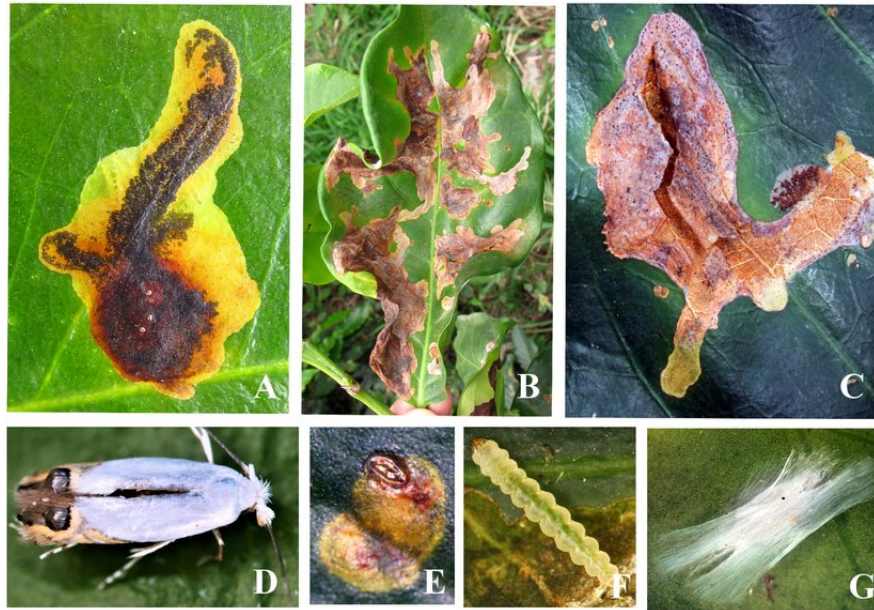
### **Leucoptera Coffeella(Minador de Café)**

Las hojas del cafeto son afectadas por larvas destruyendo la dermis debilitando la hoja. *Leucoptera Coffeella* es una plaga que sufre una transformación de larva en mariposa que permanece en los cultivos en periodos de verano, a continuación se muestra los estados biológicos durante el desarrollo de la *Leucoptera* como se muestra en la **figura 2.5** (Constantino Luis, 2011).

- Estado A.- Mina activa, donde el follaje de la hoja del cafeto ya se encuentra dañada.
- Estado B.- Daño en mas del 70 % del área foliar.
- Estado C.- Mina inactiva
- Estado D.- Adulto
- Estado E.- Huevo
- Estado F.- Larva
- Estado G.- Pupa

El término infestación es usado para evaluar insectos, caso *Leucoptera Coffeella*, se realiza de la siguiente manera, se seleccionan 10 plantas, de estas se elige una cantidad de hojas por planta, y se define cuantas tienen la infestación, el resultado es la sumatoria del cociente de la cantidad de hojas infectadas y cantidad total de hojas por planta.(SENASA, 2003).

Figura 2.5: Estado biológico de la *Leucoptera Coffeella*



Fuente:(Rueda et al., 2018)

- Biología
  - Permanecen inmóviles durante el día en el envés de las hojas.
  - En la noche se reproducen mediante la oviposición en el haz de las hojas.
  - Las larvas se alimentan de la dermis de las hojas ocasionando defoliación.
  - Disminuye el ataque a las plantas gracias a las precipitaciones fluviales (mayor infestación en temporada seca).
  - Las plagas se expanden por el exceso de sombras.
- Daños
  - Minaduras en el envés de las hojas a un principio claras posterior a ello color pardo o marrón oscuro.
  - A mayor índice de infestación defoliación de las hojas.
  - Crisis en producción y calidad del grano.
- Proceso de control
  - Evitar el uso excesivo de abonos nitrogenados.
  - Criar y liberar de parasitoides(*Neochrysocharis immaculatus*, *Cirrospilus*, *Microlygus* y *pediobius*) en los cultivos de café.
  - Crianza y liberación de predadores(*Crematogaster* y *Chrysoperla*)
  - Aplicar químicos como insecticidas, traslaminables, lufenurón y abamectina.

- Aplicar abonos ricos en calcio.
- Realizar podas periódicas y evitar el exceso de sombras.

### **Mycena Citricolor(Ojo de Pollo)**

Es una de las enfermedades que estropean las hojas y frutos, causada por el hongo **Mycena Citricolor** que presenta una mancha redonda y hundida en frutos, color amarillo al inicio finalizando con un color pardo, y en hojas presenta una característica peculiar de manchas circulares visibles en haz y envés de las hojas, color marron oscuro a un inicio y color blanquecino al final con perforaciones de hojas y defoliación(Lizarra Valencia, 2012).

- Proceso de diseminación
  - El emboscamiento de la parcela(concentración de la humedad) favorece la caída de hojas.
  - Ataques constantes en zonas con baja altitud sobre nivel del mar.
  - Ataques severos en regiones limítrofes con el bosque.
- Proceso de control
  - Podas periódicas para la ventilación de los cultivos.
  - Podar los árboles para evitar exceso de sombra y ventilar la parcela.
  - Utilizar caldo bordales posterior a la floración y evitar aplicarlo en otro periodo.
  - Aplicar Amistar y Folicur.

Figura 2.6: Mycena Citricolor



Fuente: Elaboración propia

Tabla 2.3: Información técnica de enfermedades en hojas de café.

	<b>Hemileia Vastatrix (Roya amarilla de café)</b>	<b>Leucoptera Coffeella (Minador de Café)</b>	<b>Mycena Citricolor (Ojo de Pollo)</b>
<b>Biología</b>	Manchas redondeadas amarillas y naranjas, con el impacto se desprende un polvo, crecimiento gradual.	En etapa adulta inactivos en el día, presencia de oviposición en el haz de las hojas durante la noche, la larva penetra en la dermis de la hoja, consumiendo la parte interna, las presipitaciones fluviales detienen el vuelo y disminuye el ataque; se presencia mayor infestación en sequía, favorecido por el exceso de sombra.	Manchas redondas undidas, diferentes tamaños, colores amarillos al inicio y pardo al final
<b>Agente</b>	Hongo Hemileia Vastatrix	polilla Perileucoptera coffeella.	Hongo Mycena Citricolor
<b>Daño</b>	Defoliación y muerte descendente de acuerdo a la infestación. Caída de hojas	Minadura en aspecto de ampolla al iniciar verdoso claro y se transforma en un color pardo marrón oscuro.	Emboscamiento de parcela, evitando el florecimiento proporcional.
<b>Control</b>	Aplicaciones de ocloruro de cobre después de la floración, Aplicar azufre si se presenta sequía, fertilización balanceada con calcio, aplicar Bonomil, Follicur y Strobby.	Evitar el abuso de abonos nitrogenados. Crianza y liberación de parasitoides(Neochrysocharis immaculatus, Cirrospilus, Microlygus y Pediobius); predadores(Crematogaster y Chrisoperla). Control químico con el uso de insecticidas traslaminantes como el lufenuron y abamectina.. Abonamiento rico en calcio(fortalecimiento de la planta. Evitar el exceso de sombra en el cultivo.	Realizar podas de ventilación, raleo del bosque aledaño, aplicar Amistar y Follicur.

**Fuente: Propia**

## 2.2. Visión computacional

Para poder entender visión computacional es esencial analizar conceptos de visión, se caracteriza por el proceso de producir información útil para el observador y que no tiene características irrelevantes, es realizada a partir de una imagen.(Marr, 1982). Sin embargo visión computacional viene a ser el proceso computacional de obtener información útil para el observador reduciendo áreas innecesarias de la imagen para su posterior procesamiento e interpretación.(Enrique Sucar, 2018a)

### 2.2.1. Procesamiento de imágenes

Tiene como objetivo, trabajar en las regiones de interés de una imagen para su posterior reutilización. Durante el procesamiento de imágenes se puede aplicar transformaciones para mejorarlas(Enrique Sucar, 2018a):

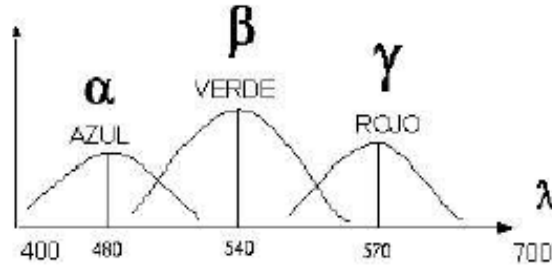
- Eliminar ruido de la imagen, así como malformaciones.
- Aplicar ecualización y otras técnicas para mejorar el contraste, color y otras propiedades de la imagen.
- Extraer objetos de interés, a través de técnicas de segmentación, etc.

Características básicas en una imagen:

- **Imagen.-** Representación bidimensional del mundo tridimensional, definida como una matriz de dimensiones  $N \times M$ .
- **Histograma.-** Es la representación de la frecuencia de aparición de cada uno de los valores de la imagen.
- **Brillo.-** Es el proceso de añadir cierta cantidad al valor de intensidad de los píxeles, si la cantidad es positiva incrementa el brillo caso contrario disminuye.
- **Contraste.-** Propiedad de la imagen que incrementa la textura.
- **Color.-** Definido como la abstracción del medio relacionado con la respuesta de la visión humana a las longitudes de onda del espectro visible, relacionado a los 3 tipos de sensores que existe en el ojo que responde de acuerdo a la longitud de onda, la combinación de estas señales de sensación forma los colores que podemos abstraer o percibir, existen formas de codificar los diferentes espacios de color que se detallan a continuación(Enrique Sucar, 2018b).
  - **RGB.-** Viene a ser la combinación de los 3 colores primarios, R(Rojo),G(verde), B(Azul).
  - **HSI.-** Modelo que mejor se adecúa a la percepción humana, sus componentes son I(intensidad), H(Croma), S(saturación).



Figura 2.7: Ondas de color



Fuente: (Enrique Sucar, 2018b), Figura 1.14

### Operaciones puntuales

Una operación puntual se define como una función de transformación de una imagen de entrada en una imagen de salida, existiendo una correspondencia entre el pixel de entrada y el pixel de salida definido matemáticamente de la siguiente manera.

$$S(x, y) = f(E(x, y)) \quad (2.2)$$

Binarización por umbral es una operación puntual para binarizar la imagen, obteniendo una imagen en escala de grises, con valores de pixel 0 y 1 de acuerdo a un parámetro llamado umbral T (threshold).

$$S(x, y) = 1, \quad E(x, y) > T \quad S(x, y) = 0, \quad E(x, y) \leq T \quad (2.3)$$

Esta técnica puede ser usada para separar regiones de interés, con el único problema de determinar un valor óptimo para el umbral T (Enrique Sucar, 2018c).

### Ecualización del histograma

Para definir ecualización del histograma es necesario definir histograma. El histograma de intensidades es la distribución de cada nivel de intensidades dentro de una imagen, proporcionando la probabilidad de ocurrencias en cada nivel de grises. La ecualización del histograma tiene como objetivo mejorar una imagen que tiene bajo contraste, obteniendo una mejora para la apreciación de los detalles de interés.

$$p(r_k) = \frac{n_k}{n} \quad (2.4)$$

Donde:

$p(r_k)$  : Probabilidad del nivel  $k$ .

$n_k$  : Número de pixeles que toma el valor del nivel  $k$ .

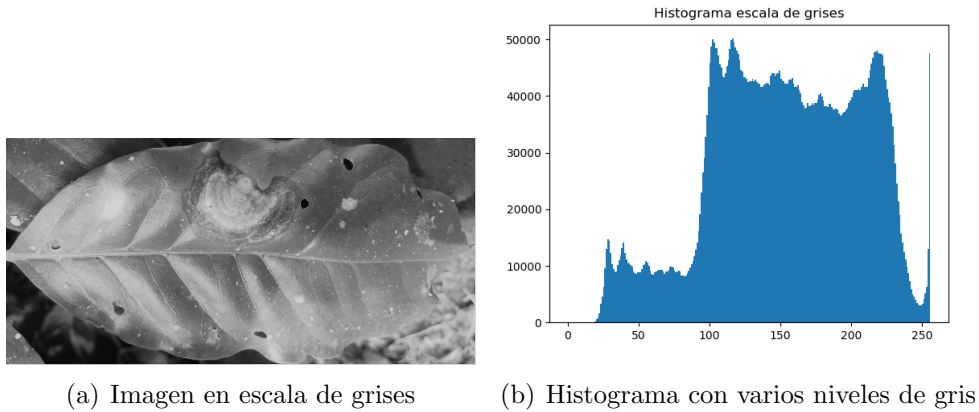
$n$  : Número total de pixeles en la imagen.

Para obtener la ecualización del histograma se tiene que realizar el siguiente proceso:

Definamos un nuevo nivel.

$$s = T(r) \quad (2.5)$$

Figura 2.8: Histograma, Minador de café.



Fuente: Propia

Donde:

$T$ : Función monótonicamente creciente.

$$0 \leq T \leq 1$$

Considerando:

$p(s), p(r)$ : Densidades de probabilidad.

Por teoría de probabilidad se tiene:

$$p(s) = \left[ p(r) \frac{dr}{ds} \right] \quad (2.6)$$

Utilizando como función de transformación la función acumulativa se tiene:

$$s = T(r) = \int p(r) dr \quad (2.7)$$

Derivando  $s$  con respecto a  $r$  en 2.7, se obtiene:

$$\frac{d(s)}{d(r)} = p(r) \quad (2.8)$$

Sustituyendo 2.8 en 2.6, se obtiene:

$$p(s) = 1 \quad (2.9)$$

Así se obtiene una distribución uniforme para el histograma y se logra mejorar el contraste, en caso discreto se tiene:

$$s(k) = T(r) = \sum_{i=0}^k \frac{n_i}{n} \quad (2.10)$$

### Filtrado de imágenes

El filtrado de una imagen consiste en aplicar una transformación  $T$  a una imagen de entrada ( $f$ ), generando una imagen de salida ( $g$ ) con la finalidad de acentuar o disminuir ciertas características de la imagen.

$$g(x, y) = T[f(x, y)] \quad (2.11)$$

El conjunto de puntos que componen una imagen se denomina **dominio espacial**, donde se aplican transformaciones ( $T$ ) sobre la imagen de entrada  $f(x, y)$ , la transformación es un operador que se aplica a las vecindades del pixel de interés. Una de las formas de aplicar las transformaciones es a través de una máscara, una máscara se define como una matriz bidimensional de tamaño 3x3 o más, que se desplaza como una ventana deslizante a lo largo de la imagen multiplicando los valores en común de la imagen y de la máscara (Gonzales, 1995).

$$\begin{pmatrix} w_1(x-1, y-1) & w_2(x, y-1) & w_3(x+1, y-1) \\ w_4(x-1, y) & w_5(x, y) & w_6(x+1, y) \\ w_7(x-1, y+1) & w_8(x, y+1) & w_9(x+1, y+1) \end{pmatrix}$$

Al aplicar la máscara se obtiene la nueva imagen  $g(x)$

$$a = w_1f(x-1, y-1) + w_2f(x, y-1)$$

$$b = w_3f(x+1, y-1) + w_4f(x-1, y) + w_5f(x, y)$$

$$g(x) = a+b+w_6f(x+1, y)+w_7f(x-1, y+1)+w_8f(x, y+1)+w_9f(x+1, y+1) \quad (2.12)$$

### Suavizado de imágenes

Son utilizados generalmente para disminuir los efectos negativos que puede tener una imagen digital, por ejemplo el ruido y otros que no sean de interés, a continuación se muestra algunos métodos de suavizado.

- **Promedio de vecinos.-** Este viene a ser el resultado del promedio de los valores del nivel de gris de los puntos de  $f$  contenidos por cierta vecindad en  $(x, y)$ , representado por  $g(x, y)$  la imagen suavizada; asumiendo que se tiene una imagen de entrada  $f(x, y)$  de  $N \times N$  (Gonzales, 1995).

$$g(x, y) = \frac{1}{M} \sum_{(n,m) \in S} f(n, m) \quad (2.13)$$

Donde:

$$(x, y) = 1, 2, \dots, N - 1$$

$S$ : Conjunto de coordenadas de puntos vecinos a  $(x, y)$ .

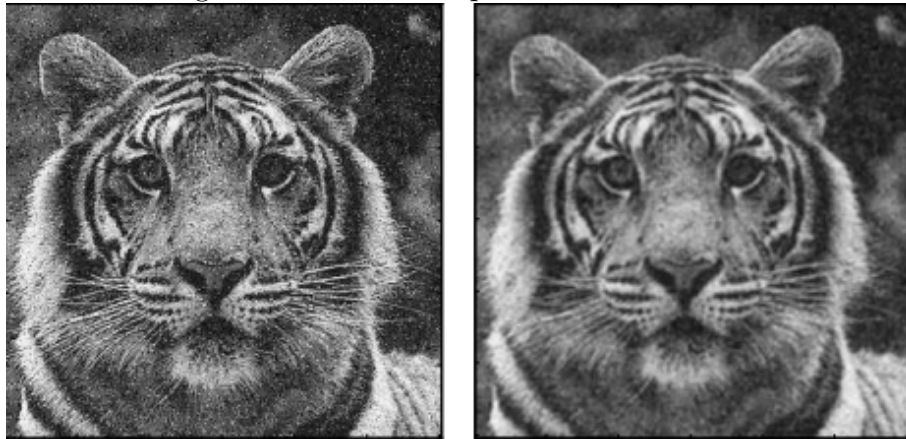
$M$ : Conjunto de puntos de vecindad.

$$\begin{bmatrix} & & \dots & & \\ \dots & (x-1, y-1) & (x, y-1) & (x+1, y-1) & \dots \\ & (x-1, y) & (x, y) & (x+1, y) & \\ \dots & (x-1, y+1) & (x, y+1) & (x+1, y+1) & \dots \\ & & \dots & & \end{bmatrix} \quad (2.14)$$

$$\begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix} \quad (2.15)$$

Consideramos que 2.14 es un fragmento de una imagen y 2.15 es el filtro a aplicar.

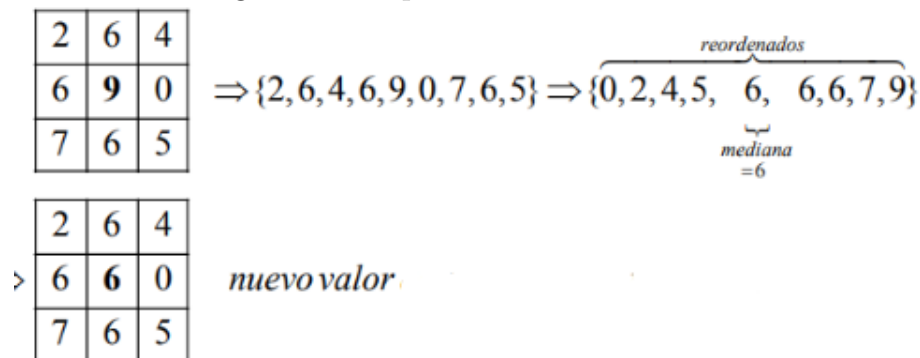
Figura 2.9: Filtro con promedio de vecinos.



Fuente: (Gonzales, 1995), Figura Pag. 35

- Filtro de la mediana.**- El método anterior difumina los bordes y otros detalles de contraste, para resolver el problema hacemos uso de filtros de mediana donde remplazamos el valor de gris de un punto por la mediana de los niveles de gris de la vecindad. La mediana  $m$  de un conjunto de valores es el valor medio representado por  $m$  de un conjunto de valores ordenados. “La función principal de los filtros de mediana es forzar a los puntos con valores de intensidad muy distintos a sus vecinos a tener valores más próximos a sus vecinos, de modo que se eliminan los picos de intensidad que aparecen en áreas aisladas. Un ejemplo de la utilidad de este tipo de filtro se muestra en la siguiente imagen. La primera imagen es la original” (Gonzales, 1995).

Figura 2.10: Aplicación de la mediana.

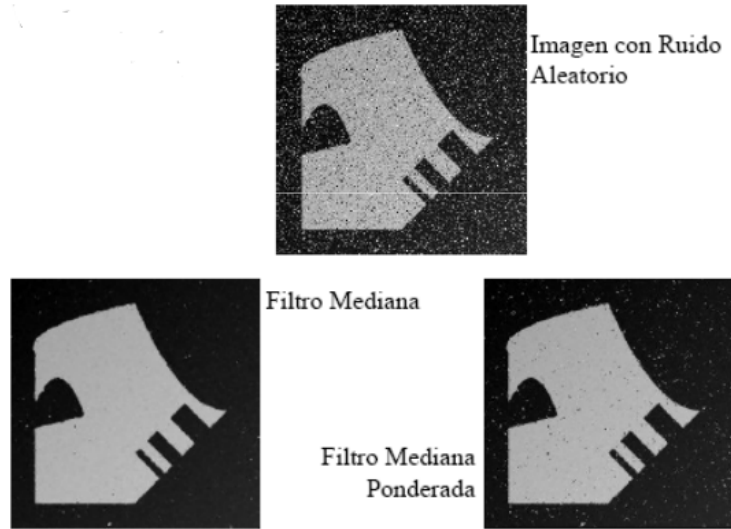


Fuente: Jorge Marquez, Figura Pag. 3

- Filtro Gaussiano.**- Filtro donde el valor máximo se encuentra en el pixel central, disminuye los valores hacia los extremos de la matriz de la imagen, cuando el valor de la desviación típica es muy pequeña se obtiene una matriz con valores entre 0 y 1, para convertirlo en enteros se divide por el valor menor (GAU, 2019).

$$g(x, y) = e^{-\frac{x^2+y^2}{2*s^2}} \quad (2.16)$$

Figura 2.11: Filtro de la mediana.



Fuente: <http://alojamientos.us.es/gtocom/pid/tema3-1.pdf>, Pag 31

Tabla 2.4: Filtro gaussiano con  $s=1$  y matriz de filtrado identidad.

1	4	7	4	1	1	0	0
4	20	33	20	4	0	1	0
7	33	55	33	7	0	0	1
4	20	33	20	4			
1	4	7	4	1			

$$G(x, y) = \frac{g(x, y)}{\text{Min}_{x,y}(g(x, y))} \quad (2.17)$$

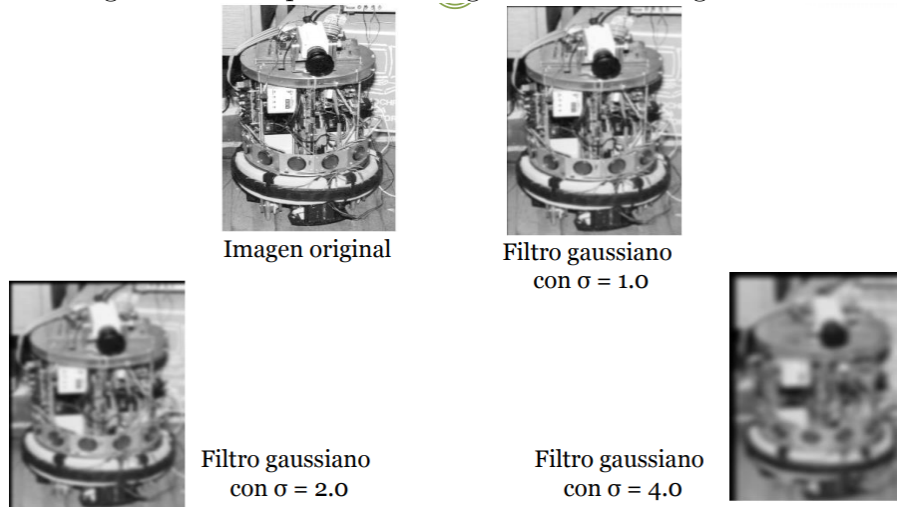
### 2.2.2. Técnicas de segmentación

La segmentación de imágenes es el proceso de subdividir una imagen en diferentes objetos, partes constituyentes de la imagen; con el objetivo de separar las partes de interés del resto de la imagen de acuerdo a la aplicación en este caso contribuyendo al reconocimiento eficiente de la enfermedad en hojas de café (La Serna Palomino, 2009b). Así mismo se puede definir como la clasificación de píxeles de la imagen que presentan cierta similitud, el proceso de segmentación tiene atributos básicos como luminancia en imágenes monocromáticas, componentes de color, textura, forma, etc (La Serna Palomino, 2009a).

#### Segmentación por umbralización

El umbral es una función que tiene el objetivo de convertir una imagen de diferentes tonalidades en una imagen en blanco y negro de acuerdo al valor obtenido por el umbral. Si se aplica a segmentación básicamente vendría a ser el proceso

Figura 2.12: Representación gráfica del filtro gaussiano.



Fuente:

[http://asignatura.us.es/imagendigital/Tema2-ParteII\\_Filtros.pdf](http://asignatura.us.es/imagendigital/Tema2-ParteII_Filtros.pdf), Pag 24

realizado por el umbral aplicado a una imagen.

Sea  $f(x, y)$  la imagen original, y  $g(x, y)$  imagen umbralizada.

Sea el umbral fijo definido por:  $0 \leq U \leq 255$

Las ecuaciones de umbralización se definen de la siguiente manera.

$$g(x, y) = 255, \text{ si } f(x, y) < \text{Umbral} \quad , g(x, y) = 0, \text{ si } f(x, y) \geq \text{Umbral} \quad (2.18)$$

De esta manera se elige un valor para el umbral permitiendo agrupar los pixeles de una imagen pertenecientes a los objetos de interés dependiente del umbral. Así la segmentación basada en el histograma esta relacionada con la elección de varios umbrales que permite agrupar pixeles pertenecientes a características de interés en función a los niveles de gris.

El umbral de una imagen esta definido por:

$$T = T[x, y, p(x, y), f(x, y)] \quad (2.19)$$

Donde:

$f(x, y)$ : Intensidad o nivel de gris en el punto  $(x, y)$ .

$p(x, y)$ : Representa alguna propiedad local en la vecindad del punto de interés. La ecuación de la imagen de salida esta definido por:(La Serna Palomino, 2009c)

$$\left. \begin{aligned} g(x, y) &= 0 \quad \text{si, } f(x, y) > T \\ g(x, y) &= 1 \quad \text{si, } f(x, y) \leq T \end{aligned} \right\} \text{ Ecuación de la imagen de salida, aplicando umbralización.}$$

**Elección del umbral óptimo.-** La función global del histograma esta representada por la siguiente expresión matemática.

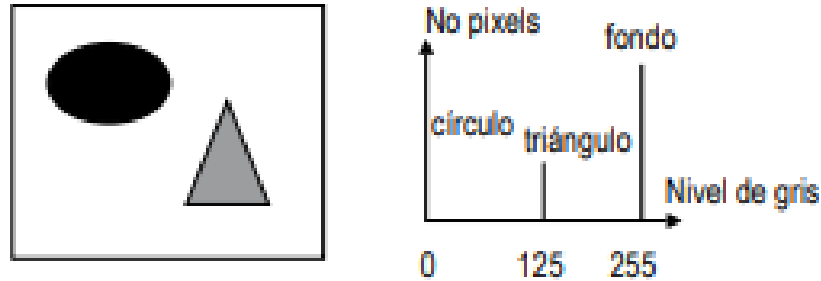
$$p(z) = P_1 p_1(z) + P_2 p_2(z) \quad (2.20)$$

Donde:  $z$ : Variable aleatoria de intensidad.

$p_1(z), p_2(z)$ : Funciones de densidad de probabilidad.

$P_1, P_2$ : Probabilidades a priori.

Figura 2.13: Ejemplo de umbralización



Fuente: La Serna Palomino (2009c), Pag.15

### Segmentación basada en el color

La aplicación de segmentación basada en el color tiene un vínculo con la umbralización, eligiendo umbrales  $T_1$  y  $T_2$  los cuales están relacionados con el rango de tonalidad verde con propiedades RGB para aislar la región sana de la enferma en la hoja de café.

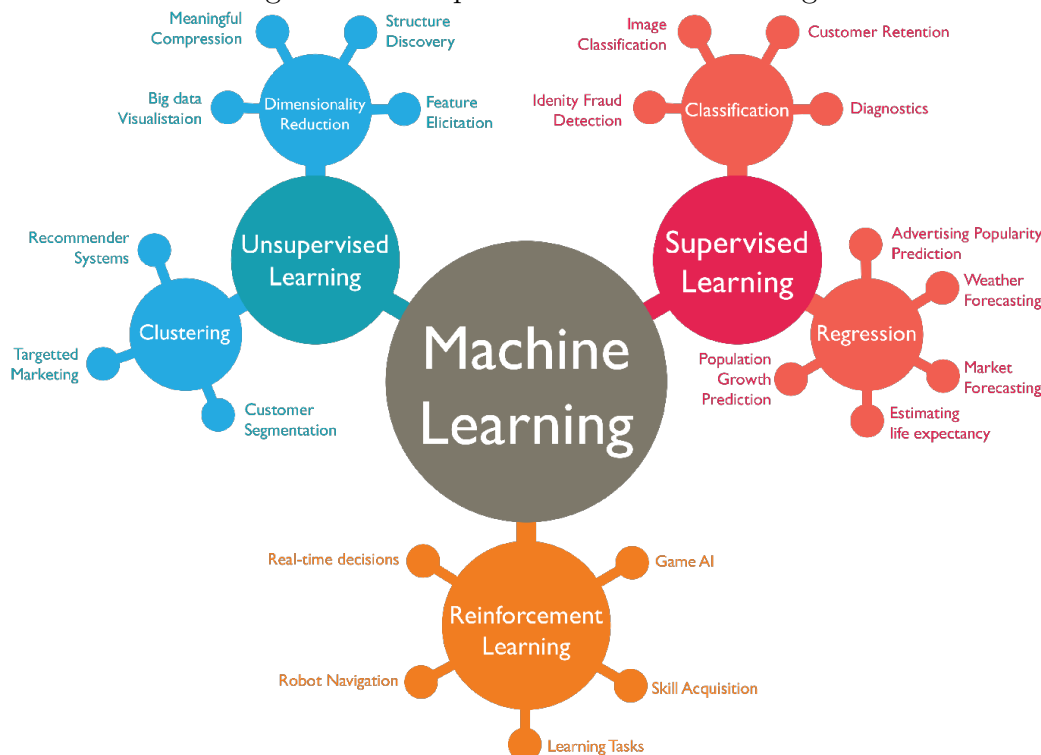
## 2.3. Aprendizaje máquina

Es una rama de la inteligencia artificial, definido como el conjunto de técnicas computacionales basado en brindar una capacidad cognitiva a una máquina a través del entrenamiento de una arquitectura de aprendizaje máquina; puede ser una red neuronal, un clasificador como K-Vecinos Próximos, Máquina de Vector de Soporte, entre otros. El proceso de entrenamiento puede realizarse a través de ejemplos, con la finalidad de educar y fomentar la autonomía en la concretización de una tarea. Tom M. Mitchell define aprendizaje máquina como, “un programa informático que aprende de la experiencia  $E$  respecto a algún tipo de tareas  $T$ , tal y como es medido por  $P$ , mejora con la Experiencia  $E$ .” (Tom M., 2001) Si se tiene un conjunto de entrada y sus correspondientes salidas denominados datos de entrenamiento, el programa debe ser capaz de entrenar y ajustarse a la algoritmia interna para poder predecir el resultado de nuevas entradas que no pertenezcan al conjunto de entrenamiento (Cantero Lorenzo, 2018).

Las técnicas de aprendizaje máquina tienen muchas aplicaciones interdisciplinarias tales como: Investigación biomédica, reconocimiento de patrones, conducción automática (Chenyi Chen, 2019), etc; a continuación se muestra un esquema general de los conceptos a tratar relacionados a la clasificación de enfermedades en hojas de café (OTECH, 2019).

El aprendizaje automático también es considerado como una rama específica de las ciencias de la computación e inteligencia artificial en el cual se crean sistemas

Figura 2.14: Esquema de machine learning



Fuente: [www.profesionalreview.com/wp-content/uploads/2019/08/Machine-Learning-1.png](http://www.profesionalreview.com/wp-content/uploads/2019/08/Machine-Learning-1.png)

capaces de aprender automáticamente. El aprendizaje automático está constituido por tres áreas de estudio tales como aprendizaje supervisado, aprendizaje no supervisado y aprendizaje por refuerzo (OTECH, 2019).

- **Aprendizaje supervisado.**- Es el aprendizaje en el cual se ingresan datos y sus respectivos resultados con los que crea relaciones entre ideas.
- **Aprendizaje no Supervisado.**- Es un entorno donde no se tiene respuestas y la propia inteligencia artificial tendrá que generar sus respuestas, sean o no acertadas, a consecuencia de ello crea su propia regla.
- **Aprendizaje por refuerzo.**- Cuando se le entrena a la arquitectura en las habilidades que flaquea para fortalecer, caso de una arquitectura que no detecte imágenes con poca luminosidad se le entrena con imágenes tomadas en la oscuridad, este aprendizaje es llamado por refuerzo.

### 2.3.1. Aprendizaje supervisado

Consiste en realizar predicciones a futuro basadas en comportamientos o características ya registradas (histórico de datos), permitiendo la búsqueda de patrones de datos históricos relacionando todo con un campo específico, denominado campo objetivo. Por ejemplo, la clasificación de correos mediante etiquetas de “spam” o “válido”; dando inicio al análisis de características o patrones que tienen los correos etiquetados. Dentro de la categoría del aprendizaje supervisado destacan



algoritmos de clasificación y regresión.

## Regresión

Predice valores continuos (valores numéricos, 1,2,3, etc) (Varios, 2015), implica que el resultado a las preguntas esta expresado mediante una cantidad que se define de manera flexible según a los parámetros de entrada (Ligdi, 2018).

Ejemplos:

- Predecir el precio de venta de una propiedad.
- Predecir el tiempo de permanencia de un empleado en una empresa.
- Tiempo estimado de recorrido de un vehículo del origen al destino.

## Clasificación

Muestras etiquetadas como pertenecientes a una determinada clase, dos o más clases tambien denominado categoría, nos referimos a categorías arbitrarios según el tipo de problema, entre las técnicas de clasificación de aprendizaje automático se tienen a las redes neuronales, K-Vecinos Próximos, Máquinas de Vector de Soporte, entre otros. (Gato Utrera, 2017)

Ejemplos:

- Determinar la categoría de una imagen, si este corresponde a un gato o perro, u otra clase.
- Definir el tipo de tumor, si es benigno o maligno.

### 2.3.2. Redes neuronales convolucionales

Las redes neuronales artificiales son objetos de programación que imitan el funcionamiento de las neuronas biológicas, formadas por una interconexión de redes que tienen una estructura jerárquica con información en cada neurona, así mismo se define como un método de resolver problemas complejas de manera automática (Rivas-Asanza et al., 2018). Una red neuronal artificial puede asimilarse a una estructura del sistema nervioso.

En una red de neuronas existe un valor sináptico o peso que pondera las señales que recibe por sus entradas y que determina la fuerza de conexión entre dos neuronas, al evaluar una neurona se calcula el conjunto de fuerzas o valores sobre la neurona (llamado NET), la combinación de señales que puede recibir una neurona esta expresada por la siguiente fórmula (Gestal Pose, 2019).

$$NET_i(t) = \sum_{j=1}^{N-1} [W_{ij} \cdot O_j \cdot (t-1)] \quad (2.21)$$

Donde  $W_{ij}$  es el peso de conexión de una neurona emisora  $i$  a una neurona receptora  $j$ .

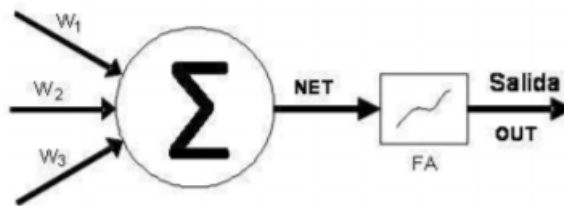
**Capas neuronales.-** Una vez definida una neurona computacional se procede con la construcción de una red neuronal completa, se encuentra constituida por

Tabla 2.5: Características de una red neuronal artificial

Característica	Descripción
Topología de red	Permite determinar la capacidad representativa de como una cantidad de neuronas estan distribuidos en capas y distribuidas entre sí. Donde la topología de red se basa en el problema a resolver.
Regla de aprendizaje	Una red neuronal es una integración de diversos sistemas de aprendizaje, es por ello que tienen la capacidad de aprender a través de un entrenamiento previo.
Tipo de entrenamiento	Una red neuronal artificial presenta dos tipos de entrenamiento, una que durante la etapa de aprendizaje la red se entrena para que los pesos sinápticos se adecuen a la red. La segunda es una etapa de ejecución en donde la red ya pasa ser operativa cuya red toma un valor de funcionamiento real.

Fuente: Rivas-Asanza et al. (2018)

Figura 2.15: Neurona artificial



Fuente: Gestal Pose (2019) , Figura 2

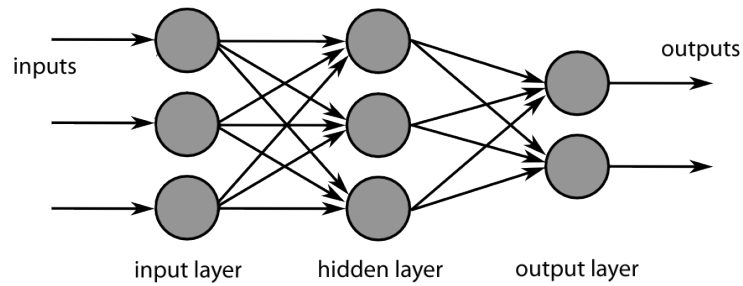
capas neuronales tales como la capa de entrada, capas ocultas y una capa de salida. La estructura de cada capa neuronal esta compuesto por nodos o neuronas, la capa de entrada es la encargada de proyectar los datos de entrada sobre los primeros nodos de la capa oculta, esta capa tiene la cantidad de nodos o neuronas que son requeridos para la aplicación(arquitectura de red neuronal para resolver un problema); las capas ocultas se encargan del procesamiento de datos recibidos de otras neuronas aplicando una función de transformación  $f(x)$ . La última capa oculta envía los datos procesados a la capa de salida, se llama **Feedforward**(Auer, 2008) al esquema de conexión donde la información viaja de la entrada hacia la salida en la red(Cantero Lorenzo, 2017).

**Función de activación.-** Es la función de transformación aplicada sobre la entrada de información de un nodo para producir la salida; existen funciones de activación tales como Sigmoides, Relu, Softmax, etc. La función de transformación esta definida por:

$$f : \mathbb{R} \rightarrow \mathbb{R}$$

Las redes neuronales convolucionales tienen un comportamiento similar a la

Figura 2.16: Red neuronal bajo el esquema Feedforward



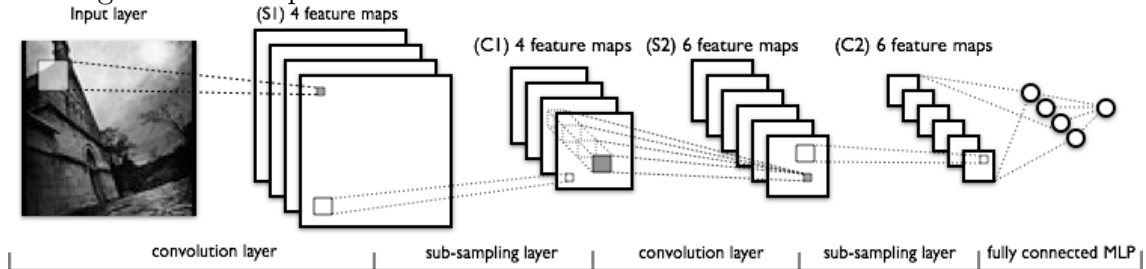
Fuente: Cantero Lorenzo (2017), Figura 6

corteza de la visión humana y tiene una característica matricial bidimensional lo que hace más eficiente en aplicaciones de clasificación y segmentación de imágenes.

### Arquitectura de red neuronal convolucional

Las redes neuronales convolucionales presentan diferentes capas como filtros convolucionales, una arquitectura neuronal enfocada a la clasificación esta basado en la extracción de características, la clasificación esta ligado a neuronas convolucionales y reducción de muestreo.

Figura 2.17: Arquitectura referencial del una red neuronal convolucional



fuelle: [bit.ly/2KQQq4](http://bit.ly/2KQQq4)

El objetivo de la convolución es extraer características de la imagen de entrada, que consiste en añadir cada elemento ponderado del núcleo a sus vecinos. El kernel o núcleo es una matriz bidimensional cuadrada de tamaño impar menor o igual a las dimensiones de la imagen de entrada, el objetivo del kernel es realizar operación para determinar los bordes horizontales, verticales, rectas, que son características propiamente dichas de la imagen, al efectuar la convolución se procede con los siguientes pasos.

- Teniendo un kernel se procede con el cálculo matemático de multiplicar cada elemento del kernel con su parecido de la imagen.
- Se obtiene un valor de la suma de los productos del paso anterior.
- El elemento central del fragmento toma el valor calculado.

El procedimiento se va repitiendo hasta recorrer toda la imagen; en una estructura de red neuronal convolucional, se presenta las siguientes capas.

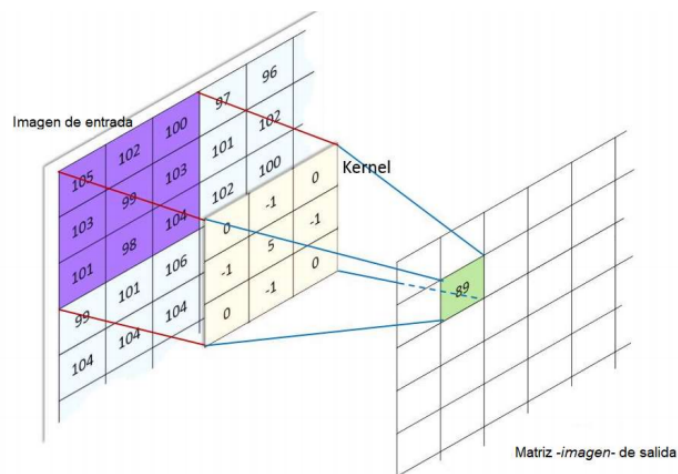
## Capas de convolución

Esta capa esta constituida por neuronas convolucionales, cada una de estas estan asociadas a un filtro no necesariamente igual en todas las neuronas de la capa, cada neurona convolucional presenta una función de activación en la redes neuronales convolucionales, utilizamos *Relu* por las siglas del ingles Unidad Neuronal Rectificada.

La función de activación *Relu* esta expresada por la siguiente fórmula matemática, así mismo existe otra función de activación denominado Softplus(Pablo Pastor, 2018).

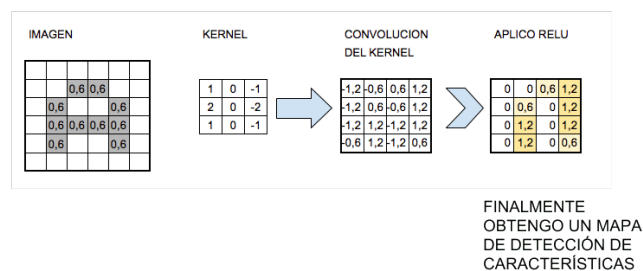
$$Relu : f(x) = \max(0, x). \quad softplus : f(x) = \ln(1 + e^x) \quad (2.22)$$

Figura 2.18: Tranformación de convolución en una neurona.



Fuente: Cantero Lorenzo (2017), Figura 2.4 Ejemplo de operación de convolución

Figura 2.19: Aplicación de una convolución y Relu.



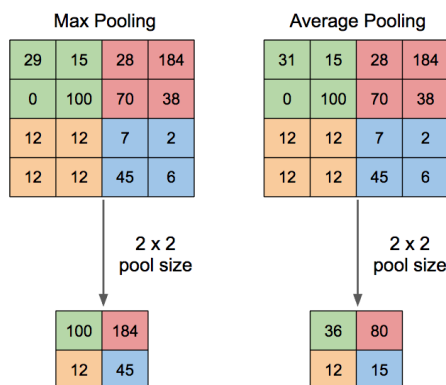
Fuente: <https://il.wp.com/www.aprendemachinelearning.com/wp-content/uploads/2018/11/CNN-04.png>

## Capas de reducción de muestreo

Es la encargada de reducir paulatinamente el consumo espacial de la representación de una imagen, así mismo el número de parámetros y el costo computacional de la red, una de las técnicas que se puede aplicar a la reducción de

muestreo es el *max pooling*, consiste en elegir la activación máxima de la región seleccionada, también existen otras técnicas tales como *average pooling* y *L2 – normpooling*(Pablo Pastor, 2018).

Figura 2.20: Max pooling y average pooling



Fuente: [https://www.researchgate.net/figure/Illustration-of-Max-Pooling-and-Average-Pooling-Figure-2-above-shows-an-example-of-max\\_fig2\\_333593451](https://www.researchgate.net/figure/Illustration-of-Max-Pooling-and-Average-Pooling-Figure-2-above-shows-an-example-of-max_fig2_333593451)

## Capas de clasificación

Una vez obtenida una estructura que conste de capas de convolución y pooling continúa una capa de clasificación. Esta capa consta de neuronas que suelen recibir las características de las capas convolucionales y de pooling, para predecir que categoría o clase corresponde a la imagen de entrada, caso de clasificación de imágenes(Pablo Pastor, 2018), previo la capa de clasificación generalmente se aplana en ancho, altura y los mapas de características convirtiendolas en la capa de neuronas, al cual se le aplica una función de activación denominado Softmax que se conecta a la capa de salida final, que tiene la cantidad de neuronas correspondiente a las categorías a clasificar(Juan, 2018).

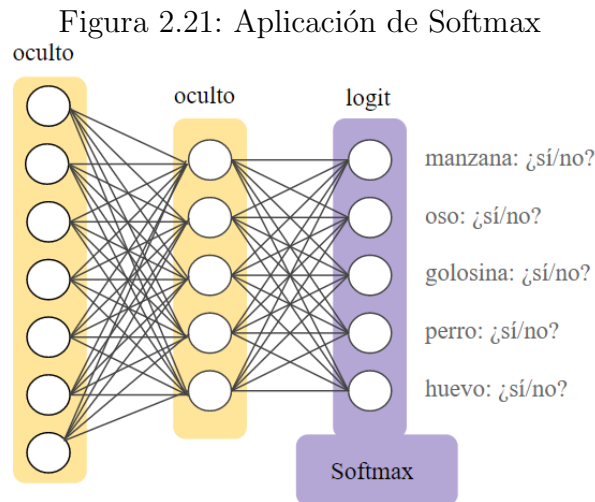
La función softmax es la encargada de pasar a probabilidad entre el rango (0 y 1) a las neuronas de salida. por ejemplo<sup>2</sup>

<sup>2</sup>ver en: <https://developers.google.com/machine-learning/crash-course/multi-class-neural-networks/softmax?hl=es-419>

Tabla 2.6: Ejemplo de probabilidades por clase para la capa Softmax

Clase	Probabilidades
Manzana	0.001
Oso	0.04
Caramelos	0.008
Perro	0.95
Huevo	0.001

Fuente: Elaboración Propia



Fuente: Redes neuronales de clases múltiples: Softmax, Figura 2

### 2.3.3. Evaluación

Para el proceso de evaluación del modelo construido se optó por la técnica de entropía cruzada que también fue usado para la evaluación de redes neuronales convolucionales (David Escobedo, 2017), que es aplicado en problemas de clasificación de imágenes, en el entorno de investigación que tenemos usamos la entropía cruzada categórica debido a la **existencia de múltiples clases y una sola etiqueta, la métrica usada es la exactitud más conocido como accuracy**, para definir la exactitud precisamos de conceptos de matriz de confusión que corresponde a la clase que se predijo y a la clase verdadera como se muestra en la **figura 2.22**, “la matriz de confusión es una tabla de un clasificador de dos a más clases que contiene información de las clasificaciones actuales y las predichas por el sistema de clasificación” (Sanchez Garrido, 2012).

Sea el caso considerado, donde se evalúan los valores para la clase A que representa a una determinada enfermedad, a esta se le considera como verdadera y todas las demás como falsas, a continuación se describe cada uno de los parámetros.

- Verdadero Positivo (TP).- Casos que pertenecen a una clase, y el clasificador lo define en esa clase.
- Falso Positivo (FP).- Casos que no pertenecen a una clase, y el clasificador lo define en esa clase.

Figura 2.22: Matriz de confusión en un entorno multiclase

Clase \ Clase predicha \ verdadera	A	B, C, D, etc.
A	Verdadero positivo (TP)	Falso positivo (FP)
B, C, D, etc.	Falso negativo (FN)	Verdadero negativo (TN)

Fuente: Sanchez Garrido (2012)

- Falso Negativo(FN).- Casos que pertenecen a esa clase, y el clasificador no lo define en esa clase.
- Verdadero Negativo(TN).- Casos que no pertenecen a esa clase, y el clasificador no lo define en esa clase.

La métrica que se usa es la exactitud y esta definido como la proporción del número total de predicciones correctas definido por la siguiente ecuación, en caso de clasificación multiclase se usa la exactitud promedio donde C es el número de clases(Zamorano Ruiz, 2018).

$$Exactitud = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.23)$$

$$ExactitudPromedio = \frac{1}{C} \sum_{i=1}^C \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i} \quad (2.24)$$

Softmax es una función que parametriza a un vector asignándole valores entre 0 y 1, donde la sumatoria del vector es 1, aplicados a las puntuaciones de salida  $s$ , también representados como probabilidades de clase <sup>3</sup>.

$$f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}} \quad (2.25)$$

Donde:

- C: Conjunto de clases.
- $s_i$  definido como la clase  $i$ .
- $s_j$  puntaje inferido por la red para cada clase de C.

La entropía cruzada categórica es usado en problemas de clasificación multiclase en caso de redes neuronales, consiste en asignar una probabilidad a cada clase para todos los ejemplos, si existen N muestras que pertenecen a M clases, la entropía cruzada categórica esta definido como una función de pérdida que viene

<sup>3</sup>ver en: [https://gomburu.github.io/2018/05/23/cross\\_entropy\\_loss/](https://gomburu.github.io/2018/05/23/cross_entropy_loss/)

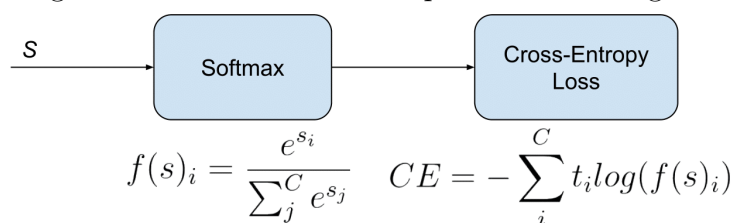
a ser una activación de Softmax más una entropía cruzada, que esta representada por:(Rahul, 2019)

$$LogarithmicLoss = \frac{-1}{N} \sum_{i=1}^N \sum_{j=1}^M y_{ij} * \log(p_{ij}) \quad (2.26)$$

Donde:

- $y_{ij}$  Es 1, si el ejemplo  $i$  pertenece a la clase  $j$  caso contrario es 0.
- $p_{ij}$  Es la probabilidad que nuestro clasificador predice de la muestra  $i$  perteneciente a la clase  $j$ .

Figura 2.23: Pérdida de entropía cruzada categórica



Fuente: [towardsdatascience.com](https://towardsdatascience.com)

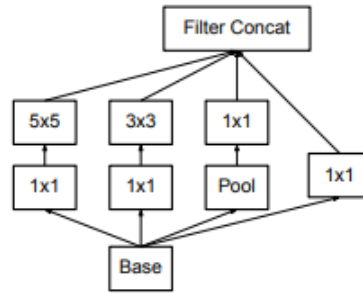
## 2.4. Propuesta de arquitecturas de redes neuronales convolucionales

### 2.4.1. InceptionV3

El módulo de InceptionV3 esta constituido por cuatro ramas en paralelo, primero una convolución de filtro cuyo tamaño es 1x1 seguida de dos convoluciones de tamaño 3x3, una convolución de tamaño 1x1 seguida de una convolución de tamaño 3x3, un pooling seguido de una convolución 1x1 y por último una convolución 1x1. “El resultado del módulo esta conformado por la concatenación de las cuatro ramas, constituido por un total de 10 módulos que conformar la arquitectura completa de InceptionV3 con ciertas modificaciones ligeras según la profundidad de la red. Se cambia ligeramente 5 módulos con el fin de reducir el coste computacional sustituyendo las convoluciones nxn por dos convoluciones, una 1x7 seguida de una 7x1. Los dos últimos módulos sustituyen las dos últimas convoluciones de 3x3 de la primera rama por dos convoluciones cada una, 1x3 seguida de otra 3x1, esta vez en paralelo. En total, InceptionV3 tiene 42 capas con parámetros” (Gómez-Ríos et al., 2019).

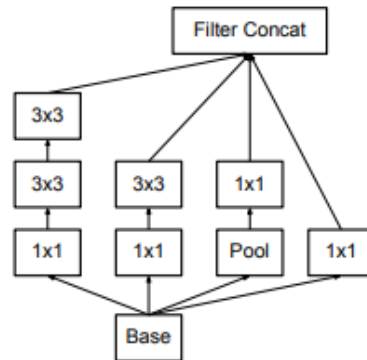


Figura 2.24: Módulo de la arquitectura Inception original.



Fuente: Szegedy (2016), Figura 4

Figura 2.25: Módulo de Inception donde el filtro de la convolución 5x5 es reemplazado por dos filtros de convolución de 3x3



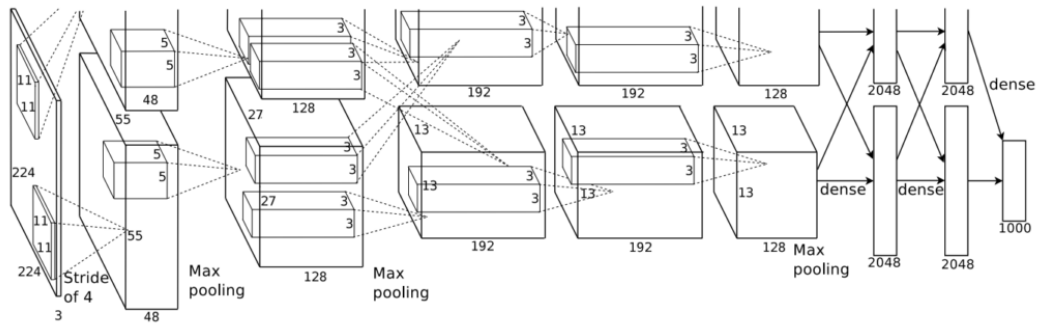
Fuente: Szegedy (2016), Figura 5

## 2.4.2. AlexNet

Este modelo consta de cinco capas convolucionales en la parte de extracción de características y tres capas completamente conectadas en la parte clasificadora del modelo. Las imágenes de entrada se ajustan al tamaño 224x224 con tres canales de color. “En términos del número de filtros utilizados en cada capa convolucional. Se utiliza el patrón de disminución del tamaño del filtro con la profundidad. Comenzando desde el tamaño más pequeño de 11x11 y disminuyendo a 5x5, y luego a 3x3 en las capas más profundas. El uso de filtros pequeños como 5x5 y 3x3 es ahora la norma. Se utiliza un patrón de una capa convolucional seguida de una capa de agrupamiento al principio y al final de la parte de detección de características del modelo” (Rizwan, 2018b).

“El modelo se entrena con el aumento de datos. Lo que aumenta artificialmente el tamaño del conjunto de datos de entrenamiento brindando mayor recurrencia para aprender las mismas características en diferentes condiciones” (Rizwan, 2018b).

Figura 2.26: Modelo Alexnet



Fuente:

<http://sitiobigdata.com/wp-content/uploads/2019/05/Innovaciones-arquitect%C3%B3nicas-en-redes-neuronales2.png>

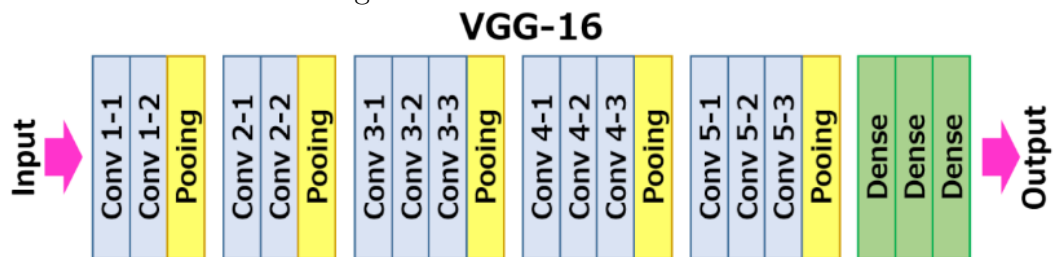
- Primera capa: La entrada para AlexNet es una imagen RGB de 224x224x3 que pasa a través de la primera capa convolucional con 96 mapas de características o filtros que tienen un tamaño de 11x11 y una zancada de 4. Las dimensiones de la imagen cambian a 55x55x96. Luego, AlexNet aplica la capa de agrupación máxima o la capa de submuestreo con un tamaño de filtro de 3x3 y una zancada de 2. Las dimensiones de la imagen resultante se reducirán a 27x27x96.
- Segunda capa: A continuación, hay una segunda capa convolucional con 256 mapas de características que tienen un tamaño de 5x5 y un paso de 1. Luego, nuevamente hay una capa de agrupación máxima con un tamaño de filtro de 3x3 y una zancada de 2. Esta capa es la misma que la segunda capa, excepto que tiene 256 mapas de características, por lo que la salida se reduce a 13x13x256.
- Tercera, cuarta y quinta capas: Las capas tercera, cuarta y quinta son capas convolucionales con un tamaño de filtro de 3x3 y un paso de 1. Los dos primeros utilizaron 384 mapas de características, mientras que el tercero utiliza 256 filtros. Las tres capas convolucionales son seguidas por una capa de agrupación máxima con un tamaño de filtro de 3x3, un paso de 2 y 256 mapas de características.
- Sexta capa: La salida de capa convolucional se aplanada a través de una capa completamente conectada con mapas de características 9216 cada uno de tamaño 1x1.
- Séptima y octava capa: Lo siguiente es nuevamente dos capas completamente conectadas con 4096 unidades.
- Capa de salida: Finalmente, hay una capa de salida Softmax y con 1000 valores posibles que tienen un valor de probabilidad(Rizwan, 2018a).

### 2.4.3. VGG-16

VGG-16 es una arquitectura de red neuronal que fue propuesto por K. Simons y A. Zisserman alcanzando una precisión de 92.7% en la prueba de ImageNet del año 2014, ImageNet viene a ser un conjunto de datos representado por 15 millones de imágenes con 22000 categorías aproximadamente, a comparación de la arquitectura AlexNet este consta de kernels de dimensiones 11 y 3 en la primera y la segunda capa convolucional respectivamente y múltiples filtros de tamaño 3x3(ul Hassan, 2018). El nombre VGG-16 proviene del hecho de tener 16 capas, ya sean de convolución, max pooling, capas de activación y capas completamente conectadas, las características se detallan a continuación.

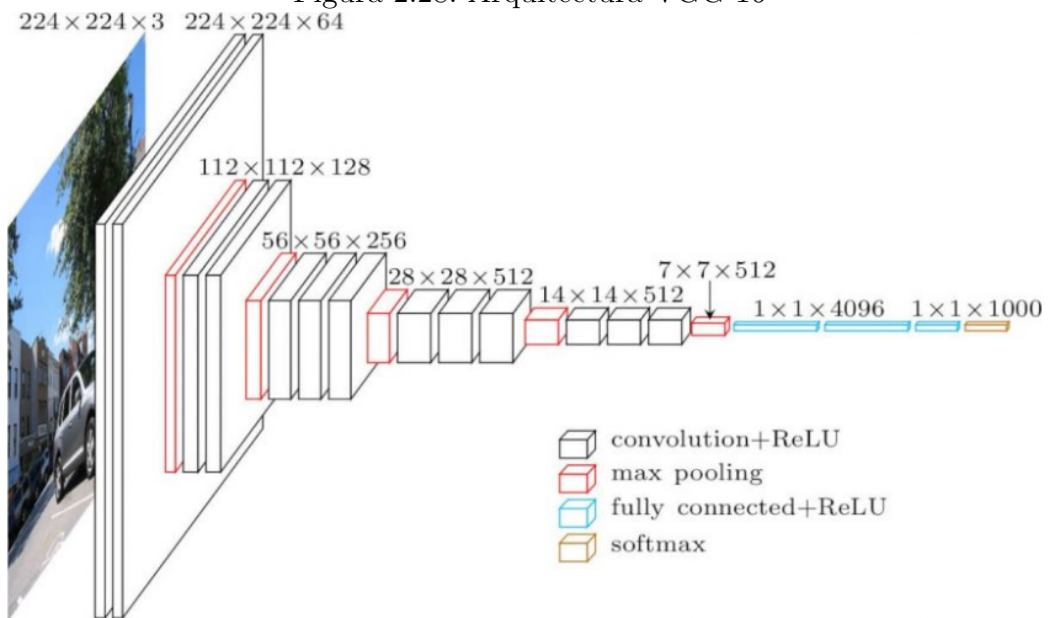
Esta arquitectura cuenta con 13 capas convolucionales, 5 capas de max pooling y 3 capas densas que suman hasta 21 capas, con la diferencia de 16 capas de peso. Conv 1 tiene 64 filtros, Conv 2 tiene 128 filtros, Conv 3 tiene 256 filtros, Conv 4 y Conv 5 tiene 512 filtros. La red VGG-16 está entrenada en el conjunto de datos ImageNet que tiene más de 14 millones de imágenes y 1000 clases, y alcanza un 92.7% de precisión de top 5. Supera la red AlexNet al reemplazar filtros grandes de tamaño 11 y 5 en la primera y segunda capas de convolución con filtros pequeños de tamaño 3x3(Tewari, 2019).

Figura 2.27: Modelo VGG-16



Fuente: <https://neurohive.io/wp-content/uploads/2018/11/vgg16.png>

Figura 2.28: Arquitectura VGG-16



Fuente: <https://neurohive.io/wp-content/uploads/2018/11/vgg16-neural-network.jpg>

# Capítulo 3

## Construcción del prototipo

Para la construcción de un prototipo de sistema de clasificación de enfermedades en hojas del cafeto basado en visión computacional, se requiere construir un set de datos de imágenes y una arquitectura de red neuronal convolucional para así cumplir con los objetivos de clasificación de la enfermedad.

Para hacer posible el entrenamiento de nuestra red neuronal convolucional optamos por crear el conjunto de datos que contiene 2200 imágenes que corresponden a las enfermedades de *Hemilea Vastatrix*, *Mycena Citricolor* y *Leucoptera Coffeella*, en caso de *Hemilea Vastatrix* existen diferentes grados de severidad enfocado directamente a la hoja el cual tiene un tratamiento especial para el cálculo de la severidad, capítulo 3.1.4.

- **Hemilea Vastatrix.**- Conocido comúnmente como Roya de Café Amarilla, es el término más usado por los especialistas y caficultores (Provincia de la Convención), enfermedad con más recurrencia en los cultivos.
- **Mycena Citricolor.**- Conocido comúnmente como Ojo de Pollo (Provincia de la Convención), enfermedad con alta recurrencia en los cultivos.
- **Leucoptera Coffeella.**- Conocido comúnmente como Minador de Café (Provincia de la Convención), enfermedad con poca recurrencia.

Se tiene 1000 imágenes por enfermedad que contiene una única enfermedad<sup>1</sup>, excepto por *Leucoptera Coffeella* que tiene 200 imágenes por la poca recurrencia, estos se distribuyen tanto en datos de entrenamiento y validación. Las imágenes son tomadas en su mayoría hoja completa y en fondo blanco (sobre un papel bond A4) para facilitar la segmentación de la región sana y enferma en el caso de ser el resultado de la clasificación *Hemilea Vastatrix*.

Para capturar la imagen a ser procesada (clasificada en su categoría correspondiente), la hoja del cafeto debe estar en una superficie plana y color blanco de preferencia hoja bond A4 y a su vez la hoja del cafeto debe estar lo más plano posible.

---

<sup>1</sup>Durante la captura de imágenes se pudo apreciar pocos casos donde una hoja de cafeto podía tener dos enfermedades, para el tratamiento de estos casos se utiliza clasificación multi-etiqueta, por la naturaleza de este trabajo y la presencia masiva de las enfermedades de manera única en la hoja se utiliza clasificación multi-clase, por otro lado este trabajo esta centrado en contemplar la clasificación de enfermedades en las hojas del cafeto que tengan mayor recurrencia

Para el desarrollo del proyecto, una vez obtenida la imagen de entrada compuesta por la enfermedad se considera cinco canales de proceso que esta constituido por las fases de preprocesamiento de la imagen, clasificación de la enfermedad, definición de la información técnica, segmentación y cálculo del grado de severidad de daño en la hoja; cada una de estas fases constan de una serie de procedimientos que se muestra en la siguiente tabla.

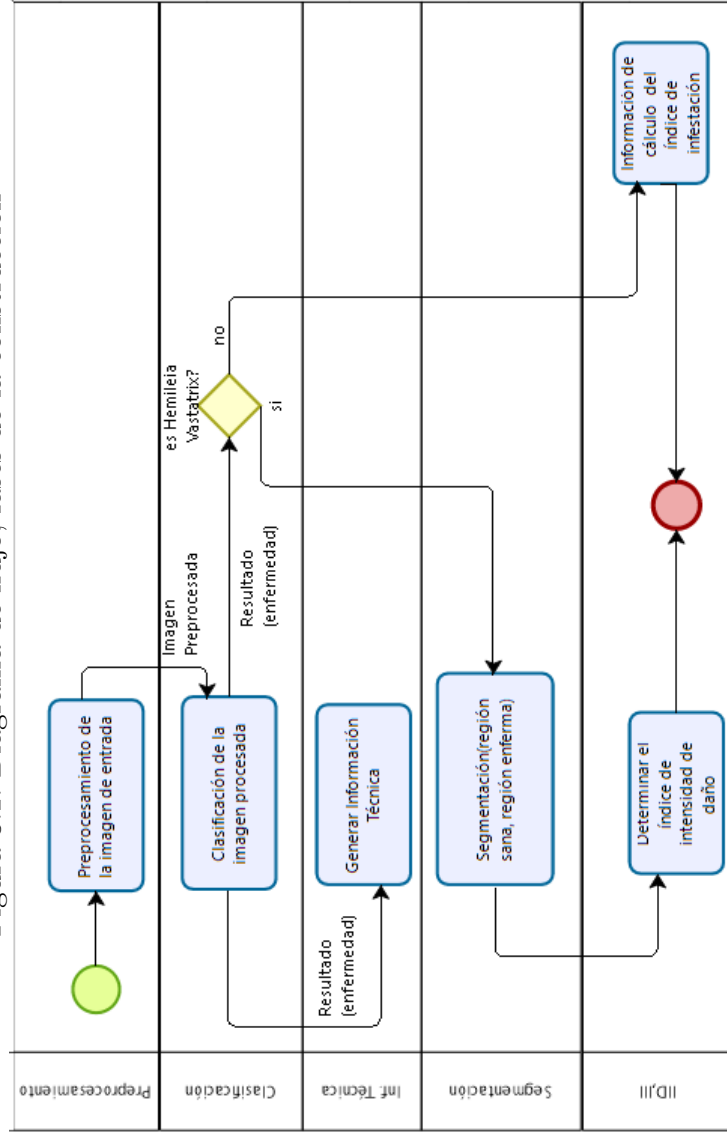
Tabla 3.1: Procedimiento para la construcción del prototipo

<b>Fases</b>	<b>Descripción</b>
Preprocesamiento	Un vez obtenida la imagen de entrada se aplica operaciones puntuales tales como; ecualización del histograma, filtrado de la imagen, suavizado de la imagen y filtro de la mediana.
Clasificación de la enfermedad	Al dataset de imágenes se aplica las arquitecturas propuestas tales como; red neuronal convolucional diseñada, AlexNet, VGG-16, InceptionV3 para realizar el proceso de entrenamiento y su correspondiente predicción.
Información técnica	Se obtiene en base a la clasificación de la enfermedad mostrando datos como; biología, agente causal, daño y control de la enfermedad.
Segmentación	Haciendo uso de segmentación por umbralización y segmentación por tonalidad de color.
Grado de severidad en la hoja	Se obtiene de acuerdo al cálculo mencionado en el apartado de la Roya de Café, capítulo 2.1.4.

**Fuente: Propia**

### 3.1. Fases de la construcción

Figura 3.1: Diagrama de flujo, fases de la construcción

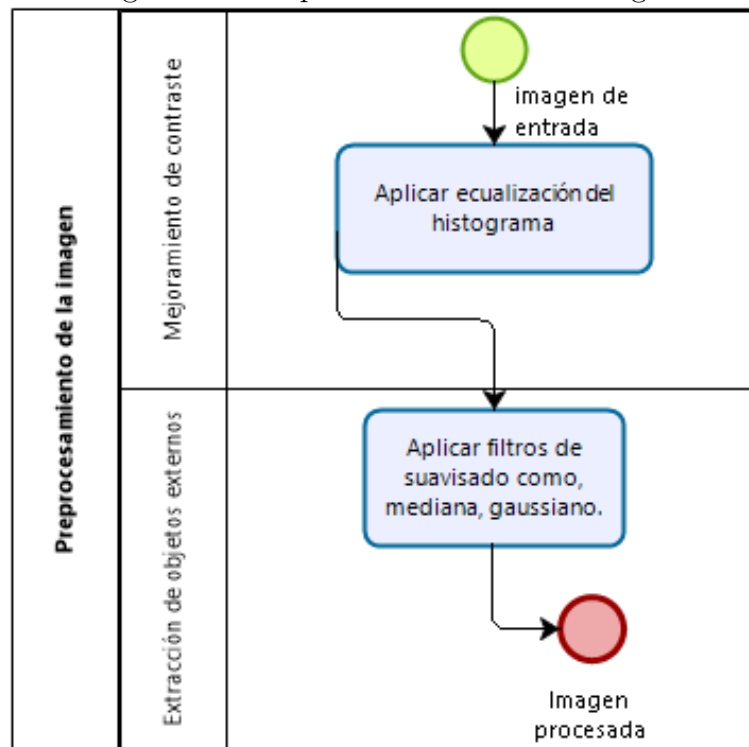


Fuente: Propia

### 3.1.1. Preprocesamiento de la imagen

Consiste en el uso de técnicas de mejoramiento aplicado a la imagen de entrada a ser procesada, para así tener una imagen en buenas condiciones y nuestra clasificación sea lo más adecuado al resultado real. Durante la captura de datos (imágenes) se identificó casos donde la hoja del café presenta ciertos cúmulos de suciedad o tierra en la superficie, lo que puede ocasionar confusión con la enfermedad durante su clasificación, así mismo se pudo apreciar que algunas imágenes no tenían claridad y contraste; para abordar estas deficiencias se utilizan técnicas de ecualización del histograma para mejorar el contraste y filtros de suavizado como la mediana para extraer los cúmulos que no pertenecen a la enfermedad en la hoja, se muestra los procedimientos aplicados en la figura 3.2.

Figura 3.2: Preprocesamiento de la imagen



Fuente: Propia

#### Ecualización del histograma

El proceso de ecualización básicamente es usado para mejorar el contraste de algunas imágenes que presentan poca claridad para la clasificación de la enfermedad, de esta manera los detalles o características en la imagen tienen mayor comprensión para optimizar los resultados.

Una vez obtenida la imagen en su representación bidimensional (matriz) de longitud  $N \times M$  donde  $N$  es la longitud de la imagen y  $M$  es la altura de la imagen, tenemos que inicializar una nueva matriz (imagen resultado de la ecualización) con las mismas dimensiones de la imagen de entrada definiendo los valores iniciales de píxeles en cero. Recorremos la matriz de la imagen de entrada y almacenamos en una variable el valor obtenido de la frecuencia del histograma representado



por la formula 2.10 de capítulo 2.2.1 y es almacenando en la posición respectiva de la matriz de ecualización del histograma la frecuencia de intensidad calculada. Haciendo uso de las librerías OPENCV en python logramos ecualizar la imagen mediante las funciones *cv2.equalizedHist* y *cv2.cvtColor* que es usado para aplicar la ecualización a una imagen de color en formato RGB capítulo 2.2.1, representado por el siguiente algoritmo 1.

---

**Algorithm 1** Cálculo de la función de ecualización del histograma

---

**Require:** *ImagenIn*: Matriz [0,1,...,255] tipo entero

**Ensure:** *Hist*: Matriz [0,1,...,255] de byte

```

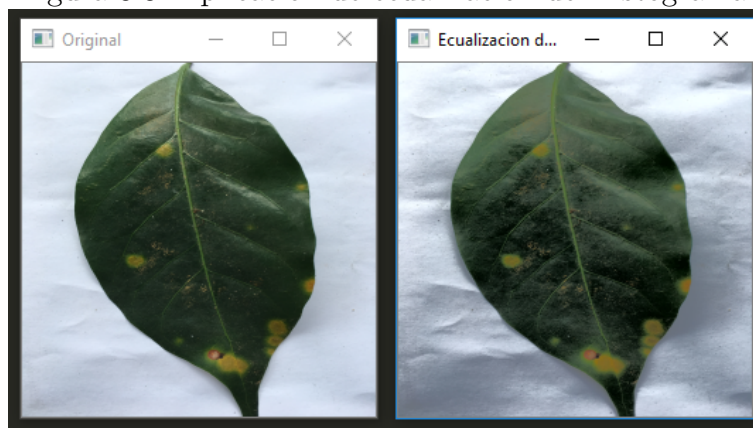
1: Hist[256] = 0
2: for i = 0; i < 256; i ++; do
3:   Hist[i] = 0
4: end for
5: for r = 0; r < ImagenIn.rows r ++; do
6:   for c = 0; c < ImagenIn.cols c ++; do
7:     ind = input.at < unsignedchar > (r, c)
8:     Hist[ind] = Hist[ind] + 1;
9:   end for
10: end for

```

---

**Fuente:** Solar (2015), Pag 34

Figura 3.3: Aplicación de ecualización del histograma.



**Fuente:** Propia

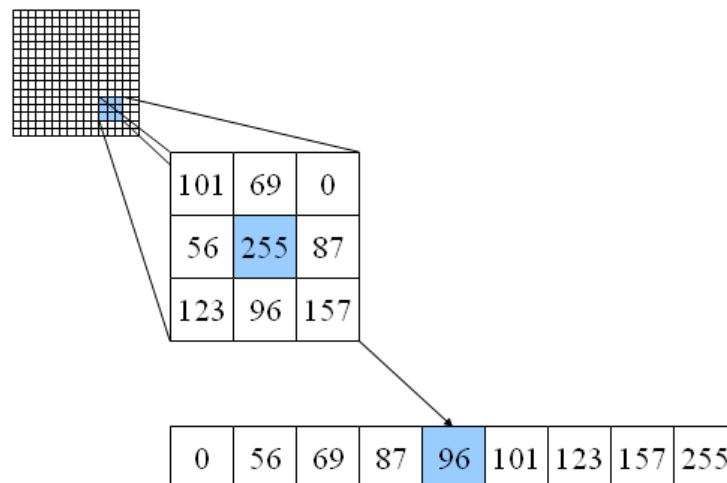
### Filtro de la mediana

Es usando básicamente para poder mejorar la imagen de la hoja, debido a que existen componentes externos a la hoja como cúmulos de tierra, polvo, etc que no forman parte de las enfermedades, de esta manera nos enfocamos solo en la enfermedad evitando confusiones en el proceso de clasificación. Esta técnica es la que menos difumina los bordes y no altera demasiado las características de la hoja, en nuestro caso para mejorar la imagen lo más adecuado es usar el filtrado de la mediana y es lo que utilizamos en la construcción del prototipo.

El filtro de la mediana esta catalogada para la eliminación del ruido conocido como impulso sal y pimienta, cuyos valores de los pixeles afectados por él son muy diferentes en color e intensidades almacenados por los pixeles cercanos que tiene una peculiaridad similar a algunas partículas que se encuentran sobre la hoja y es lo que hace fácil su eliminación utilizando el filtro de la mediana(Fuentes, 2015).

Para aplicar el filtro de la mediana se tiene que representar la imagen en una matriz bidimensional y definir el tamaño del filtro generalmente número entero impar, para el cálculo del valor del pixel se superpone el filtro que toma los mismos valores de la imagen como una ventana deslizante sobre la imagen, se ordenan los valores que cubre el filtro para obtener el valor central que reemplaza a la posición central de la imagen que es proyectada por el filtro; este proceso se realiza hasta recorrer toda la matriz de la imagen como se muestra en la figura 3.4.

Figura 3.4: Filtro de la mediana



Fuente: <https://blitzman.gitbooks.io/cuda/pr%C3%A1ctica-2-filtro-mediana.html> Figura

1

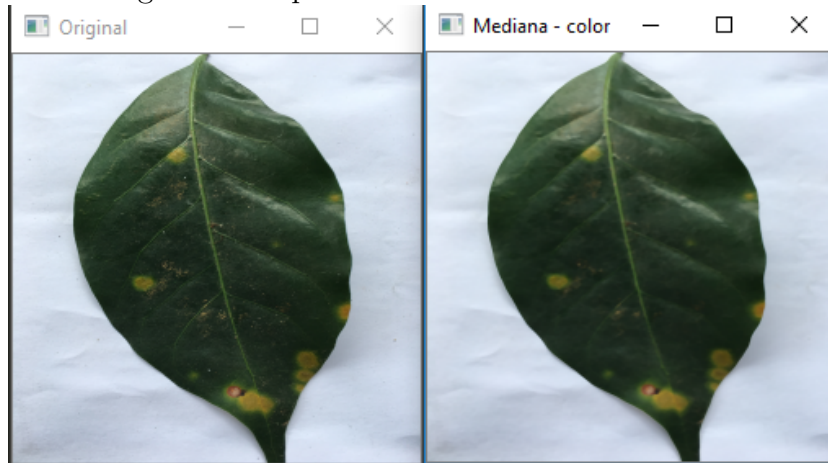
El tamaño del filtro de la mediana depende mucho de las dimensiones que tenga nuestra imagen de entrada, en caso las dimensiones mantengan 3024 x 4032 se recomienda utilizar un filtro de tamaño 27, en nuestro caso para estandarizar las dimensiones de las imágenes se hace un reajuste a 150 x 150 trabajando con un filtro de tamaño 3 donde la eliminación del ruido se aplica de manera adecuada; la función que se utiliza para el filtro de la mediana es *medianblur* de la librería OPENCV con parámetros una imagen y el tamaño del filtro, obteniendo como resultado la figura 3.5.

```

img=ruta de la imagen de entrada
imagen1 = cv2.resize(img,(150,150), interpolation = cv2.INTER_AREA)
aplicar filtro de mediana de tamaño 3
imagen_procesado = cv2.medianBlur(imagen1,3)

```

Figura 3.5: Aplicación del filtro de la mediana



Fuente: Propia

### Filtro Gaussiano

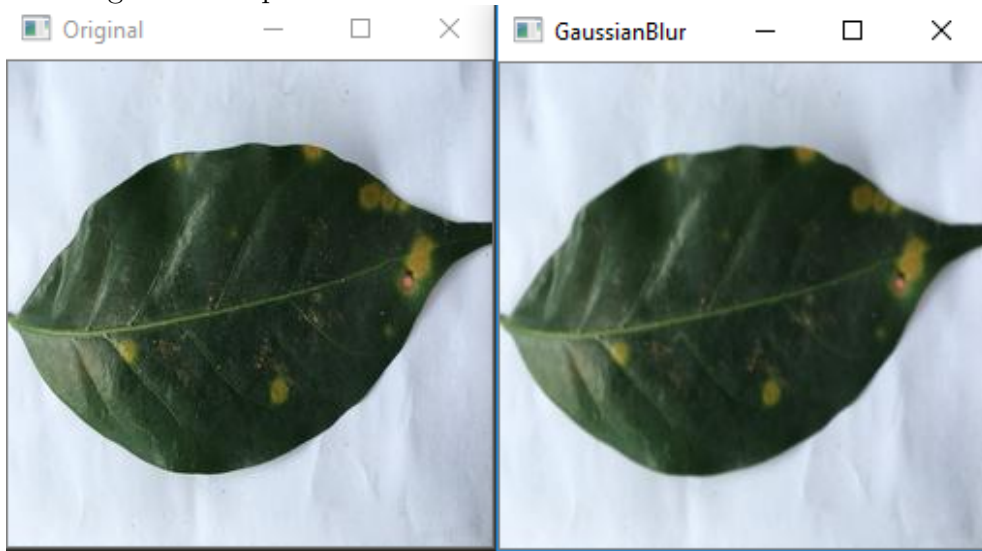
Es usado para la eliminación del ruido sobre la imagen dependiendo de la variación de la desviación típica, en caso este incremente elimina más el ruido pero difumina la imagen, este algoritmo no es el adecuado debido a que existe la posibilidad de perder información de interés que puede ser útil en la etapa de clasificación, solo en el caso de asignar un valor menor a la desviación típica elimina en poca cantidad el ruido sin perder información; se esta utilizando las librerías de python tales como *GaussianBlur* definiendo un valor de 9 para *desviación*.

Este tipo del filtro tiene el problema de difuminar los bordes de la imagen y reduce especialmente el ruido tipo Gaussiano producido por pequeñas variantes en la imagen originado por diferentes ganancias de sensor, ruido en la digitalización, etc; este filtro Gaussiano es aplicado de la siguiente manera.

```
imagen_filtrada = imagen_original.filter(ImageFilter.GaussianBlur(desviacion))
```

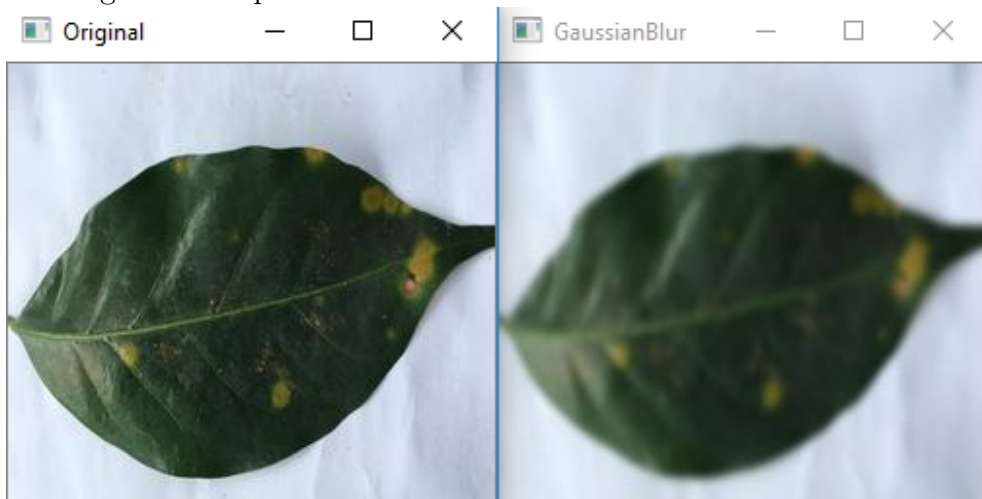
En el presente trabajo para la construcción del prototipo en la fase de preprocesamiento no utilizamos el filtro Gaussiano, solo se utiliza con fines demostrativos porque al aplicar el filtro la imagen de salida no mantiene las características y pierde información en los bordes (Jhosi, 2015). La función *GaussianBlur* recibe como parámetro la desviación, y depende del valor que este tome para poder ver los resultados que se muestran a continuación.

Figura 3.6: Aplicación del filtro de Gauss con desviación de 9



Fuente: Propia

Figura 3.7: Aplicación del filtro de Gauss con desviación de 27



Fuente: Propia

Como se puede ver en la figura 3.6 con una desviación menor no difumina la imagen pero tampoco lo mejora y extrae los objetos de la superficie de la hoja, de lo contrario cuando el valor de la desviación es muy alta, figura 3.7 difumina la imagen y se pierde el relieve y la textura que tiene la enfermedad en la hoja; si comparamos con el filtro de la mediana figura 3.5 veremos que este tiene mejores resultados y es por ello la utilización en este proyecto. El filtro Gaussiano es aplicado a través de un operador de convolución bidimensional, además de un filtro expresado de la siguiente manera:

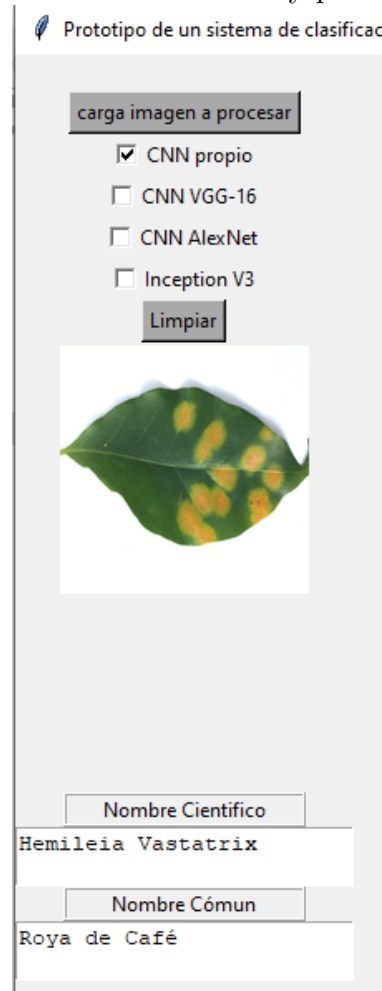
$$g(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.1)$$

### 3.1.2. Clasificación de la imagen

Para efectuar el proceso de clasificación de una enfermedad se precisa de una imagen de entrada que es preprocesada aplicando ecualización del histograma para mejorar el contraste y filtro de la mediana para eliminar el ruido. Posterior a ello es procesada por un modelo de red neuronal convolucional pre-entrenado para concretar el proceso de clasificación de la enfermedad. Es necesario realizar el proceso de entrenamiento de la red neuronal convolucional y se necesita un conjunto de imágenes etiquetadas distribuidas en datos de entrenamiento y validación, Anexo B.

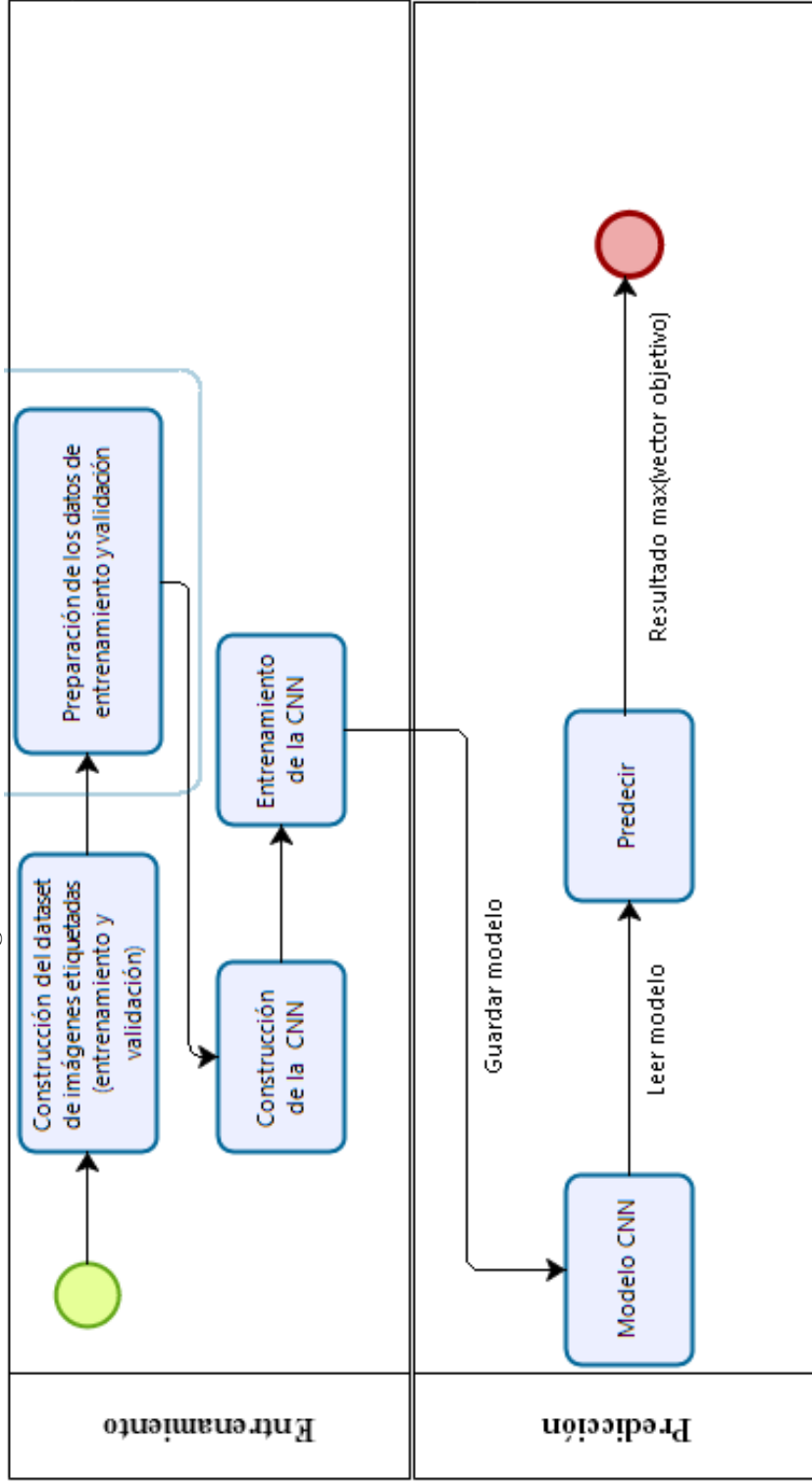
Así mismo se necesita de un módulo para realizar el proceso de predicción, donde se usa el modelo de red neuronal diseñado y entrenado que fue guardado como un archivo con extensión *.h5*, este proceso cuenta con la función Softmax para poder identificar la enfermedad que corresponde a la predicción a través del vector objetivo que contiene las probabilidades de similitud relacionado a cada enfermedad.

Figura 3.8: Clasificación y predicción



Fuente: Propia

Figura 3.9: Fases de la clasificación



Fuente: Propia

En la figura 3.9 se muestran las etapas que surgen para construir el clasificador de enfermedades basado en redes neuronales convolucionales y a continuación se describen cada una de las etapas.

- El proceso de construcción del conjunto de imágenes etiquetadas es previa a la clasificación de enfermedades, estos corresponden a los grupos de entrenamiento y validación, están estructurados como se muestra en el (**Anexo B**).
- Preparación del conjunto de datos de entrenamiento y validación que consiste, primero, en definir las dimensiones de las imágenes que serán procesados con una altura de 150 y una longitud de 150 y facilitar el procesamiento, también definimos la cantidad de 32 imágenes que son procesados en cada paso almacenado en la variable *batch\_size*. Para la preparación de datos de entrenamiento y validación se realiza un reajuste de valores de los píxeles comprendidos entre 0 y 1 de esta manera se facilita las operaciones y cálculos almacenado en la variable *rescale*, se aplica algunas funciones para generar datos con inclinación (*shear\_range*), zoom e invertidas (*zoom\_range*, *horizontal\_flip*), todo ello para que el entrenamiento se realice en diferentes condiciones.

Así mismo se define el modo de clasificación **Categorical** ya que se cuenta con más de dos clases y los parámetros iniciales para la red neuronal son los que se muestra a continuación.

```
data_entrenamiento = './entrenamiento'  
data_validation = './validation'  
longitud, altura = 150, 150  
batch_size = 32  
  
#Generador de datos de entrenamiento  
entrena_datagenerador = ImageDataGenerator(  
    rescale=1./255,  
    shear_range=0.2,  
    zoom_range=0.2,  
    horizontal_flip=True)  
testval_datagen = ImageDataGenerator(rescale=  
1./255)  
entrenamiento_generador =  
entrena_datagenerador.flow_from_directory(  
    data_entrenamiento,  
    target_size=(altura, longitud),  
    batch_size=batch_size,  
    class_mode='categorical')  
validacion_generador = testval_datagen.flow_from_  
_directory(data_validation,  
    target_size=(altura, longitud),  
    batch_size=batch_size,  
    class_mode='categorical')
```

Fuente: AMP-Tech, CNN desde cero, train.py

- El proceso de construcción de la red neuronal convolucional esta definido en el apartado **3.1.2** y modelos propuestos en los apartados **3.1.2**
- Una vez contruido el modelo se procede con el entrenamiento de la red neuronal convolucional a través del siguiente codigo, haciendo uso de los parámetros iniciales definidos aplicados a la red neuronal construida.

```
def entrenar():
    modeloconvolucional.fit_generator(
        entrena_datagenerador,
        steps_per_epoch=pasos,
        epochs=epocas,
        validation_data=validacion_generador,
        validation_steps=validation_steps)
```

Fuente: AMP-Tech, CNN desde cero, train.py

- Al finalizar el proceso de entrenamiento se procede a guardar el modelo de la red así como los pesos respectivos (archivo con extensión .h5) para su posterior uso durante la predicción.
- para efectuar el proceso de predicción es necesario utilizar el vector objetivo que básicamente es la última capa de la red neuronal convolucional, constituida por 3 neuronas cuyos valores se encuentran entre 0 y 1 correspondiente al porcentaje de similitud (probabilidades de clase). El módulo predecir consiste en leer una imagen, reajustar el tamaño y utilizar la opción **predict** de la CNN cargada para guardar la información de la última capa (Softmax) en un array o vector objetivo **Result**, en el array se elige la posición del valor máximo que representa a la etiqueta del resultado y nos muestra la etiqueta de la enfermedad.

Para poder definir la correspondencia del índice de etiquetas de las clases y el vector objetivo, es suficiente con imprimir la información de índices de la siguiente manera:

```
print(entrenamiento_generador.class_indices)
```

```
{'Hemileia Vastatrix': 0, 'Leucoptera Coffeella': 1, 'Mycena Citricolor': 2}
```

Con esto se define la posición en la que se encuentra cada enfermedad en el vector objetivo y aplicado la función softmax obtenemos el resultado con la posición de mayor probabilidad de similitud.

```
#Leer modelo de red neuronal convolucional y pesos
def predecir(imagen):
    arreglo = cnn.predict(imagen)
    Resultado = arreglo[0]
    respuesta = np.argmax(Resultado)
    if respuesta == 0:
        print("Hemileia Vastatrix")
    elif respuesta == 1:
        print("Leucoptera Coffeella")
```



```

elif respuesta == 2:
    print("Mycena Citricolor")

return respuesta
#predecir('IMG_2345001.png')
Fuente: AMP-Tech, CNN desde cero, predict.py

```

## Redes neuronales convoluciones

Para construir la red neuronal convolucional se plantea una metodología general que se muestra en la siguiente figura 3.10; la normalización consiste en establecer los parámetros y condiciones iniciales, extracción de características a través de capas convolucionales, pooling y activación; para la clasificación se utiliza una capa de Softmax que consta de 3 clases y finalmente se realiza una validación mediante un módulo predecir que recibe como parámetro una imagen.

Figura 3.10: Metodología general para construir el modelo de red neuronal



Fuente: [https://www.researchgate.net/figure/Metodologia-general-de-una-CNN\\_fig7\\_309455781](https://www.researchgate.net/figure/Metodologia-general-de-una-CNN_fig7_309455781)

Para construir la red neuronal convolucional se trabaja con los siguientes parámetros iniciales se considera 5 épocas los cuales están definidos como iteraciones de entrenamiento, pasos está definido como el número de ejecuciones en cada iteración, la variable *filtrosConv1* define el número de neuronas que corresponde a cada filtro en la primera capa, *filtrosConv2* define la cantidad de neuronas correspondientes a cada filtro en la segunda capa, las dimensiones de los filtros de la primera capa es 3 y en la segunda capa es 2, las dimensiones del filtro de pooling es 2, se considera tres clases que corresponden a las enfermedades de *Hemileia Vastatrix*, *Mycena Citricolor* y *Leucoptera Coffeella*.

Finalmente se tiene la variable **lr** que viene a ser la tasa de aprendizaje (learning rate), definido como la rapidez o lentitud con el que un modelo de red neuronal aprende un problema, desafortunadamente no es posible el cálculo de la tasa de aprendizaje de manera analítica o a priori, solo se establece a prueba y error que se encuentran entre los valores de 0 y 1; en nuestro caso se trabaja con un valor de 0.0004 (Zulkifli, 2018).

A continuación se muestra la definición de los parámetros iniciales.

```

epocas=5
longitud, altura = 150, 150
batch_size = 32

```

```

pasos = 1000
validation_steps = 400
filtrosConv1 = 64
filtrosConv2 = 32
tamano_filtro1 = (3, 3)
tamano_filtro2 = (2, 2)
tamano_pool = (2, 2)
clases = 3 # clases de enfermedades
lr = 0.0004

```

Fuente: Propia

El modelo de red neuronal convolucional cuenta con dos capas convolucionales, dos capas de pooling, posterior a ello una capa densa completamente conectada y al finalizar una capa Softmax con tres neuronas, se detallan cada uno de los elementos de la red neuronal.

- La primera es una capa convolucional de 64 neuronas con filtros de tamaño 3x3 más una función de activación **ReLU** (Rectified Linear Unit), estos permiten el paso de todos los valores positivos y aquellos que son negativos se reemplazan por 0; así mismo tiene como entrada la imagen con las dimensiones definidas en los parámetros iniciales. La capa de entrada recibe características específicas de la imagen en cada neurona representado por matrices bidimensionales, en cada recorrido de la imagen se identifican sombras, relieves, bordes, etc que están alojados en cada neurona al cual se aplica filtro de convolución como se muestra en la figura 2.3.2.
- Seguidamente cuenta con una capa de max pooling con un filtro de tamaño 2x2, esta capa reduce el consumo espacial o las dimensiones de la representación matricial de las nuevas imágenes (mapa de características) almacenadas en las neuronas.
- Continuando con otra capa de convolución de 32 neuronas con filtros de tamaño 2x2.
- Finalmente se tiene una capa de maxpooling de tamaño 2x2.

Así mismo el modelo presenta una capa de aplanamiento (**FLATERN**)<sup>2</sup> con 256 neuronas más una función de activación **ReLU** guardando todas las características obtenidas por las capas de convolución y pooling, también cuenta con una capa con función de inactivación **DROPOUT**<sup>3</sup> equivalente al 50 % para así inactivar ciertas neuronas y hacer eficiente el entrenamiento. Finalmente se agrega al modelo una capa densa con 3 neuronas equivalente a las categorías de enfermedades correspondiente a la probabilidad de clase en relación a la imagen de entrada mediante una función de activación **SOFTMAX**.

<sup>2</sup>La capa de aplanamiento consiste en tomar la información de nuestras capas de maxpooling y ponerlo en una columna con el fin de aplanarlo y transferirlo al perceptrón (Crispi, 2019).

<sup>3</sup>Dropout es una función de inactivación de neuronas durante el entrenamiento lo que mejora la generalización porque obliga a su capa a aprender con diferentes neuronas el mismo concepto (Maklin, 2019).

Tabla 3.2: Descripción del modelo de CNN

Dimensión de la matriz	operaciones	Detalles
[150x150x3]	Imagen de entrada	Redimensión
[150x150x64]	convolución 1	64 FM, 3x3
[75x75x64]	Maxpooling 1	64FM
[75x75x32]	convolución 2	32 FM, 3x3
[37x37x32]	Maxpooling 2	32FM
	Capa Flatern	43808 neuronas
	Capa densa	256 neuronas
	Dropout	256 neuronas
	Capa Softmax	3 neuronas

Fuente: Elaboración Propia

La función softmax normaliza el vector objetivo para tener valores comprendidos entre 0 y 1, produciendo una sumatoria de elementos igual a 1, como cada elemento representa una clase puede interpretarse como una probabilidad de clase. Esta función no se puede aplicar a cada  $s_i$ , ya que depende de  $s$ , para una clase  $s_i$ , donde  $s$  es el vector objetivo (Rahul, 2019).

Al finalizar se compila la arquitectura de red neuronal convolucional con una pérdida de entropía cruzada categórica haciendo uso del optimizador de Adam y la métrica Accuracy o exactitud usado para evaluar el rendimiento de la arquitectura.

Continuando con el diseño del modelo de red neuronal almacenado en la variable **cnn** se tiene lo siguiente:

```

cnn = Sequential()
cnn.add(Convolution2D(filtrosConv1, tamaño_filtro1,
padding = "same", input_shape=(longitud, altura,
3), activation='relu'))
cnn.add(MaxPooling2D(pool_size=tamaño_pool))

cnn.add(Convolution2D(filtrosConv2, tamaño_filtro2,
padding = "same"))
cnn.add(MaxPooling2D(pool_size=tamaño_pool))

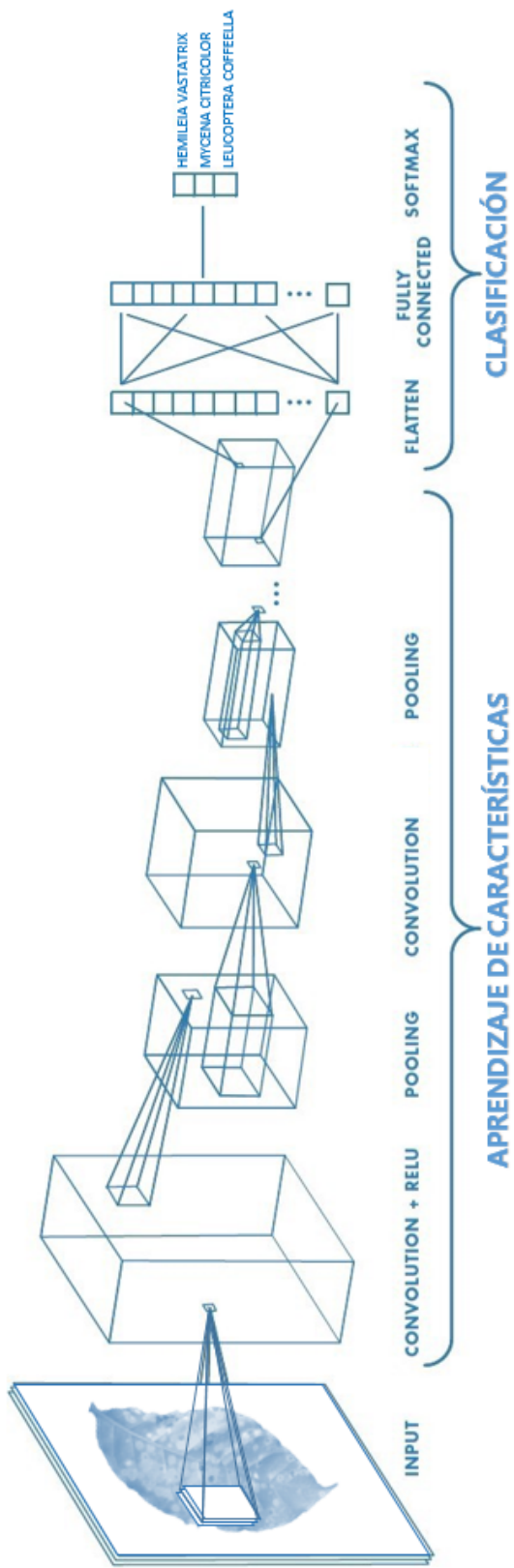
cnn.add(Flatten())
cnn.add(Dense(256, activation='relu'))
cnn.add(Dropout(0.5))
cnn.add(Dense(clases, activation='softmax'))

cnn.compile(loss='categorical_crossentropy',
optimizer=optimizers.Adam(lr=lr),
metrics=['accuracy'])

```

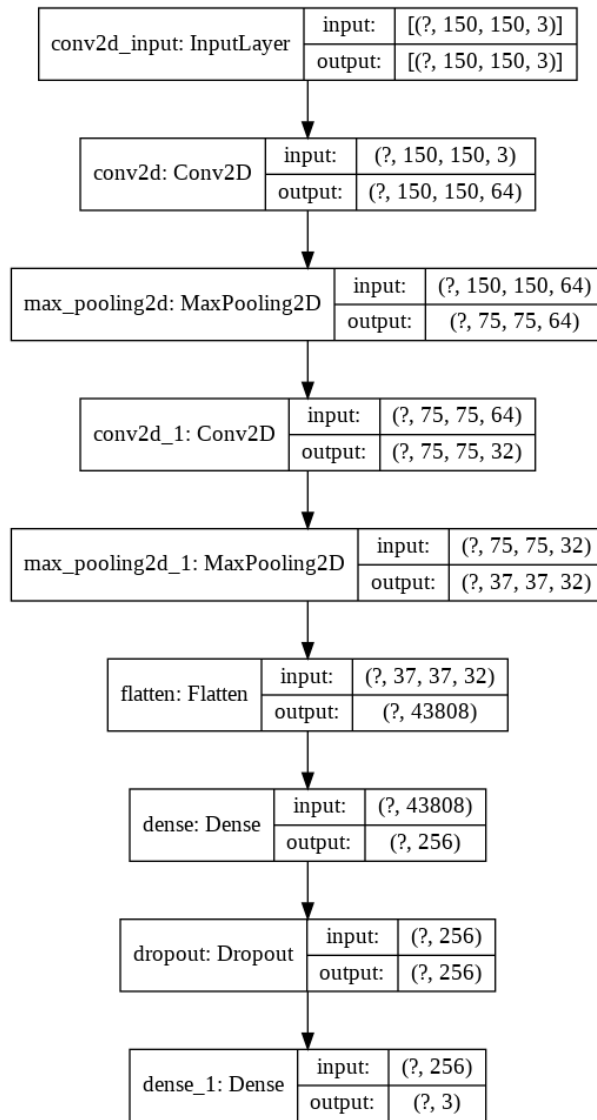
Fuente: Propia

Figura 3.11: Modelo de red neuronal convolucional



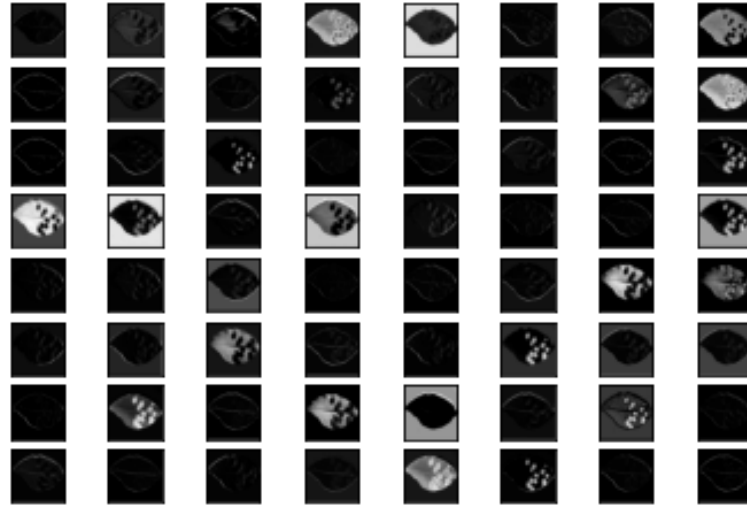
Fuente: Propia

Figura 3.12: Modelo de red neuronal convolucional



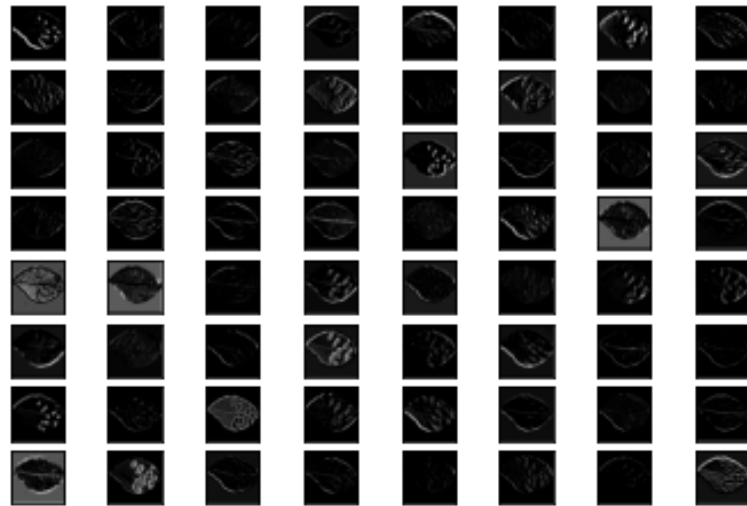
Fuente: Propia

Figura 3.13: Visualización de mapa de características convolución 1



Fuente: Propia

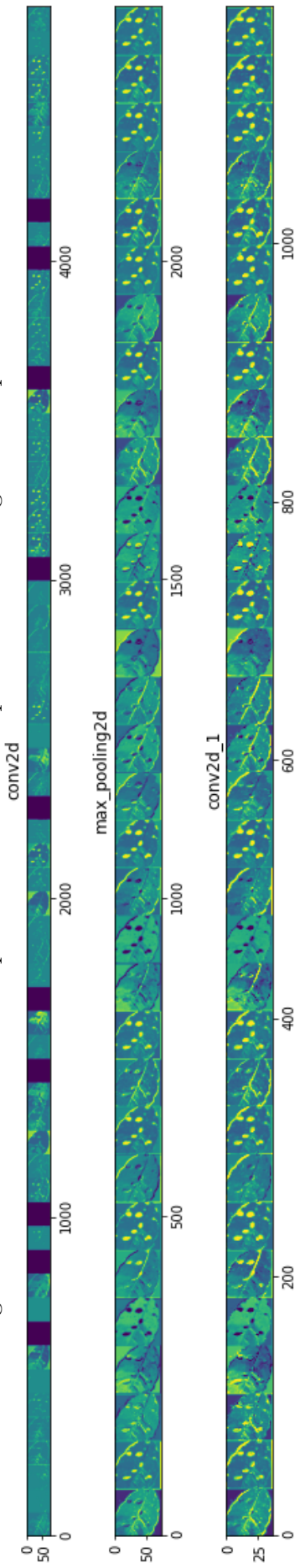
Figura 3.14: Visualización de mapa de características convolución 2



Fuente: Propia

En la figura 3.15 se puede apreciar que en la primera capa convolucional se generó 64 imágenes con filtros aleatorios aplicados, donde cada imagen extrae ciertas características, posterior a ello se generan imágenes reducidas mediante maxpooling (reducción de muestreo); finalmente se tiene la última capa del convolución que genera 32 imágenes a los cuales se les aplica filtros aleatorios, también conocidos como mapas de funciones o mapas de características.

Figura 3.15: Visualización de mapas de funciones o mapas de activación generados por la CNN

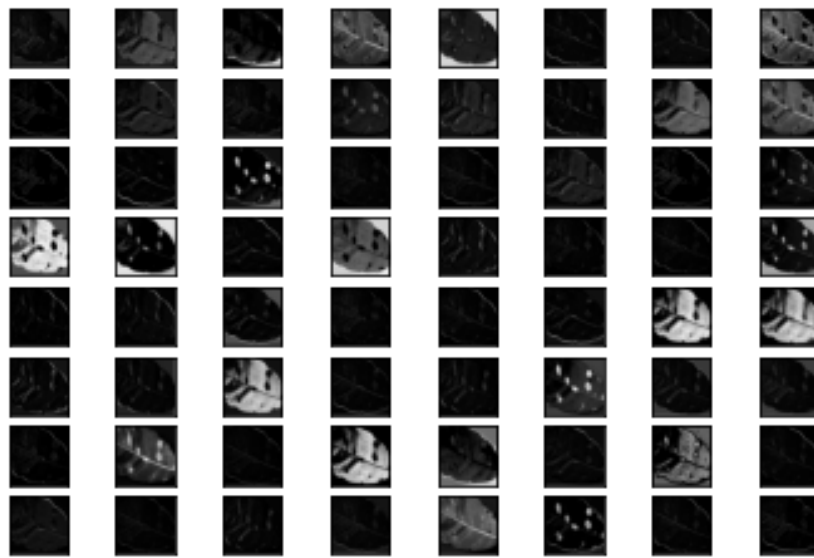


Fuente: Propia

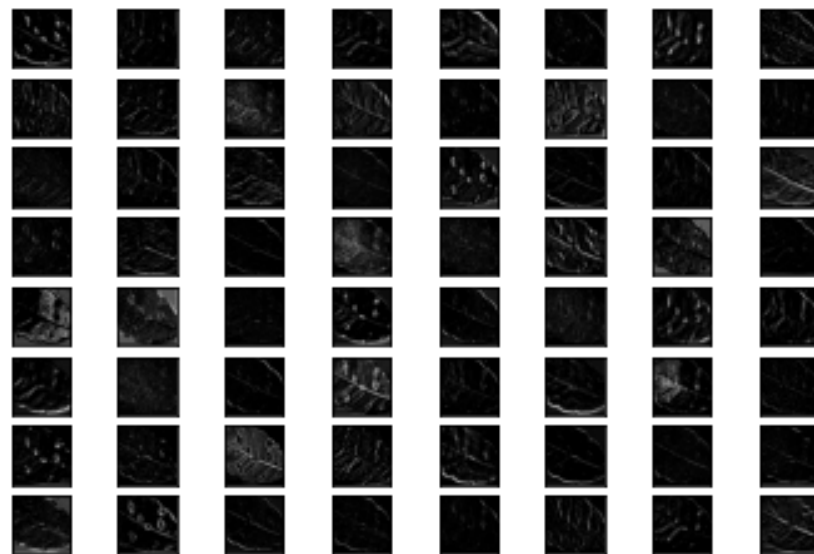
## Arquitectura VGG-16

Para la utilización de la arquitectura VGG-16 propuesto en el **capítulo 2.4.3** ya descrito las características, se procede a mostrar la implementación utilizada para el entrenamiento. La descripción por cada capa se encuentra en la **figura 1.3**, con la única diferencia que debe agregarse una capa densa de 3 neuronas que corresponde a las categorías de las enfermedades con una función de activación softmax. En cada una de las capas convoluciones se aplican filtros aleatorios asociados a cada neurona con el objetivo de extraer características como se muestran en las figura 3.17; todo ello contribuye al aprendizaje del características del modelo que es almacenado como un archivo con extensión .h5.

Figura 3.16: Visualización de los mapas de características VGG-16



(a) Primera capa convolucional

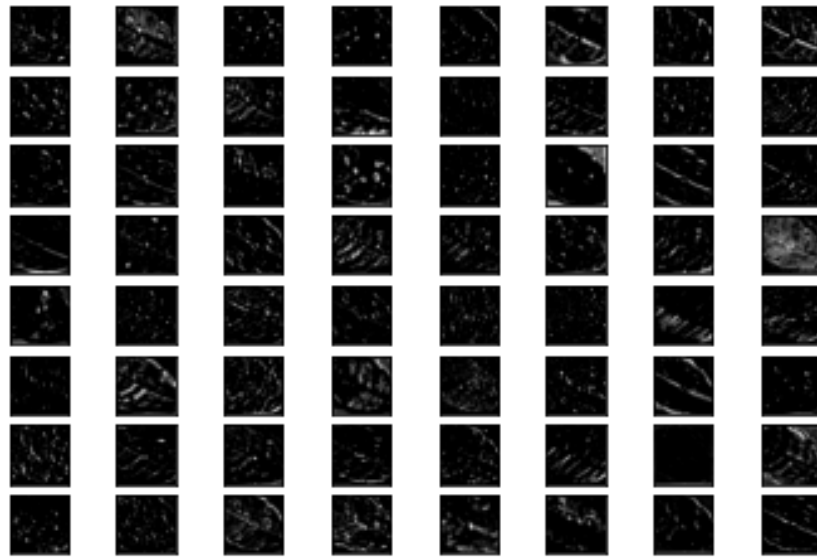


(b) Segunda capa convolucional

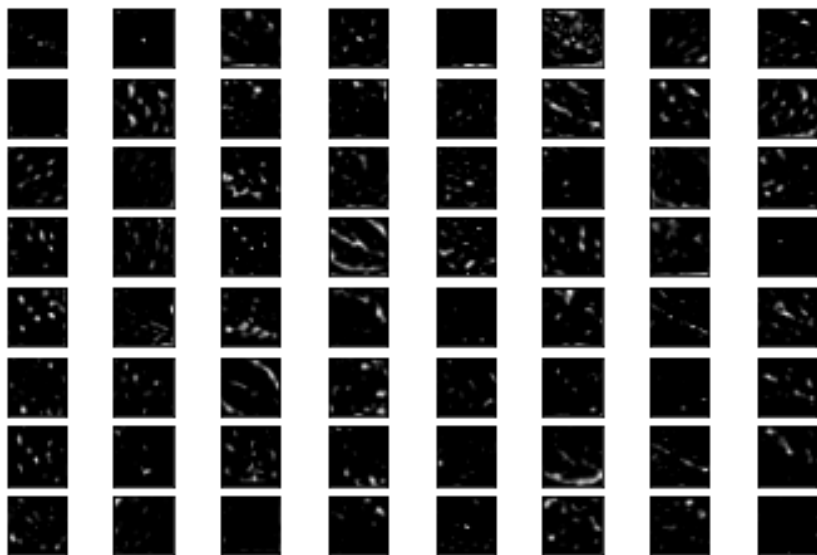
Fuente: Propia



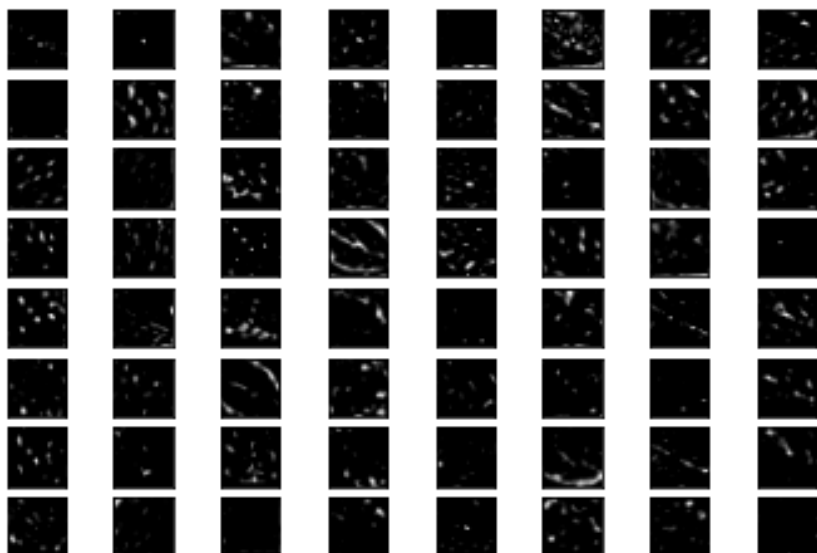
Figura 3.17: Visualización de los mapas de características VGG-16



(a) Tercera capa convolucional



(b) Cuarta capa convolucional



(c) Quinta capa convolucional

## Arquitectura AlexNet

La arquitectura AlexNet tiene cinco capas convolucionales, algunas de las cuales son seguidas por capas de max pooling, tres capas completamente conectadas y finalmente un clasificador softmax de 1000 nodos que es adaptado a tres para fines demostrativos del proyecto.

- La primera capa convolucional consta de 96 filtros, la imagen de entrada es redimensionada a 224x224 y 3 canales que corresponde a RGB, con un stride de 4 más una función de activación ReLU. Seguidamente una capa de pooling de tamaño 2 y stride de 2.
- La segunda capa convolucional consta con 256 filtros, tamaño de kernel 11 y stride 1. Seguidamente una capa pooling de tamaño 2 y stride 2.
- La tercera capa consta de 384 filtros con tamaño 3 y stride de 1 más una función de activación Relu.
- La cuarta capa consta de 384 filtros de tamaño 3 y stride de 1 más una función de activación Relu.
- La quinta capa consta de 256 filtros de tamaño 3 y stride de 1 con una función de activación Relu y a continuación una capa de maxpooling de tamaño 2 con stride de 2.
- Finalmente se tiene 3 capas completamente conectada de 4096, 4096 y 1000 neuronas respectivamente con Dropout de 0.4 para prevenir **overfitting**<sup>4</sup>.

## InceptionV3

Esta arquitectura de red neuronal convolucional descrita en el **capítulo 2.4.1**, cuya implementación se detalla a continuación. Se realiza la lectura de los pesos de modelo InceptionV3 preentrenado en la variable `local_weights_file`, posterior a ello se invoca a la función InceptionV3 donde cargamos la imagen de entrada a ser procesada definiendo las dimensiones a 150x150 según a parámetros establecidos por el modelo para su posterior entrenamiento, también se definen las capas densas de 1024 neuronas con activación Relu, capa de Dropout con un parámetro de 20 % y una capa densa con función de activación softmax constituido por 3 neuronas correspondiente a las categorías de las enfermedades.

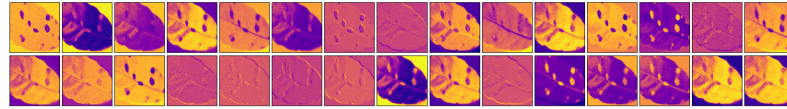
El modelo InceptionV3 extrae características a medida van ingresando las imágenes en las primeras capas como se muestra en la figura 3.18, tiene la peculiaridad de detectan bordes, algunos filtros detectan el fondo, mientras que otros están detectando solo el límite exterior de la hoja y así sucesivamente. En la tercera capa convolucional aumenta la abstracción, los filtros ahora pueden reconocer los bordes de la hoja y algunos filtros también activan la textura de la superficie de la imagen. En la cuarta capa de convolución las imágenes se han vuelto un poco borrosas debido a la operación MaxPooling realizada justo antes de esta capa; como se ingresan más capas de agrupación, el conocimiento que llega a la capa de

---

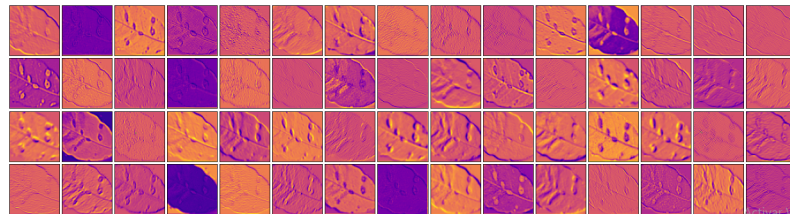
<sup>4</sup>Overfitting se refiere a sobreentrenar un algoritmo de aprendizaje con unos ciertos datos para los que se conoce el resultado deseado.

convolución se vuelve cada vez más abstracto, lo que ayuda a la red completa a clasificar la imagen correctamente, pero visualmente no nos proporcionan mucha información.

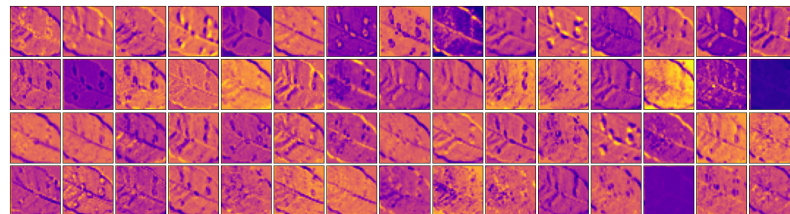
Figura 3.18: Visualización de los mapas de características InceptionV3



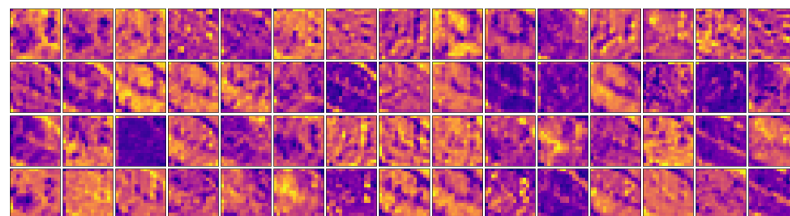
(a) Primera capa convolucional



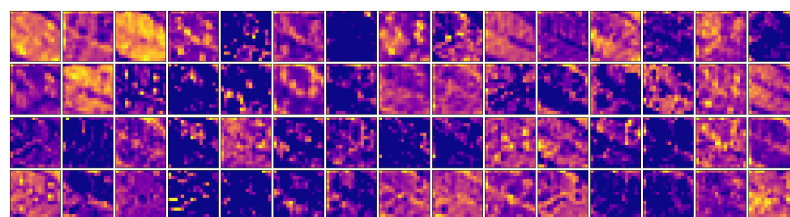
(b) Segunda capa convolucional



(c) Tercera capa convolucional



(d) Cuarta capa convolucional



(e) Quinta capa convolucional

**Fuente:Propia**

### 3.1.3. Información técnica

Para proceder con la determinación de la información técnica correspondiente a una enfermedad, es necesario obtener a que enfermedad corresponde la clasificación, por ello se tiene que esperar el resultado del proceso de clasificación para así generar la información técnica en base a las características de las enfermedades que básicamente consisten en mostrar la información. La información que nos muestra es lo siguiente; biología, agente causal de la enfermedad, daño causado por la enfermedad y el control de la enfermedad todo ello detallado en el capítulo 2.1.4.

Figura 3.19: Información técnica correspondiente a la enfermedad clasificada

Prototipo de un sistema de clasificación de enfermedades en las hojas del café basado

**INFORMACIÓN TÉCNICA**

**Biología**  
Manchas redondeadas amarillas naranjas, en el impacto se desprende un polvo, aumentan gradualmente.

**Agente**  
Hongo Hemileia Vastatrix

**Daño**  
Defoliación y muerte descendente de acuerdo a la infestación, caída de hojas.

**Control**  
Aplicaciones de oxicloruro de cobre después de la floración, aplicar azufre si se presenta sequía, fertilización balanceada con calcio, aplicar Bonomil, Folicur y Strobby

Nombre Científico  
Hemileia Vastatrix

Nombre Cómico  
Roya de Café

Fuente: Propia

### 3.1.4. Segmentación de la imagen y determinación del grado de severidad caso Hemileia Vastatrix

El proceso de segmentación de la imagen con respecto a la región interés, consiste en la extracción de las regiones sanas y enfermas en la hoja del cafeto. El proceso de extracción se basa en técnicas de umbralización, este método pretende segmentar las regiones diferentes a los umbrales pertenecientes a la tonalidad de la familia verde, que vendría a ser la segmentación de la región enferma y para definir la región sana se realiza el proceso inverso, de esta manera se logra la segmentación, se debe considerar que la imagen a segmentar debe tener el fondo con tonalidad blanca, debido a que el presente trabajo se enfoca más a la parte de la clasificación de la enfermedad.

Definimos la imagen de entrada como  $f(x, y)$  y la imagen de salida  $g(x, y)$  para todo el recorrido de la imagen, así mismo establecemos los umbrales  $T1$  y  $T2$  que esta representado por un rango de tonalidad verde expresado en formato RGB  $T1[36, 25, 25] \leq f(x, y) \leq T2[86, 255, 255]$  línea 2 del algoritmo 2, posterior a ello definimos las ecuaciones de umbralización para nuestro caso basado en el capítulo 2.2.2, prácticamente se le asigna tonalidad blanca a las regiones de la hoja que no tiene la enfermedad quedando solo la region enferma (se extrae regiones que estan fuera del rango de la tonalidad verde) línea 5, 6 y 7 del algoritmo 2.

$$g(x, y) = 255, si f(x, y) < T1 \quad , g(x, y) = 255, si f(x, y) \geq T2 \quad (3.2)$$

---

**Algorithm 2** Segmentación región enferma

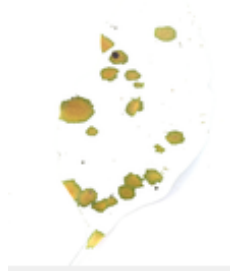
---

**Require:** *Imagen* : Hoja original

**Ensure:** Imagen: Región enferma

- 1: convert to hsv
  - 2: hsv = cv2.cvtColor(img, cv2.COLOR\_BGR2HSV)
  - 3: mask = cv2.inRange(hsv, (36, 25, 25), (86, 255, 255))
  - 4: slice the green
  - 5: imask = mask menor a 255 inicializando mask para la nueva imagen
  - 6: enfermedad = 255\*np.ones.like(img)
  - 7: enfermedad[imask] = img[imask]
  - 8: save
  - 9: cv2.imwrite(enfermedad\_segmentado.png, enfermedad)
- 

Figura 3.20: Región enferma



Fuente: Propia

para obtener la imagen de la región sana se realiza la operación inversa, conservando la máscara obtenida mediante en área del rango de tonalidad verde que es aplicada a una imagen generada con valores de pixel 255.

---

**Algorithm 3** Segmentación región sana

---

**Require:** *Imagen* : Hoja original

**Ensure:** Imagen: Región enferma

- 1: convert to hsv
  - 2: `hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)`
  - 3: `mask = cv2.inRange(hsv, (36, 25, 25), (86, 255,255))`
  - 4: slice the green
  - 5: `imask = mask mayor a 0`
  - 6: `enfermedad = 255*np.ones_like(img)`
  - 7: `enfermedad[imask] = img[imask]`
  - 8: save
  - 9: `cv2.imwrite("sana_segmentado.png", enfermedad)`
- 

**Fuente: Propia**

Figura 3.21: Región sana



**Fuente: Propia**

Para efectuar el proceso del cálculo del grado de severidad en la hoja se necesita del parámetro que corresponde a la clasificación de la enfermedad, en caso la enfermedad detectada sea *Hemileia Vastatrix* se procede con el proceso del cálculo del grado de severidad de de la hoja.

Se necesita la imagen segmentada de la región sana y región enferma, si complementamos ambas imágenes se obtiene la totalidad de pixeles de la hoja que representa el 100 % de la región de la hoja. También se necesita el total de pixeles de la región sana y enferma que son valores representativos del área correspondiente a cada región. Para obtener el grado de severidad de la hoja se aplica la siguiente fórmula en base a la tabla 2.2 mediante el algoritmo 5 y para el conteo de pixeles de la región sana y enferma algoritmo 4.

$$\text{Grado\_severidad(hoja)} = \frac{N_{\text{pixeles\_Enfermo}}}{N_{\text{pixeles\_Enfermo}} + N_{\text{pixeles\_Sano}}} * 100 \quad (3.3)$$

---

**Algorithm 4** Cálculo de área de región segmentada a través del conteo de pixeles

---

**Require:** *img*: Imagen segmentada ya sea sana o enferma

**Ensure:** cantidad-pixeles-región: Cantidad total de pixeles de la región

```

1: rows = len(img)
2: cols = len(img[0])
3: cantidad - pixeles - region = 0
4: for k = 0; k < cols; k ++; do
5:   for i = 0; i < rows; i ++; do
6:     for j = 0; j < cols; j ++; do
7:       if (img[i, j, k] != 255) : then
8:         cantidad_pixeles_region = cantidad_pixeles_region + 1
9:       end if
10:    end for
11:  end for
12: end for
13: return cantidad - pixeles - region

```

---

**Fuente: Propia**

---

**Algorithm 5** Grado de severidad en la hoja, caso *Hemileia Vastatrix*

---

**Require:** *img*: Imagen segmentada, sana y enferma

**Ensure:** Grado de severidad: Número real

```

1: pixeles_sanos = contar_pixeles_no_blanco(imagen_sana)
2: pixeles_enfermos = contar_pixeles_no_blanco(imagen_enferma)
3: total = pixeles_enfermos + pixeles_sanos
4: severidad = pixeles_enfermos/total

```

---

**Fuente: Propia**

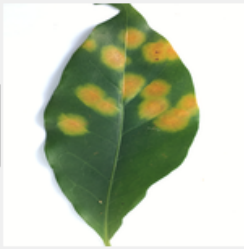
Figura 3.22: Resultado del prototipo

Prototipo de un sistema de clasificación de enfermedades en las hojas del café basado en visión computacional


**carga imagen a procesar**

CNN propio  
 CNN VGG-16  
 CNN AlexNet  
 Inception V3


**Limpiar**



**Región sana y enfermeda**



Región sana



Región enferma

**INFORMACIÓN TÉCNICA**

**Biología**

Manchas redondeadas amarillas naranjas, en el impacto se desprende un polvo, aumentan gradualmente.

**Agente**

Hongo *Hemileia Vastatrix*

**Daño**

Defoliación y muerte descendente de acuerdo a la infestación, caída de hojas.

**Control**

Aplicaciones de oxicloruro de cobre después de la floración, aplicar azufre si se presenta sequía, fertilización balanceada con calcio, aplicar Benomil, Follicur y Strobil

**Nombre Científico**

*Hemileia Vastatrix*

**Nombre Común**

Roya de Café

**Grado de Severidad**

Grado Severidad Roya  
{GRADO 3:} 0.4913

Fuente: Propia



# Capítulo 4

## Análisis de resultados y discusiones

### 4.1. Análisis de resultados

Dado el problema de diagnóstico de enfermedades en las hojas de cafeto de acuerdo a la clasificación eficiente de las mismas, para lo cual se desempeñó una serie de técnicas de clasificación basados en redes neuronales convolucionales. Así mismo se consideró el método de segmentación basado en el color de la imagen para proporcionar un valor aproximado del grado de severidad en la hoja, en caso el resultado de la clasificación fuese *Hemileia Vastatrix* o comúnmente denominado *Roya de Café*.

#### 4.1.1. Comparativa de arquitecturas de redes neuronales convolucionales en clasificación de enfermedades en las hojas de cafeto

Para la obtención del resultado, que viene a ser una enfermedad clasificada en su correspondiente categoría, se ha diseñado un modelo de red neuronal convolucional que se usa para el proceso de identificación de la enfermedad. Así mismo se realiza la comparativa con otros modelos de red neuronal convolucional ya construidos denominados AlexNet, InceptionV3 y VGG-16.

Se trabaja con un total de 1000 imágenes por cada enfermedad, con la única diferencia de la enfermedad *Leucoptera Coffeella* que posee poca recurrencia en los cultivos, por lo que las imágenes recolectadas llegan a 200, cuya distribución esta constituido por el 76.6 % datos entrenamiento y 23.3 % datos de validación<sup>1</sup>. El conjunto de imágenes construido es usado para el entrenamiento de todos los modelos de redes neuronales convolucionales propuestos.

Para el entrenamiento del modelo definido con los parámetros propios trabaja con imágenes de entrada redimensionados a 150X150, para el modelo InceptionV3 tambien se trabaja con imágenes de entrada redimensionados a 150X150, para los

---

<sup>1</sup>Se considera estos parámetros basado en el trabajo de investigación que consiste en la arquitectura de red neuronal convolucional LeNet para la identificación de edificios históricos obteniendo una precisión equivalente a 95 % (David Escobedo, 2017), Pagina 8.

Tabla 4.1: Resultados de evaluar arquitecturas de redes neuronales convolucionales

Exactitud/Iteración	1	2	3	4	5
CNN	0.916	0.926	0.844	0.913	0.906
AlexNet	0.437	0.451	0.442	0.482	0.571
VGG-16	0.4619	0.4628	0.4634	0.4614	0.4830
InceptionV3	0.7175	0.7675	0.7866	0.8040	0.8085

Fuente: Propia

modelos de VGG16 y AlexNet se trabajan con imágenes de entrada redimensionadas a 224X224.

Se considera 5 iteraciones con 1000 ejecuciones para el entrenamiento de cada modelo propuesto, cuya métrica es la exactitud o accuracy y para la evaluación de los modelos se usa la pérdida de la entropía cruzada categórica y optimizadores de coste Adam. Los resultados obtenidos por los 4 modelos de clasificación propuestos se muestra en la tabla 4.1.

Los resultados finales de exactitud por cada modelo se tiene, en el caso del modelo de red neuronal convolucional diseñado es de 90.1 %, en caso del modelo AlexNet es el 47.7 %, en caso de la arquitectura VGG-16 se obtiene una exactitud de 46.65 % y en caso de la arquitectura InceptionV3 se obtiene una exactitud de 77.68 %<sup>2</sup>.

Figura 4.1: Entrenamiento del modelo CNN diseñado

```

2071 images belonging to 3 classes.
757 images belonging to 3 classes.
VG:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow_core/python/ops/resource_variable_ops.py:1630: ca
ctions for updating:
ing Keras pass *_constraint arguments to layers.
1/5
1000 [=====>.] - ETA: 7s - loss: 0.2533 - acc: 0.8999 Epoch 1/5
1000 [=====] - 9441s 9s/step - loss: 0.2533 - acc: 0.9000 - val_loss: 0.2131 - val_acc: 0.916
2/5
1000 [=====>.] - ETA: 5s - loss: 0.0792 - acc: 0.9713 Epoch 1/5
1000 [=====] - 7108s 7s/step - loss: 0.0792 - acc: 0.9713 - val_loss: 0.2479 - val_acc: 0.926
3/5
1000 [=====>.] - ETA: 4s - loss: 0.0350 - acc: 0.9883 Epoch 1/5
1000 [=====] - 6331s 6s/step - loss: 0.0350 - acc: 0.9883 - val_loss: 0.5988 - val_acc: 0.844
4/5
1000 [=====>.] - ETA: 4s - loss: 0.0311 - acc: 0.9896 Epoch 1/5
1000 [=====] - 6138s 6s/step - loss: 0.0311 - acc: 0.9896 - val_loss: 0.4020 - val_acc: 0.913
5/5
1000 [=====>.] - ETA: 4s - loss: 0.0234 - acc: 0.9920 Epoch 1/5
1000 [=====] - 6006s 6s/step - loss: 0.0234 - acc: 0.9920 - val_loss: 0.3699 - val_acc: 0.906
ileia Vastatrix': 0, 'Leucoptera Coffella': 1, 'Mycena Citricolor': 2}

```

Fuente: Propia

Figura 4.2: Entrenamiento del modelo AlexNet

```

1000/1000 [=====] - 8039s 8s/step - loss: 0.9810 - acc: 0.4374 - val_loss: 0.9539 - val_acc: 0.5444
Epoch 2/20
53/1000 [>.....] - ETA: 1:29:18 - loss: 0.9744 - acc: 0.4513

```

Fuente: Propia

<sup>2</sup>Los modelos AlexNet, InceptionV3 y VGG-16 tienen arquitecturas complejas y por ello tiene ciertas dificultades con poca cantidad de información para el entrenamiento, como podemos ver incrementan su exactitud lo que puede mejorar los resultados a más entrenamiento

Figura 4.3: Entrenamiento del modelo VGG-16

```
ound 2071 images belonging to 3 classes.
ound 757 images belonging to 3 classes.
poch 1/5
000/1000 [=====] - 7259s 7s/step - loss: 8.3323 - acc: 0.4821 - val_loss: 8.6736 - val_acc: 0.4619

poch 00001: val_acc improved from -inf to 0.46187, saving model to /content/drive/My Drive/TESIS/CNN/Modelos/vgg16_2.h5
poch 2/5
000/1000 [=====] - 7296s 7s/step - loss: 8.3438 - acc: 0.4823 - val_loss: 8.6581 - val_acc: 0.4628

poch 00002: val_acc improved from 0.46187 to 0.46283, saving model to /content/drive/My Drive/TESIS/CNN/Modelos/vgg16_2.h5
poch 3/5
000/1000 [=====] - 7137s 7s/step - loss: 8.3389 - acc: 0.4826 - val_loss: 8.6498 - val_acc: 0.4634

poch 00003: val_acc improved from 0.46283 to 0.46335, saving model to /content/drive/My Drive/TESIS/CNN/Modelos/vgg16_2.h5
poch 4/5
000/1000 [=====] - 7198s 7s/step - loss: 8.3208 - acc: 0.4838 - val_loss: 8.6820 - val_acc: 0.4614

poch 00004: val_acc did not improve from 0.46335
poch 5/5
15/1000 [.....] - ETA: 43:09 - loss: 7.9247 - acc: 0.5083
```

Fuente:Propia

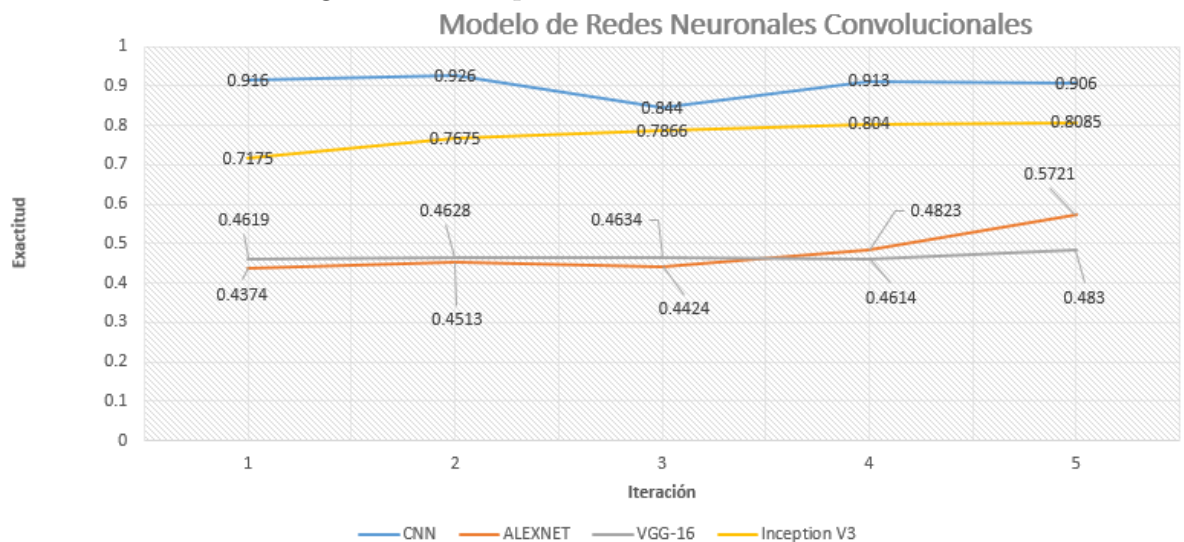
Figura 4.4: Entrenamiento del modelo Inception V3

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow_core/python/ops/resource_variable_ops.py:1630: calli
Instructions for updating:
If using Keras pass *_constraint arguments to layers.
last layer output shape: (None, 7, 7, 768)
Found 2071 images belonging to 3 classes.
Found 757 images belonging to 3 classes.
Epoch 1/5
999/1000 [=====>.] - ETA: 4s - loss: 0.7465 - acc: 0.6439Epoch 1/5
1000/1000 [=====] - 6126s 6s/step - loss: 0.7464 - acc: 0.6439 - val_loss: 0.7243 - val_acc: 0.7175
Epoch 2/5
999/1000 [=====>.] - ETA: 3s - loss: 0.6137 - acc: 0.7649Epoch 1/5
1000/1000 [=====] - 4654s 5s/step - loss: 0.6137 - acc: 0.7649 - val_loss: 0.6218 - val_acc: 0.7675
Epoch 3/5
999/1000 [=====>.] - ETA: 3s - loss: 0.5401 - acc: 0.8023Epoch 1/5
1000/1000 [=====] - 4573s 5s/step - loss: 0.5402 - acc: 0.8022 - val_loss: 0.5597 - val_acc: 0.7866
Epoch 4/5
999/1000 [=====>.] - ETA: 3s - loss: 0.4880 - acc: 0.8193Epoch 1/5
1000/1000 [=====] - 4485s 4s/step - loss: 0.4882 - acc: 0.8191 - val_loss: 0.5104 - val_acc: 0.8040
Epoch 5/5
999/1000 [=====>.] - ETA: 3s - loss: 0.4440 - acc: 0.8333Epoch 1/5
1000/1000 [=====] - 4664s 5s/step - loss: 0.4438 - acc: 0.8333 - val_loss: 0.4792 - val_acc: 0.8085
```

Fuente:Propia

En las figuras 4.1, 4.2, 4.3 y 4.4 se aprecia el proceso de entrenamiento de la red neuronal convolucional, que consta de 5 iteraciones con 1000 ejecuciones cada una y al finalizar la iteración se puede ver el promedio mostrado en la variable val\_acc que representa la métrica de exactitud que corresponde a cada iteración; donde se hace uso de la pérdida de entropía cruzada categórica y optimizadores de ADAM.

Figura 4.5: Comparación de resultados



**Fuente: Propia**

En la figura 4.5 se aprecia el comportamiento que tiene cada modelo durante el entrenamiento, la red neuronal convolucional diseñado se mantiene constante salvo una caída en la tercera iteración y es la más adecuada por el momento; en cambio el modelo de red neuronal que tiene un progreso constante durante el entrenamiento es el modelo InceptionV3 que en más iteraciones puede lograr obtener los mejores resultados, los modelos que no cumplen con las expectativas y no surgen mejoras durante el entrenamiento son los modelos AlexNet y VGG-16.

#### 4.1.2. Resultados de segmentación, caso Hemileia Vastatrix

En caso de ser el resultado de la clasificación Hemileia Vastatrix, se procede con la segmentación de la hoja enferma a procesar para la obtención de la región enferma y región sana, para el cálculo de la región o área se realiza a través del conteo de píxeles, la dimensión de las imágenes son importantes aunque no influye en los resultados obtenidos, pero si en el tiempo de ejecución al momento de realizar el conteo de píxeles de la región.

- La región enferma esta representado por número de píxeles del área afectada por la enfermedad.
- La región sana esta representado por número de píxeles del área sana de la hoja.

- El grado de severidad en la hoja es obtenida del cociente de la región enferma entre el total del área de la hoja, que representa el porcentaje del grado de severidad en la hoja, y es categorizado de acuerdo a la tabla del grado de severidad, **tabla 2.2**.

Los resultados obtenidos de acuerdo a la redimensión de las imágenes segmentadas influyen en el tiempo de ejecución para el cálculo del grado de severidad en la hoja, se trabaja con la imagen de la **Figura 3.1.4**.

Dimensiones	Pixeles región sana	Pixeles región enferma	Ejecución
100x100	12126 - 9397	9397	1.2 seg
1000x1000	1123848 - 748986	748986	12.9 seg

**Fuente:Propia**

Figura 4.6: Resultados, cálculo del grado de severidad en la hoja

```
100 100
-----
100 100
-----
12126 9397
('PERTENECE AL GRADO 3:', 0.43660270408400315)
[Finished in 1.2s]
```

(a) Resultados con dimensión 100x100

```
1000 1000
-----
1000 1000
-----+
1123848 748986
('PERTENECE AL GRADO 3:', 0.3999211889574837)
[Finished in 12.9s]
```

(b) Resultados con dimensión 1000x1000

**Fuente:Propia**

El resultado obtenido del grado de severidad en la hoja en los casos mencionados es el mismo, sin importar las dimensiones, en ambos casos pertenecen al grado 3, debido a que se procesan las mismas imágenes segmentadas correspondiente a la misma imagen de entrada que fue clasificado como Hemileia Vastatrix.

## 4.2. Discusión de resultados

- Cálculo de severidad a nivel de hoja realizando una observación visual objetiva haciendo uso de escalas y diagramas de evaluación; posteriormente se procedió al registro los datos en el formato de evaluación para su posterior análisis obteniendo una severidad en las variedades típica con 55.18 %, caturra 24.64 % de incidencia y por último la variedad catimor presento una prevalencia de 4.2 %; cabe resaltar que estos resultados son muy importantes para un diagnóstico óptimo, por ende un proceso de análisis de evaluación manual no es recomendable, nuestro prototipo de sistema puede realizar las tareas mencionadas y obtener los resultados del grado de severidad en la

hoja de manera automática optimizando los tiempos de respuesta para el diagnóstico de incidencia y severidad para la Roya Amarilla ( *Hemileia Vastatrix*) en cultivares de café del Sector de Mandor, Distrito de Maranura, Provincia La Convención, Departamento del Cusco.

- El trabajo de investigación que consiste en evaluar técnicas de aprendizaje máquina para la identificación de imágenes de edificios históricos de la ciudad del Cusco basado en Bags Of Words y Redes Neuronales Convolucionales es utilizado con fines aplicativos en problemas de clasificación de imágenes y evaluación del rendimiento, al igual que el ítem anterior se tienen problemáticas distintas.
- El diagnóstico automático de la Roya de Café Amarilla aplicando técnicas de procesamiento de imágenes y aprendizaje máquina cuyo objetivo es básicamente reconocer si la imagen de entrada tiene o no la enfermedad de Roya de Café y realizar el proceso de diagnóstico en base a la información recopilada del grado de severidad en la hoja, evaluando el clasificador basado en Support Vector Machine planteando 10 modelos obtienen una exactitud en la clasificación superior al 80 % (conclusión relacionado al objetivo específico 3), así mismo sus dos mejores modelos obtienen exactitud de (87.50 % y 95.53 %) que puede ser un resultado engañoso debido a que solo toma variaciones de intensidad de los píxeles en cada canal de color (conclusión relacionado al objetivo específico 4). Nuestro clasificador de imágenes abarca todas las enfermedades que dañan las hojas con una tasa de exactitud del 90.1 % y determinando el grado de severidad en la hoja que es información importante para un diagnóstico óptimo, por ende nuestro trabajo de investigación es aplicable y escalable debido a que en el prototipo podríamos mostrar más información que contribuyan al diagnóstico de todas las enfermedades en las hojas.
- El clasificador de hojas mediante Deep Learning tiene la deficiencia de realizar clasificación binaria (clase de la hoja, uva o tomate) de acuerdo a ello recién se determina la enfermedad (clasificación categórica) y falta de información de control de sanidad en base a la enfermedad para la herramienta de simulación; sin embargo obtiene mejores resultados de exactitud (98 %) a consecuencia de usar gran cantidad de imágenes de hojas enfermas y capas de extracción de características del modelo AlexNet que se adaptó rigurosamente al problema en particular.

### 4.3. Detalles técnicos

Las condiciones básicas para desarrollar adecuadamente el siguiente trabajo de investigación y cumplir con las tareas propuestas se detallan a continuación.

#### 4.3.1. Recursos hardware

- **Cámara de celular de 12mpx.-** Con la finalidad de realizar la captura de imágenes.

- **Notebook HP, 8gb de memoria Ram, procesador intel Core i5, tarjeta gráfica dedicada Radeon R2.-** Para la construcción del prototipo del sistema.

### 4.3.2. Recursos software

- **Sistema operativo.-** Windows 10.
- **Google Colaboratory.-** Como máquina virtual de Google para poder agilizar el entrenamiento de la red neuronal convolucional construida y los modelos propuestos.
- **Lenguaje de programación.-** Python version 3.7.0.
- **Librerías**
  - Tensor Flow 1.14.0rc1 .- Usado para la construcción del modelo de red neuronal convolucional a través de las funciones que nos ofrece el framework, también para calcular la exactitud mediante la entropía cruzada categórica usando optimizador de Adam.
  - Google.colab.- Usado para la carga del conjunto de datos al drive para que pueda ser entrenado en la máquina virtual del Google.
  - Numpy 1.16.4 .- Librería usada para ejecutar operaciones matemáticas entre matrices aplicado a filtros e imágenes.
  - Keras 2.2.0.- Para la lectura de los modelos pre-entrenados y ser usados en el prototipo del sistema.
  - TKinter.- Construcción de la interface básica.
  - PIL.- Usado para el tratamiento de las imágenes en el desarrollo del prototipo.
  - Opencv 4.1.1.26.- Usado para el tratamiento de las imágenes y algunas operaciones sobre ellas.
- **Link del proyecto:** <https://gitlab.com/softerp/pry-tesis>

# Conclusiones

1. El proceso de captura de imágenes se realizó en un entorno semicontrolado con fondo blanco, debido a las condiciones del entorno de la hoja, estas se encuentran superpuestas y se tiene una estructura de la planta muy frondosa, aspectos medioambientales como el viento ocasionan que no se haga una captura adecuada, a mayor cantidad de imágenes por categoría mejora la exactitud del modelo de clasificación.
2. Al definir los parámetros para el diseño de la red neuronal convolucional se concluye que a mayor cantidad de capas ocultas, mayor número de filtros y sin realizar el uso adecuado de la función de inactivación DROPOUT, incrementa la complejidad de la arquitectura causando ineficiencia durante el entrenamiento y overfitting durante la predicción.
3. Con respecto al objetivo específico 3 se tiene:
  - a) El modelo de aprendizaje automático de red neuronal convolucional diseñado para clasificar enfermedades en las hojas de cafeto obtiene como resultados de evaluación una exactitud de 90.1 % y los modelos propuestos tales como InceptionV3, AlexNet, VGG16 con exactitud de 77.68 %, 47.7 %, 46.65 % respectivamente; concluyendo que el modelo diseñado obtiene mejores resultados para la clasificación.
  - b) El proceso de entrenamiento de la red neuronal convolucional diseñado se realizó en un entorno virtual denominado Google Colaboratory usando la CPU de sus servidores, el inconveniente ocasionado fue al momento de entrenar debido al tiempo de entrenamiento y las máquinas virtuales se reinician cada cierto tiempo lo que generó que se cancele el entrenamiento en la mayoría de las veces.
4. Con respecto al objetivo específico 4 se tiene:
  - a) Para calcular el grado de severidad de la enfermedad Hemileia Vastatrix en la hoja se reajusto (disminuir) las dimensiones de las imágenes segmentadas para optimizar el proceso de cálculo de la región enferma y sana en su porcentaje correspondiente con respecto al área total de la hoja.
  - b) El cálculo del área a través del conteo de píxeles no es una técnica eficiente, puede existir el caso donde la hoja no tome una forma plana o semi plana y sufra transformaciones por el clima uno de los factores es el sol, ocasionado una curvatura en la hoja, esto genera que el área calculado no sea el aproximado al real.



# Recomendaciones

- Extender el trabajo de investigación haciendo uso de otros modelos de redes neuronales convolucionales pre-entrenados como ResNet e InceptionV4 definiendo un valor alto de número de pasos ( mayor frecuencia de entrenamiento), también considerar la calidad de imágenes y que mejor si es tomada con una cámara profesional y así realizar el preprocesamiento de imágenes durante la construcción de dataset más no al iniciar el entrenamiento lo que permitiría reducir la complejidad y el tiempo de ejecución.
- Se recomienda hacer un análisis del entorno del problema de clasificación y extender el trabajo evaluando un escenario real donde la clasificación es rechazada por el modelo de aprendizaje máquina, es decir imágenes que no pertenecen a ninguna clase de la enfermedad, se debe proponer modelos capaces de rechazar la imagen.
- Extender el trabajo de investigación haciendo uso de integrales dobles u otros métodos para el cálculo del área de la región sana y enferma para así obtener mejores resultados del grado de severidad, debido a que las hojas presentan ciertas curvaturas y no siempre tienen una estructura plana.
- Existe complicaciones de compatibilidad de librerías para trabajar con redes neuronales convolucionales durante la instalación en un entorno local de python y limitaciones de hardware, por ello se recomienda hacer uso de Google Colaboratory para el entrenamiento de la red neuronal ya que nos ofrece entornos de ejecución de GPU (Unidad de Procesamiento Gráfico) y TPU (Unidad de Procesamiento de Tensores) y a su vez ya tiene las librerías que facilitan la aplicación de inteligencia artificial a la resolución de problemas.
- Se trabajo en un entorno semi controlado, se recomienda hacer uso del formato RAW porque realiza la captura de imágenes ajustando brillo, contraste, reducción de ruido e incluso sharpening y extracción del fondo de la imagen facilitando el preprocesamiento de la imagen. Así mismo se recomienda convertir la data que tenemos en formato SRGV para disminuir el tamaño de las imágenes y optimizar los tiempos de ejecución durante el entrenamiento para obtener mayor exactitud del modelo de red neuronal convolucional.
- Como ya contamos con los modelos entrenados exclusivamente para clasificar enfermedades en las hojas de cafeto, estos podrían ser reutilizandos en cualquier otro proyecto, se recomienda realizar la migración de proyecto a un entorno web haciendo uso del Framework DJANGO de python el cual es flexibe al utilizar las librerías de keras para la lectura del modelo de red neuronal convolucional en el proyecto.

## Anexos

## **Apéndice A**

### **Informe del sistema antiplagio Turnitin**

Figura A.1: Reporte de similitud parte I



Tesis\_\_cafeto\_0\_1\_3.pdf  
Dec 24, 2020  
21562 palabras/114589 caracteres

Erwin Jharin Puella Zegarra

Tesis\_\_cafeto\_0\_1\_3.pdf

Resumen de fuentes

7%

SIMILITUD GENERAL

1	es.stackoverflow.com INTERNET	<1%
2	Universidad Autónoma de Bucaramanga,UNAB on 2020-11-13 TRABAJOS ENTREGADOS	<1%
3	BENEMERITA UNIVERSIDAD AUTONOMA DE PUEBLA BIBLIOTECA on 2019-10-15 TRABAJOS ENTREGADOS	<1%
4	Universidad Europea de Madrid on 2020-09-14 TRABAJOS ENTREGADOS	<1%
5	BENEMERITA UNIVERSIDAD AUTONOMA DE PUEBLA BIBLIOTECA on 2019-10-17 TRABAJOS ENTREGADOS	<1%
6	repositorio.unsaac.edu.pe INTERNET	<1%
7	es.slideshare.net INTERNET	<1%
8	sitiobigdata.com INTERNET	<1%
9	sedici.unlp.edu.ar INTERNET	<1%
10	www.python2.net INTERNET	<1%
11	docplayer.es INTERNET	<1%
12	es.wikipedia.org INTERNET	<1%
13	towardsdatascience.com INTERNET	<1%
14	biblioteca.usbbog.edu.co:8080 INTERNET	<1%
15	oa.upm.es INTERNET	<1%
16	University of Sheffield on 2020-09-14 TRABAJOS ENTREGADOS	<1%

Figura A.2: Reporte de similitud parte II

17	udemsemilleroandroid.blogspot.com INTERNET	<1%
18	www.aprendemachinelearning.com INTERNET	<1%
19	Universidad de Lima on 2020-07-17 TRABAJOS ENTREGADOS	<1%
20	Universidad Internacional de la Rioja on 2018-07-26 TRABAJOS ENTREGADOS	<1%
21	Universidad Carlos III de Madrid on 2020-07-06 TRABAJOS ENTREGADOS	<1%
22	Universidad de Guadalajara on 2020-08-28 TRABAJOS ENTREGADOS	<1%
23	Escuela Politecnica Nacional on 2020-06-18 TRABAJOS ENTREGADOS	<1%
24	Universidad de la Amazonia on 2019-08-21 TRABAJOS ENTREGADOS	<1%
25	Universidad Católica de Santa María on 2016-01-08 TRABAJOS ENTREGADOS	<1%
26	developers.google.com INTERNET	<1%
27	repository.unad.edu.co INTERNET	<1%
28	mc.ai INTERNET	<1%
29	Escuela Politecnica Nacional on 2017-08-01 TRABAJOS ENTREGADOS	<1%
30	Universitas Sebelas Maret on 2019-05-11 TRABAJOS ENTREGADOS	<1%
31	Wright State University on 2014-12-18 TRABAJOS ENTREGADOS	<1%
32	repositorio.unfv.edu.pe INTERNET	<1%
33	Universidad Nacional San Antonio Abad del Cusco on 2020-11-21 TRABAJOS ENTREGADOS	<1%
34	download.atlantis-press.com INTERNET	<1%
35	www.repositorio.usac.edu.gt INTERNET	<1%
36	Universidad Autónoma de Madrid on 2020-06-22 TRABAJOS ENTREGADOS	<1%
37	Universidad Politecnica Salesiana del Ecuador on 2020-08-05 TRABAJOS ENTREGADOS	<1%
38	Universidad de Lima on 2018-07-24 TRABAJOS ENTREGADOS	<1%
39	enfoquesusco.wordpress.com INTERNET	<1%
40	red.uao.edu.co INTERNET	<1%

Figura A.3: Reporte de similitud parte III

41	repositorio.unsa.edu.pe	INTERNET	<1%
42	ri.uaq.mx	INTERNET	<1%
43	www.microsoft.com	INTERNET	<1%
44	Daegu Gyeongbuk Institute of Science and Technology on 2020-05-12	TRABAJO ENTREGADO	<1%
45	Politécnico Colombiano Jaime Isaza Cadavid on 2020-07-23	TRABAJO ENTREGADO	<1%
46	Universidad Carlos III de Madrid on 2018-09-15	TRABAJO ENTREGADO	<1%
47	www.camisea.com.pe	INTERNET	<1%
48	BENEMERITA UNIVERSIDAD AUTONOMA DE PUEBLA BIBLIOTECA on 2019-03-26	TRABAJO ENTREGADO	<1%
49	UISEK on 2018-11-07	TRABAJO ENTREGADO	<1%
50	Universidad Carlos III de Madrid on 2020-07-04	TRABAJO ENTREGADO	<1%
51	Universidad de San Buenaventura on 2017-11-27	TRABAJO ENTREGADO	<1%
52	Universidad de Valladolid on 2020-09-21	TRABAJO ENTREGADO	<1%
53	Universitat Politècnica de València on 2019-09-10	TRABAJO ENTREGADO	<1%
54	elvex.ugr.es	INTERNET	<1%
55	manualzz.com	INTERNET	<1%
56	prensa.ugr.es	INTERNET	<1%
57	www.jordibafoto.com	INTERNET	<1%
58	www.senasa.gob.pe	INTERNET	<1%

Se excluyeron los depósitos de búsqueda:

- Ninguno

Excluido del Informe de Similitud:

- Bibliografía
- Citas
- Coincidencias menores (10 palabras o menos)

Se excluyeron las fuentes:

- Ninguno

## Apéndice B

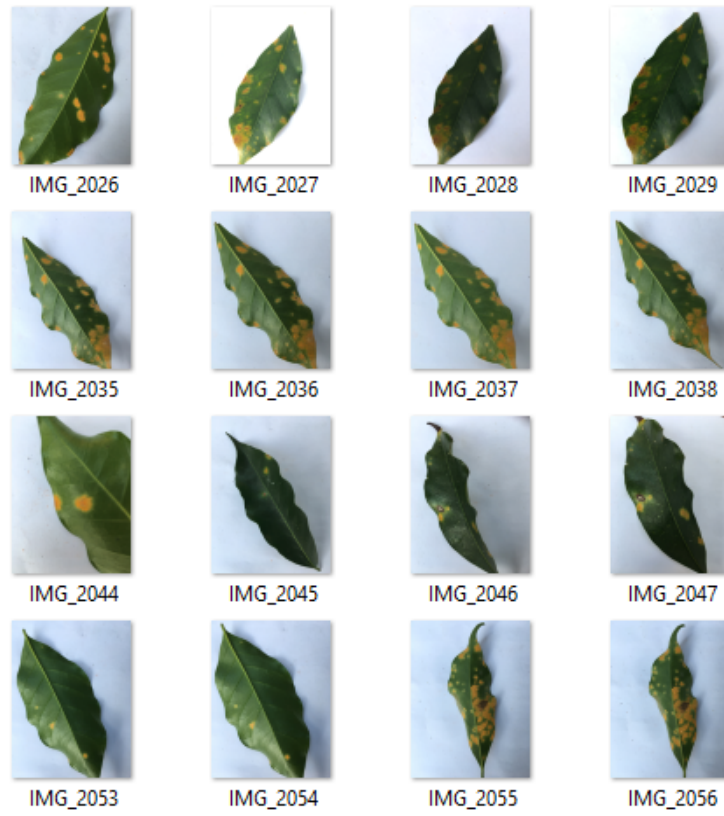
### Set de datos de entrenamiento y validación

Figura B.1: Dataset de imágenes de entrenamiento

I (D:) > tesis-cafe > dataset1 > entrenamiento

Nombre	Fecha de modifica...	Tipo	Tamaño
■ HemileiaVastatrix	8/10/2019 1:02 p. m.	Carpeta de archivos	
■ LeucopteraCoffella	8/10/2019 1:13 p. m.	Carpeta de archivos	
■ MycenaCitricolor	8/10/2019 1:10 p. m.	Carpeta de archivos	

(a) Estructura de datos



(b) Hemileia Vastatrix



Figura B.2: Dataset de entrenamiento Leucoptera Cofeella y Mycena Citricolor

> tesis-cafe > dataset1 > entrenamiento > LeucopteraCoffella



IMG\_20190603\_10  
0503



IMG\_20190603\_10  
0503\_1



IMG\_20190603\_10  
0504



IMG\_20190603\_10  
0504\_1



IMG\_20190603\_10  
0507\_1



IMG\_20190603\_10  
0507\_2



IMG\_20190603\_10  
0508



IMG\_20190603\_10  
0508\_1

(a) Leucoptera Cofeella

tesis-cafe > dataset1 > entrenamiento > MycenaCitricolor



IMG\_0965



IMG\_0966



IMG\_0967



IMG\_0968



IMG\_0974



IMG\_0975



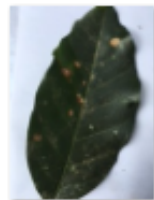
IMG\_0976



IMG\_0977



IMG\_0983



IMG\_0984



IMG\_0985






IMG\_0986

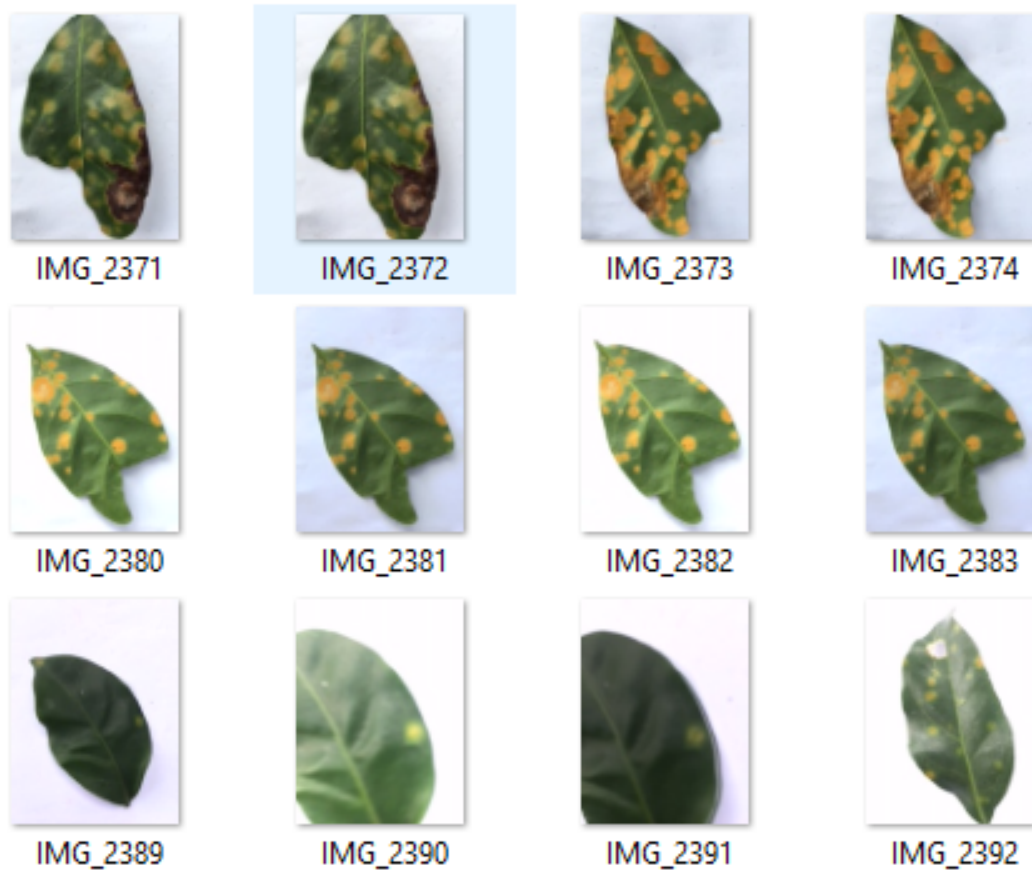
(b) Mycena Citricolor

Figura B.3: Dataset de validación Hemileia Vastatrix

(D:) > tesis-cafe > dataset1 > validacion

Nombre	Fecha de modifica...	Tipo	Tamaño
 HemileiaVastatrix	8/10/2019 12:58 p....	Carpeta de archivos	
 LeucopteraCoffella	8/10/2019 1:13 p. m.	Carpeta de archivos	
 MycenaCitricolor	8/10/2019 1:11 p. m.	Carpeta de archivos	

(a) Estructura de datos



(b) Hemileia Vastatrix

Figura B.4: Dataset de validación *Leucoptera Cofeella* y *Mycena Citricolor*



(a) *Leucoptera Cofeella*



(b) *Mycena Citricolor*

# Bibliografía

(2019). *Técnicas de Filtrado*.

Agarwal, R. (2018). La ruta del café peruano. <http://rutadelcafeperuano.com/2018/03/20/la-roya-y-el-cafe-en-el-peru-todo-lo-que-debes-saber/>.

Arrasco Ordoñez, Carlos Barriga Posada, A. C. (2018). Diagnóstico automático de roya amarilla en hojas de cafeto aplicando técnicas de procesamiento de imágenes y aprendizaje máquina.

Auer, P. H. B. W. M. (2008). Red neuronal prealimentada.

Barquero Miranda, M. (2013). *Recomendaciones para el combate de la roya del cafeto*. Centro de investigación en café(CICAFFE), Costa Rica.

Cáceda, J. J. V. (2018). *ANÁLISIS DE LA DIVERSIDAD GENÉTICA DE *Hemileia vastatrix* DE QUILLABAMBA MEDIANTE SECUENCIACIÓN DE LAS REGIONES ITS DEL ADN RIBOSOMAL*.

Cantero Lorenzo, J. (2017). *Máquinas de aprendizaje y aplicaciones*, pages 05–06.

Cantero Lorenzo, J. (2017-2018). *Máquinas de aprendizaje y aplicaciones*. España.

Chenyi Chen, Ari Seff, A. K. J. X. (2019). *DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving*.

Constantino Luis, M. Flores C., J. . P. B. (2011). *Minador de las hojas de café*. Centro Nacional de Investigaciones de Café(CICAFFE), Costa Rica.

Crispi, J. (2019). Redes neuronales convolucionales.

CUBIDES, M. L. L. (2019). *Incidencias Del Minador *Leucoptera coffeella* En El Cultivo De Café Coffea, En Dos Lotes Con Condiciones Diferentes*.

Das, S. (2017). Cnn architectures: Lenet, alexnet, vgg, googlenet, resnet. <https://medium.com/analytics-vidhya/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df>.

David Escobedo, J. F. (2017). Evaluación de técnicas de aprendizaje máquina para la identificación de imágenes de edificios históricos de la ciudad del cusco basado en bags of words y redes neuronales convolucionales.

de Sevilla, U. (2017-2018). Procesamiento de imágenes digitales. page 8.

- Enrique Sucar, L. . G. G. (2018a). *Visión computacional*. Puebla, Mexico Neuherberg, Alemania.
- Enrique Sucar, L. . G. G. (2018b). *Visión computacional*. pages 8–10. Puebla, Mexico Neuherberg, Alemania.
- Enrique Sucar, L. . G. G. (2018c). *Visión computacional*. pages 15–17. Puebla, Mexico Neuherberg, Alemania.
- Ernandez Sampieri, R. (2014). *Metodología de la investigación*. MC Graw Hill Education, Prolongación Paseo de la Reforma 1015, Torre A, México.
- Fuentes, L. M. (2015). *Tratamiento de imágenes y sonido*.
- Galarraga, J. L. (2017). Clasificador de hojas mediante deep learning. Master's thesis.
- Gato Utrera, R. (2017). Uso algoritmos de aprendizaje automático aplicado a la base de datos genéticos. pages 23–26.
- Gestal Pose, M. (2019). *Introducción a las redes neuronales artificiales*, page 05.
- Gómez-Ríos, A., Tabik, S., Luengo, J., Herrera, F., and Shihavuddin, A. (2019). Redes neuronales convolucionales para una clasificación precisa de imágenes de corales. page 1172.
- Gonzales, R.C. Wintz, P. (1995). Filtrado de imágenes. pages 30–38.
- Jhosi, P. (2015). *OpenCV With Python By Example*.
- Juan, I. B. (2018). Aprende machine learning. <https://www.aprendemachinelearning.com/como-funcionan-las-convolutional-neural-networks-vision-por-ordenador/>.
- La Serna Palomino, N. . R. C. U. (2009a). Técnicas de segmentación en procesamiento digital de imágenes. page 11.
- La Serna Palomino, Nora Concha Roman, U. (2009b). Técnicas de segmentación en el procesamiento digital de imágenes digitales. pages 10–11.
- La Serna Palomino, Nora Concha Roman, U. (2009c). Técnicas de segmentación en el procesamiento digital de imágenes digitales. pages 14–15.
- Ligdi, G. (2018). Diferencia entre algoritmos de clasificación y regresión. <http://ligdigonzalez.com/diferencia-entre-algoritmos-de-clasificacion-y-regresion/>.
- Lizarrá Valencia, L. (2012). Manejo integrado de plagas en el cultivo de café. pages 18–19.
- Maklin, C. (2019). Dropout neural network layer in keras explained.
- Mantupa Wilbert, C. (2016). Diagnóstico de incidencia y severidad de roya amarilla en cultivares de café en el sector de mandor distrito de maranúra provincia la convención.

- Marr, D. (1982). *A computational Investigation into the human representation and processing of visual information*. Library of congress cataloging in publication data, 41 Madison Avenue, New York.
- Nini Maria, N. . D. A. M. (2016). *El café y sus diversidades aplicaciones en la gastronomía*. Grupo de visión del comité español de automática, España.
- OTECH (2019). Machine learning: ¿qué es y cuál es su relación con la ia? <http://otech.uaeh.edu.mx/noti/index.php/machine-learning/machine-learning-que-es-y-cual-es-su-relacion-con-la-ia/>.
- Pablo Pastor, M. (2018). *Usando redes neuronales convolucionales para convertir características visuales en estímulos sonoros*.
- Pajares, G., E., A., and De la Escalera, A. (2016a). *Conceptos y Métodos en visión por Computador*. Grupo de visión del comité español de automática, España.
- Pajares, G., E., A., and De la Escalera, A. (2016b). *Conceptos y Métodos en visión por Computador*. Grupo de visión del comité español de automática, España.
- Perez, J. G. (2015). *Estudio de Filtros Óptimos para la Eliminación de Ruido en Imágenes Digitales usando Algoritmos de Multirresolución No Lineales*.
- Rahul, A. (2019). Towards data science.
- Rivas-Asanza, W., Mazon-Olivo, B., and Mejia, F. (2018). *Capítulo 1: Generalidades de las redes neuronales artificiales*, pages 13–15.
- Rizwan, M. (2018a). Alexnet implementation using keras. <https://engmrk.com/alexnet-implementation-using-keras>.
- Rizwan, M. (2018b). Innovaciones arquitectónicas en redes neuronales. <https://engmrk.com/alexnet-implementation-using-keras>.
- Robles-Camarillo, D., Paris, S., Nakano-Miyatake, M., and Luna-González, J. (2016). Comparación de arquitecturas de redes neuronales convolucionales para la clasificación de imágenes de ojos.
- Rueda, G., Constantino, L., Gil, Z., Ortega, O., and Benavides, P. (2018). Natural enemies of the coffee leaf miner leucoptera coffeella (lepidoptera: Lyonetiidae). AV 492:8.
- Sanchez Garrido, M. A. (2012). *Matriz de confusión*.
- SENASA (2003). Servicio nacional de sanidad agraria. page 11.
- Solar, L. J. D. (2015). Filtrado de imágenes y ecualización del histograma.
- Szegedy, C. Vanhoucke, V. . I. S. . S. J. (2016). Rethinking the inception architecture for computer vision. page 2821.
- Tewari, S. (2019). Redes neuronales convolucionales.
- Tom M., M. (2001). *Machine Learning*.

- Ttito Concha, Darwin; Flores Atauchi, P. D. (2016). Desarrollo de una arquitectura de red neuronal convolucional como un modelo del proceso cerebral humano para la clasificación de expresiones faciales.
- TÚCHEZ, J. W. G. (2019). *CARACTERIZACIÓN BIOLÓGICA DEL HONGO *Mycena citricolor* Berk Curt, CON AISLAMIENTOS OBTENIDOS DE CULTIVARES DE CAFÉ (*Coffea arabica* L.) PROVENIENTES DE LAS DIFERENTES ZONAS CAFETALERAS DE GUATEMALA, C.A.*
- ul Hassan, M. (2018). Convolutional network for classification and detection.
- Varios (2015). Diferencia entre clasificación y regresión en machine learning. <http://ingenierobeta.com/clasificacion-vs-regresion-machine-learning/>.
- Zamorano Ruiz, J. (2018). *Comparativa y análisis de algoritmos de aprendizaje automático para la predicción del tipo predominante de cubierta arbórea.*
- Zulkifli, H. (2018). Understanding learning rates and how it improves performance in deep learning.