

# Detección de cochinilla mediante el uso de las características físicas del café utilizando visión artificial

Detection of cochineal using physical characteristics of coffee through artificial vision.

Herney González<sup>1\*</sup> y Johann Sebastián Kwan Alvarado<sup>2</sup>

<sup>1</sup> Estudiante, Ingeniería de software, Universidad Surcolombiana, Neiva-Colombia Email: [u20182172514@usco.edu.co](mailto:u20182172514@usco.edu.co), <https://orcid.org/0009-0009-6578-6191>

<sup>2</sup> Estudiante, Ingeniería de software, Universidad Surcolombiana, Neiva-Colombia Email: [u20191179602@usco.edu.co](mailto:u20191179602@usco.edu.co), <https://orcid.org/0000-0001-7317-9310>  
\*(autor para correspondencia)

Fecha de recepción: xx de xxxx del xxxx      Fecha de revisión: xx de xxxx del xxxx      Fecha de aprobación: xx de xxxx del xxx

## Resumen

Este artículo presenta un proyecto para la detección de enfermedades en plantas de café utilizando redes neuronales convolucionales y procesamiento de imágenes. Se describen los detalles del proyecto, incluyendo los pasos necesarios para implementar la técnica propuesta y los resultados obtenidos. Se discuten enfermedades comunes del café y se explican conceptos de redes neuronales convolucionales y métricas de medición. También se describe la metodología utilizada para construir el dataset, preprocesar las imágenes, etiquetarlas y entrenar el modelo capaz de detectar enfermedades causadas por la cochinilla Pluto Barberi. Los resultados obtenidos muestran una alta precisión en la detección de enfermedades en las imágenes. Además, se mencionan trabajos futuros como la implementación del modelo para diferentes variedades de café y el desarrollo de un sistema de detección en tiempo real. También se discuten diferentes algoritmos de detección de objetos, como R-CNN, Fast R-CNN, Faster R-CNN Y YOLO. Los resultados obtenidos muestran una alta precisión en la detección de enfermedades en las imágenes. Se logra una precisión de aproximadamente 83% en imágenes externas a las tomadas para el dataset del proyecto. Se destaca la combinación de laplaciano del Gaussiano como uno de los mejores métodos para la detección de la hoja de café, con una precisión de hasta un 95%. Este artículo es relevante para la comunidad científica interesada en la aplicación de la visión artificial y las redes neuronales convolucionales en la detección temprana de enfermedades en plantas de café.

**Palabras Clave:** Pluto Barberi; Enfermedades del café; Convolución; Procesamiento de imágenes; Precisión.

## Abstract

This article presents a project for the detection of diseases in coffee plants using convolutional neural networks and image processing. The details of the project are described, including the necessary steps to implement the proposed technique and the results obtained. Common coffee diseases are discussed, and concepts of convolutional neural networks and measurement metrics are explained. The methodology used to build the dataset, preprocess the images, label them, and train the model capable of detecting diseases caused by the coffee mealybug Pluto barberi is also described. The results obtained show high accuracy in disease detection in the images. Furthermore, future work is mentioned, such as implementing the model for different coffee varieties and developing a real-time detection system. Different object detection algorithms, such as R-CNN, Fast R-CNN, Faster R-CNN, and YOLO, are also discussed. The results obtained show high accuracy in disease detection in the images. An accuracy of approximately 83% is achieved in images external to those taken for the project dataset. The combination of Gaussian Laplacian is highlighted as one of the best methods for coffee leaf detection, with an accuracy of up to 95%. This article is relevant to the scientific community interested in the application of artificial vision and convolutional neural networks in the early detection of diseases in coffee plants.

**Keywords:** Pluto Barberi; Coffee diseases; Convolution; Image processing; Precision

## 1. INTRODUCCIÓN

La producción de café es una actividad económica de gran importancia en muchos países del mundo. Sin embargo, esta actividad se ve afectada por diversas enfermedades que pueden disminuir la calidad y cantidad de la producción. Una de estas enfermedades es la cochinilla puto Barberi, la cual puede ser difícil de detectar y controlar. Ante esta problemática, se han desarrollado diversas investigaciones en el campo de la visión artificial aplicada al proceso de producción de café. En este sentido, se han utilizado diferentes técnicas y metodologías para la detección y clasificación de las hojas afectadas por la cochinilla. En este contexto, el presente artículo tiene como objetivo principal desarrollar un sistema de visión artificial para la detección y clasificación de la cochinilla puto Barberi. Para ello, se realizará una recopilación de imágenes y se llevará a cabo una investigación a fondo sobre los conceptos y componentes necesarios para el correcto funcionamiento del sistema. La hipótesis de esta investigación es que la utilización de sistemas neuronales convolucionales y la teoría de color HSV permitirán una detección y clasificación más precisa de la cochinilla en los colinos de café. La originalidad de esta contribución radica en la utilización de técnicas de visión artificial para la detección y clasificación de la mancha de hierro en hojas de café, lo cual puede mejorar significativamente la calidad y cantidad de la producción de café. Además, esta investigación puede sentar las bases para futuros estudios en el campo de la visión artificial aplicada a la producción de café y otras actividades agrícolas.[1][2][3]

## 2. METODOLOGÍA

Para llevar a cabo este proyecto, se utilizó un sistema de visión artificial que utilizará redes neuronales convolucionales, por lo cual es necesario la elaboración de un dataset adecuado al problema inicial propuesto.

### 2.1 Construcción del dataset

Para el dataset se tomaron 1200 imágenes de hojas de colinos de café variedad F1 en almácigos. Distribuidos en tres estados (sano, poco afectado y

muy afectado) de 400 imágenes cada uno, consecuentemente se les aplicó una técnica a partir de conceptos de Data Augmentation con el fin de incrementar el tamaño del dataset. Las imágenes se van a capturar directamente en los almácigos de café. El proceso de capturas se va a llevar a cabo dentro del siguiente horario: De 4:00 pm a 6:00 pm. Este intervalo de tiempo favorece la recolección, pues los rayos del sol son menos fuertes que al medio día y esto hace que las imágenes resulten más cálidas y no muy quemadas. Las hojas de café se van a capturar entre los estados explicados anteriormente, lo que va a ayudar a elaborar un modelo robusto más adelante en el tratamiento de esas imágenes conseguidas.

### 2.2 Data Augmentation

Con esta técnica facilita la incrementación del dataset a partir de diferentes transformaciones. Así pues, las 400 imágenes de cada grupo pasaron por un algoritmo en el que se aplicó un conjunto de transformaciones combinadas que se describen a continuación:

En el algoritmo que se construyó se usó una biblioteca de redes neuronales llamada Keras, la cual es de código abierto y está escrita en el lenguaje de programación Python. Se usa una clase de keras llamada ImageDataGenerator, que precisamente es la que permite configurar las transformaciones de las imágenes del dataset.[4]

Lo primero que se hizo, fue definir el número de transformaciones que iba a sufrir cada fotografía, el cual fue cuatro, esto con el fin de lograr un total de 4800 imágenes y así obtener un dataset de tamaño considerable para trabajar después.

Como primera función, se aplica un rango de rotación de 20°. Luego, se implementa un rango de zoom de 0.2. Las siguientes transformaciones se muestran en forma de lista a continuación:

Rango de desplazamiento de ancho: 10%

Rango de cambio de altura: 10%

Como última configuración, a las imágenes se les aplicó una inversión horizontal.

Como resultado, en cada grupo se obtuvieron 1200 fotografías. Por tanto, se tiene al final un dataset conformado por un número importante de estas imágenes conservando el tamaño de cada una.

### 2.3 Preprocesamiento de imágenes:

Se procesa las imágenes recolectadas en el dataset que constan de una altura de 3472 píxeles y un ancho de 4624 píxeles respectivamente, con el fin de hacer más pequeño el tamaño de las imágenes y de esta manera poder disminuir el tamaño del programa y no perder información importante de las imágenes, por lo cual el tamaño al que se reducirá será de 240 por 240 píxeles.[5]

### 2.4 Distribución del dataset.

Entrenamiento		80%		Validación		20%	
3840 imágenes				960 imágenes			
No afectada	afecta da	Muy afectada	No afectada	afecta da	Muy afectada		
1280	1280	1280	320	320	320		
Total:4800							

### 2.5 Métrica

La métrica que se tendrá en cuenta para este proyecto será la de precisión ya que se evaluará que tan eficiente es el programa en cuanto a la predicción que puede hacer detectando características especiales que indiquen en qué fase de la enfermedad se encuentra la planta de café, por ende se hará uso de la fórmula de precisión a expuesta en el marco teórico.

### 2.6 Etiquetado de imágenes

Esta es una tarea muy importante y requiere una gran atención, pues entre mejor etiquetadas está en las imágenes, el modelo va a detectar con mayor precisión. Este proceso al ser manual tardó alrededor de 10 horas y se realizó por medio de una herramienta de software llamada labeling. Este sencillo programa permitió generar un archivo XML por cada imagen etiquetada, el cual contiene las coordenadas de cada una de esas etiquetas en la imagen. Esto es importante porque el modelo va a saber el lugar donde están los deterioros que se quiere detectar.

### 2.7 Tratamiento de imágenes etiquetadas

Una vez se distribuyen las imágenes etiquetadas en los directorios de entrenamiento y test es necesario convertirlas en un formato de almacenamiento binario de tensorflow llamado TFRecords. Entonces, mediante un algoritmo se especifica el directorio

donde se encuentran las imágenes junto con su mapa de etiquetas y un directorio para guardarlos TFRecords. Luego de obtenidos ya es momento de configurar el script que se encargará de entrenar el modelo.

### 2.8 Uso de un modelo pre entrenado

Se va a utilizar el api de tensorflow, es decir, se hará uso de un algoritmo de detección de objetos que tiene sus capas intermedias ya entrenadas por tensorflow que viene a ser lo más pesado en cuanto a procesamiento durante un entrenamiento. Por ende se tomará un modelo llamado Movilenet\_v2 donde solo se procederá a entrenar la última capa convolucional.

### 2.9 Configuración del modelo

El algoritmo pre entrenado está integrado por unos directorios y hay uno de ellos que se configura específicamente para nuestro dataset. Es importante conocer los parámetros que están presentes en él para poder consecuentemente realizar los ajustes que se puedan adaptar no solo al proyecto sino también a la capacidad de los componentes de nuestra máquina de trabajo. De esta manera, se ajustaron las siguientes variables para la detección de hojas no afectadas, afectadas y muy afectadas:

Parámetros de ajuste del modelo	
Model	Movilenet_v2
Numero de clases	3
Nombre de las clases	No afectada
	Afectada
	Muy afectada
Épocas	5,10,15,20,50
Batch size	32
Total parametros	2.261.827
Trainable params	3.843

Tabla 1 parámetros de ajuste del modelo

### 2.10 Entrenamiento del modelo

Para el entrenamiento del modelo se tomaron ciertas características especiales que son claves a la hora de generar una métrica en el entrenamiento, por ende se procede a iniciar un entrenamiento con 5 épocas en las que se dejaron los valores

Entre namiento	loss	accuacy	Val_lo ss	Val_ acuracy
1	0.3876	0.8242	0.1167	0.9810
2	0.2343	0.8824	0.5142	0.9873

3	0.0343	0.9242	0.4983	0.9978
4	0.0165	0.9984	0.0357	0.9945
5	0.0099	1.0000	0.0224	1.0000

Tabla 2 entrenamiento del modelo

### 2.11 Predicciones

Cuando culmina el entrenamiento del modelo, se genera un directorio que tensorflow entiende para luego empezar a realizar pruebas. Hecho esto, se corre un último comando el cual realiza predicciones de imágenes situadas en un directorio específico. El modelo leerá las imágenes que tengamos allí, y estas deben estar en formato jpg. En el proceso se generan matrices con las anotaciones que el modelo detecta en cada imagen.

### 2.12 Métrica de Evaluación

Para evaluar los resultados del modelo entrenado, se hizo uso de una métrica de evaluación popular dentro de la detección de objetos y la visión por computadora. Según el marco teórico, mAP nos permite evaluar nuestro modelo aplicando cálculos básicos con respecto a las predicciones que el modelo hizo.[14]

## 3. RESULTADOS Y DISCUSIÓN

En este capítulo se evalúa la precisión del modelo que se trabajó. Para ello se utilizará la Métrica de evaluación que se abordó en el Marco teórico. Se tiene entonces un grupo de ecuaciones que permiten encontrar el porcentaje de precisión en la detección de cada clase de una imagen, para luego realizar una sumatoria y obtener finalmente la precisión global del modelo en toda la imagen.

Tenemos entonces nuestras tres clases: no afectado, afectado y muy afectado. En esta evaluación, se estudiará la precisión de nuestro modelo en las siguientes imágenes tomadas de la web con el fin de identificar imágenes que no son propias del dataset del proyecto:

- Hoja afectada



Ilustración 1 hoja afectada

- Hoja muy afectada



Ilustración 2 hoja muy afectada

- Hoja no afectada



Ilustración 3 hoja no afectada

- Resultado de una hoja afectada

```

#url = 'https://mundocafeto.com/wp-content/uploads/2018/05/hoja-cafeto-1.jpg' #debe ser 2
#url = 'https://quecafe.info/wp-content/uploads/2020/02/Roya-1..png'
url = 'https://i.ytimg.com/vi/V2FI8HUfFgk/hqdefault.jpg'
prediccion = categorizar(url)
resultado = prediccion
if resultado <= .5 :
    print(resultado)
    print("no afectada")
else:
    if resultado > .5 and resultado <= 1.5:
        print(resultado)
        print("afectada")
    else:
        print(resultado)
        print("muy afectada")

```

1  
afectada

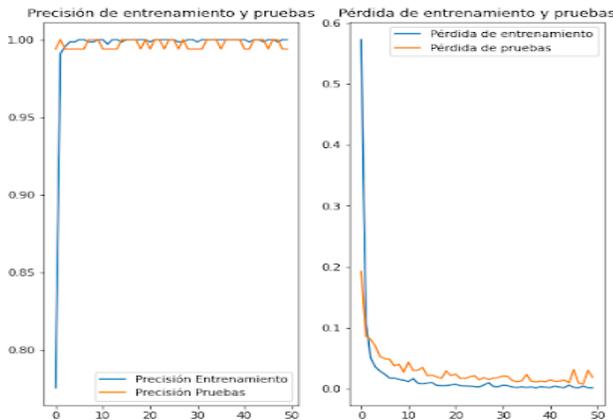
Ilustración 4 resultado hoja afectada

El modelo pre-entrenado Movilenet\_v2 ha demostrado su eficacia al arrojar un resultado correcto al analizar una imagen de una hoja afectada de internet. Esto indica que el modelo se encuentra bien entrenado y cuenta con las características necesarias para reconocer y clasificar hojas afectadas de manera precisa. En definitiva, los resultados obtenidos son una muestra de la efectividad de este modelo pre-entrenado.

### 3.1 Simulación

En esta simulación, se ha utilizado un total de 50 épocas para garantizar que el modelo pre-entrenado se entrena adecuadamente y pudiera realizar una clasificación precisa. Al completar estas 50 épocas, el modelo ha adquirido el conocimiento y la capacidad necesarios para clasificar de manera excelente.

épocas	50
Batch size	32
Val_accuracy	0,9876
Loss_accuracy	0,0128



La imagen muestra la precisión del entrenamiento y de pruebas, con un resultado obtenido del 0,98 de

precisión. Además, muestra la pérdida del entrenamiento con un resultado obtenido del 0,012.

### 3.2 Matriz de precisión

Al realizar la matriz de precisión se logró obtener los resultados de los verdaderos positivos y falsos positivos.

		Predicted label		
		No afectadas	Afectadas	Muy afectadas
True label	No afectadas	11	2	0
	Afectadas	1	10	2
	Muy afectadas	0	2	11

Tabla 3 matriz de precisión

Se obtuvo en la métrica de precisión dentro de las no afectadas 11 de 12 verdaderos positivos, lo que indica que solo se tuvo un falso positivo; dentro de las afectadas se obtuvieron 10 de 14 verdaderos positivos y 4 falsos positivos; y en las muy afectadas de obtuvo 11 de 13 verdaderos positivos, con 2 falsos positivos. Se hace una comparación con la métrica recall, como se muestra en la Tabla 4.

	precisión	recall
No afectada	$\frac{11}{12} = 0,9166$	$\frac{11}{13} = 0,8461$
Afectada	$\frac{10}{14} = 0,7142$	$\frac{10}{13} = 0,7692$
Muy afectada	$\frac{11}{13} = 0,8461$	$\frac{11}{13} = 0,8461$
Total	0,83 = 83 %	0,82 = 82%

Tabla 4 matriz de precisión 2

El resultado obtenido en la métrica de precisión fue de un 83%, mientras que en la métrica recall fue de 82%. Estos resultados son necesarios para evaluar la métrica F1-score como se muestra en la Tabla 5.

F1-score	$\frac{(2 \cdot 0,83) \cdot 0,82}{0,83 + 0,82} = \frac{1,3612}{1,65} = 0,83$ $= 83\%$
----------	---

Tabla 5 fl-score

Como se observa, el F1-score de las imágenes es alto, esto debido a la buena detección que hace el algoritmo pese a algunas pruebas que se hicieron con imágenes descargadas de la web (Imágenes 1, 2 y 3).

#### 4. CONCLUSIONES Y TRABAJOS FUTUROS

A continuación, se explican las conclusiones de este proyecto y algunos de los posibles trabajos futuros que puedan darse como resultado de este estudio.

En base a la tabla anterior, concluimos que, teniendo en cuenta los resultados de detección del modelo, las métricas arrojan buenos porcentajes con imágenes del dataset. No obstante, se observó la baja precisión en imágenes externas a las tomadas para el dataset del proyecto (Considerar en la tabla de métricas los porcentajes de las imágenes 1, 2 y 3). Esto tiene una explicación sencilla. Este modelo se entrenó con imágenes que tenían propiedades específicas (Distancia focal estándar, dimensiones iguales, velocidad de obturación, fondo blanco, poco ruido en las imágenes, entre otras) que se mantuvieron a lo largo de su entrenamiento, entonces cuando ingresamos imágenes con atributos diferentes el modelo no va a detectar los granos con igual precisión. No obstante, se comprueba que el algoritmo hace el intento y teniendo en cuenta el modelo matemático, se logra una precisión de aproximadamente 83%.

##### Trabajos futuros

Como continuación de este proyecto y como la mayoría de los proyectos de Ingeniería, quedan líneas de investigación abiertas para seguir estudiando. Durante el desarrollo de este proyecto surgieron algunas líneas que se esperan atacar en un futuro; estas surgieron a causa de cuestionamientos que se originan en el proceso de desarrollo de este trabajo. A continuación, se presentan algunos trabajos futuros que pueden apoyar y mejorar el modelo desarrollado en este estudio. Estos trabajos son:

- Implementar el modelo a diferentes variedades de café presentes en el departamento del Huila, con el fin de tener algoritmos específicos para cada variedad y detectar cómo afecta la cochinilla a estas variedades.

- Desarrollar un prototipo que permite en conjunto con estos algoritmos darle un valor añadido al proyecto, realizando no solo la detección de las características propias de la plaga Pluto Barberi que se visualizan en los almácigos de café sino también aplicada a los cultivos ya en producción.

- Modificar el algoritmo para que una vez entrenado, no detecte por medio de imágenes, sino que lo haga en tiempo real, es decir, por medio de video.

#### REFERENCIAS

- [1] Z. E. G. VILLEGAS G., A. RAMOS P. y P. BENAVIDES M., identificación y hábitos de la cochinilla arinosa asociadas a las raíces de café en quindío, *quindío: cenicafe* 60(4):362-373, 2009.
- [2] cenicafe, «cenicafe, centro nacional de investigaciones de café,» [En línea]. Available: [https://www.cenicafe.org/es/index.php/cultivemos\\_cafe/manejo\\_de\\_enfermedades#:~:text=La%20roya%20del%20cafe%20es,distribuci%C3%B3n%20y%20cantidad%20de%20lluvia..](https://www.cenicafe.org/es/index.php/cultivemos_cafe/manejo_de_enfermedades#:~:text=La%20roya%20del%20cafe%20es,distribuci%C3%B3n%20y%20cantidad%20de%20lluvia..) [Último acceso: 22 febrero 2022].
- [3] E. O. G. G. C. A. R. O. c. E. O. T. Oscar a. Gusman P, utilización del procesamiento de imágenes para determinar la severidad de la mancha de hierro, en hojas de café., 2003.
- [4] H. I. C. K. L. V. C. M. J. V. Luis Alberto Vives Garnique, «VISIÓN ARTIFICIAL: APLICACIÓN DE FILTROS Y SEGMENTACIÓN EN IMÁGENES DE HOJAS DE CAFÉ,» *INGENIERIA: CIENCIA, TECNOLOGIA E INNOVACION*, vol. 1, pp. 73-80, 2014-07-30.
- [5] G. Kuntimad and H. S. Ranganath, «IEEE Transactions on Neural Networks,» *Perfect image segmentation using pulse coupled neural networks,*, vol. 10, n° 3, pp. 591-598, 1999.
- [6] J. Hui, «MEDIUM,» 06 marzo 2018. [En línea]. Available: <https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173>. [Último acceso: 15 febrero 2022].