

Type of the Paper (Article)

Algoritmo de clasificación de enfermedades en la hoja de café

Paula Rojas ¹, Diego Sandoval ² and Brian Sandoval ³

¹ Universidad Libre; paulaa-rojasv@unilibre.edu.co

² Universidad Libre; diegoa-sandovalf@unilibre.edu.co

³ Universidad Libre; briann-sandovalm@unilibre.edu.co

Received: 14/01/2021; Accepted: 18/04/2021.; Published: 30/06/2021,

Resumen: El interés del proyecto es a través del análisis de las hojas de café, determinar si la planta está sana o si contiene alguna de las siguientes 2 plagas: Roya y Minador, tomando así las respectivas medidas de cuidado para poder controlar estas plagas en caso de que se produzcan y así poder reducir la pérdida del cultivo del café. Para esto se realizará un algoritmo de clasificación desarrollado en Python el cual utiliza diferentes herramientas y técnicas de aprendizaje supervisado, junto con varias imágenes de diferentes tipos de hojas de café: unas sanas y las otras afectadas por alguna de las plagas a estudiar.

Abstract: The interest of the project is through the analysis of the coffee leaves, to determine if the plant is healthy or if it contains any of the following 2 pests: Rust and Miner, thereby taking the respective care measures to be able to control these pests in case that these occur and thus reduce the loss of coffee cultivation. For this, a classification algorithm developed in Python will be carried out, which will use different tools and supervised learning techniques, together with several images of different types of coffee leaves: some healthy and others affected by some of the pests to be studied.

Keywords: café; aprendizaje supervisado; roya; plaga minera.

1. Introducción

El café es muy importante para Colombia, pues a nivel mundial, nuestro país ocupa la posición número 3 de los países que más producen este producto [1], pero para lograr esto se ha tenido que enfrentar a diferentes enfermedades y plagas que dañan la producción, como lo es la roya y el minador, las cuales afectan las hojas de la planta, provocando que el nivel de producción de esta se vea drásticamente afectada [2]. Debido a esto, este trabajo quiere presentar una herramienta para que los cafeteros diagnostiquen a sus plantas oportunamente y tomen las medidas correspondientes, ayudando a que no se vea afectada la economía de todos ellos, así que para lograrlo se ha revisado los distintos estudios presentes, como: el artículo llamado "Sistemas de producción de café en Colombia", donde se presentan investigaciones hechas para mejorar la producción del café [3], "Clima andino y café en Colombia" en él se habla sobre la relación que

existe entre el clima, la producción, las enfermedades y otros aspectos [4], y también otro artículo, “La variedad Colombia: selección de un cultivar compuesto resistente a la roya del cafeto”, en el que se habla sobre la importancia de seleccionar semillas modificadas genéticamente para que tengan una mayor resistencia a la roya [5].

Así que este trabajo presenta un sistema de clasificación de imágenes que permite determinar si una hoja de café tiene una enfermedad u otra, para que así el productor tome las acciones pertinentes. Para ello ha sido necesario contar con una enorme cantidad de fotos correspondientes a hojas con la roya, el minador y saludables, para que así el sistema al momento de recibir la fotografía de la hoja que el productor quiere evaluar, determine correctamente la enfermedad que esta tiene, o si está saludable.

2. Materiales y Métodos

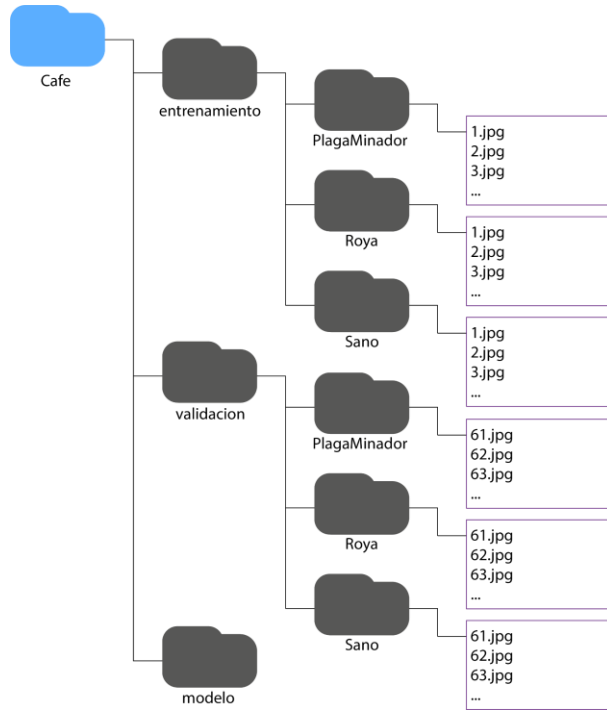
Para la clasificación de hojas de café se ha usado una gran cantidad de imágenes, con ellas se busca detectar uno o más aspectos, así que para lograrlo se necesita los siguientes materiales:

- Plantas de café
- Hojas con la enfermedad de la Roya.
- Hojas con la plaga Minador.
- Hojas sanas.
- Una cámara para obtener sus fotografías.
- Archivo zip.
- Directorio.
- El lenguaje de programación Python plasmado en un colab.

Por otro lado en python se hizo uso de las herramientas planteadas en clase para realizar la categorización de la data por medio de métodos como:

- NumPy.
- Importación de librerías (Keras, tensorflow y sklearn).
- Parámetros.
- Cargar imagen.
- Creación de etiquetas y clases.
- API's.
- Algoritmo supervisado.

Teniendo todos los materiales mencionados anteriormente se pasa a organizar las imágenes, pues para realizar este proyecto es necesario manejar una estructura de carpetas de una forma especial, tal como se muestra en la *imagen 1*.



Imágen 1. Estructura de carpetas

En la *imágen 1* se ve que se parte de una carpeta llamada **Cafe**, en ella se encuentran tres carpetas llamadas **entrenamiento**, **validación** y **modelo**, las dos primeras son las que contendrán las imágenes pero separadas en carpetas, donde cada carpeta corresponde a una categoría, para este proyecto dichas carpetas son **PlagaMinador**, **Roya** y **Sano**, y la última carpeta, **modelo**, es la que contendrá dos archivos generados por el código usado, los cuales nos va a permitir clasificar sin la necesidad de que el algoritmo entrene nuevamente.

Se usaron en total 267 imágenes, de las cuales 120 se destinaron para la carpeta de **validacion**, 40 para cada categoría y las restantes, 147, para **entrenamiento**, 49 para cada categoría.

Ya con la imágenes almacenadas y organizadas correctamente se pasa a manipular el código, para ello es necesario usar las siguientes librerías:

```

import sys
import os
from tensorflow.python.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras import optimizers
from tensorflow.python.keras.models import Sequential
from tensorflow.python.keras.layers import Dropout, Flatten, Dense, Activation
from tensorflow.python.keras.layers import Convolution2D, MaxPooling2D
from tensorflow.python.keras import backend as K
from tensorflow.python.keras.optimizer_v2.adam import Adam
    
```

Con estas librerías se usa el código de entrenamiento, el cual se encarga de entrenar el modelo, es decir usar todas las imágenes que se tienen para determinar cuáles son las características de cada categoría que permitan diferenciarse entre sí este código es el que se muestra a continuación;

```
K.clear_session()
data_entrenamiento = '/content/drive/MyDrive/Cafe/entrenamiento'
data_validacion = '/content/drive/MyDrive/Cafe/validacion'

"""
Parameters
"""
epocas=10
longitud, altura = 100, 100
batch_size = 32
pasos = 5
validation_steps = 4
filtrosConv1 = 32
filtrosConv2 = 64
tamano_filtro1 = (3, 3)
tamano_filtro2 = (2, 2)
tamano_pool = (2, 2)
clases = 3
lr = 0.0004

##Preparamos imágenes
entrenamiento_datagen = ImageDataGenerator(
    rescale=1. / 255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True
)

test_datagen = ImageDataGenerator(
    rescale=1. / 255
)

entrenamiento_generador = entrenamiento_datagen.flow_from_directory(
    data_entrenamiento,
    target_size=(altura, longitud),
    batch_size=batch_size,
    class_mode='categorical'
)

validacion_generador = test_datagen.flow_from_directory(
    data_validacion,
    target_size=(altura, longitud),
    batch_size=batch_size,
    class_mode='categorical'
)
```

```

#Crear CNN
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'))
adam = Adam()
cnn.compile(
    loss='categorical_crossentropy',
    optimizer=adam,
    metrics=['accuracy']
)
cnn.fit(
    entrenamiento_generador,
    steps_per_epoch=pasos,
    epochs=epocas,
    validation_data=validacion_generador,
    validation_steps=validation_steps
)
target_dir = '/content/drive/MyDrive/Cafe/modelo/'
if not os.path.exists(target_dir):
    os.mkdir(target_dir)
cnn.save('/content/drive/MyDrive/Cafe/modelo/modelo.h5')
cnn.save_weights('/content/drive/MyDrive/Cafe/modelo/pesos.h5')

```

Como se puede apreciar en la segunda y tercera línea se indica la ruta donde se encuentran las imágenes para entrenar y validar el modelo, luego se definen una serie de parámetros que van a determinar ciertas características propias de las Redes Neuronales Convolucionales, después se preparan las imágenes, el entrenamiento y validación, para que finalmente se cree la CNN que es la que se encargará de generar el modelo gracias a los parámetros y alistamiento de las imágenes hecho anteriormente. El modelo y sus pesos son almacenados en la carpeta **modelo**.

El otro código es el que usará dicho modelo entrenado y validado para determinar a qué categoría pertenece alguna imagen que se pase, a continuación se muestra dicho código:

```

import numpy as np
from keras.preprocessing.image import load_img, img_to_array
from keras.models import load_model
longitud, altura = 100, 100
modelo = '/content/drive/MyDrive/Cafe/modelo/modelo.h5'
pesos_modelo = '/content/drive/MyDrive/Cafe/modelo/pesos.h5'
cnn = load_model(modelo)
cnn.load_weights(pesos_modelo)

```

```

def predict(file):
    x = load_img(file, target_size=(longitud, altura))
    x = img_to_array(x)
    x = np.expand_dims(x, axis=0)
    array = cnn.predict(x)
    result = array[0]
    answer = np.argmax(result)
    if answer == 0:
        print("pred: Plaga minador")
    elif answer == 1:
        print("pred: Plaga roya")
    elif answer == 2:
        print("pred: Sano")

    return answer
predict('/content/drive/MyDrive/Cafe/minador.jpg')

```

En este se especifica la ruta de la imagen que se quiere clasificar, la cual se pasa como argumento en una función que se encarga de usar el modelo creado para determinar a qué categoría pertenece.

3. Resultados

Teniendo el modelo se ha probado con 10 imágenes de cada categoría para determinar si el modelo cumple o no con su trabajo, para ello se modificó un poco el segundo código mencionado anteriormente para que así prediga todas las imágenes de una vez, el segmento de código que se añadió es el que se muestra a continuación:

```

print("****Imágenes de la plaga minador****")
predict('/content/drive/MyDrive/Cafe/prueba/PlagaMinador/101.jpg')
predict('/content/drive/MyDrive/Cafe/prueba/PlagaMinador/102.jpg')
predict('/content/drive/MyDrive/Cafe/prueba/PlagaMinador/103.jpg')
predict('/content/drive/MyDrive/Cafe/prueba/PlagaMinador/104.jpg')
predict('/content/drive/MyDrive/Cafe/prueba/PlagaMinador/105.jpg')
predict('/content/drive/MyDrive/Cafe/prueba/PlagaMinador/106.jpg')
predict('/content/drive/MyDrive/Cafe/prueba/PlagaMinador/107.jpg')
predict('/content/drive/MyDrive/Cafe/prueba/PlagaMinador/108.jpg')
predict('/content/drive/MyDrive/Cafe/prueba/PlagaMinador/109.jpg')
predict('/content/drive/MyDrive/Cafe/prueba/PlagaMinador/110.jpg')

print("****Imágenes de la roya****")
predict('/content/drive/MyDrive/Cafe/prueba/Roya/101.jpg')
predict('/content/drive/MyDrive/Cafe/prueba/Roya/102.jpg')
predict('/content/drive/MyDrive/Cafe/prueba/Roya/103.jpg')
predict('/content/drive/MyDrive/Cafe/prueba/Roya/104.jpg')
predict('/content/drive/MyDrive/Cafe/prueba/Roya/105.jpg')
predict('/content/drive/MyDrive/Cafe/prueba/Roya/106.jpg')
predict('/content/drive/MyDrive/Cafe/prueba/Roya/107.jpg')
predict('/content/drive/MyDrive/Cafe/prueba/Roya/108.jpg')

```

```

predict('/content/drive/MyDrive/Cafe/prueba/Roya/109.jpg')
predict('/content/drive/MyDrive/Cafe/prueba/Roya/110.jpg')

print("****Imágenes de hojas sanas****")
predict('/content/drive/MyDrive/Cafe/prueba/Sano/101.jpg')
predict('/content/drive/MyDrive/Cafe/prueba/Sano/102.jpg')
predict('/content/drive/MyDrive/Cafe/prueba/Sano/103.jpg')
predict('/content/drive/MyDrive/Cafe/prueba/Sano/104.jpg')
predict('/content/drive/MyDrive/Cafe/prueba/Sano/105.jpg')
predict('/content/drive/MyDrive/Cafe/prueba/Sano/106.jpg')
predict('/content/drive/MyDrive/Cafe/prueba/Sano/107.jpg')
predict('/content/drive/MyDrive/Cafe/prueba/Sano/108.jpg')
predict('/content/drive/MyDrive/Cafe/prueba/Sano/109.jpg')
predict('/content/drive/MyDrive/Cafe/prueba/Sano/110.jpg')

```

De esta manera se obtienen los 30 resultados de las 30 imágenes dispuestas para tal fin, el resultado obtenido es el que se muestra a continuación:

```

****Imágenes de la plaga minador****
pred: Minador
pred: Roya
pred: Roya
pred: Minador
pred: Minador
pred: Roya
pred: Minador
pred: Minador
pred: Minador
pred: Minador
****Imágenes de la roya****
pred: Minador
pred: Minador
pred: Minador
pred: Minador
pred: Minador
pred: Roya
pred: Roya
pred: Roya
pred: Roya
****Imágenes de hojas sanas****
pred: Sano
pred: Roya
pred: Minador
pred: Minador
pred: Minador
pred: Roya
pred: Sano
pred: Sano
pred: Sano
pred: Minador

```

Este resultado corresponde a los siguientes parámetros establecidos en el primer código mencionado en el presente artículo:

```

epocas= 10
longitud, altura = 100, 100
batch_size = 20
pasos = 8
validation_steps = 4
filtrosConv1 = 32
filtrosConv2 = 64
tamano_filtro1 = (3, 3)
tamano_filtro2 = (2, 2)
tamano_pool = (2, 2)
clases = 3
lr = 0.0004
    
```

A partir de dichos resultados se elaboró la *tabla 1* para conocer el porcentaje de éxito:

Categoría	Cantidad imgs	Aciertos	Porcentaje de éxito
Minador	10	7/10	70%
Roya	10	5/10	50%
Sano	10	4/10	40%
TOTAL	30	16/30	53.3%

Tabla 1. Porcentaje de éxito

4. Discusión

Los resultados no fueron los mejores, pero es necesario tener en cuenta que hay factores que influyen, como lo es la cantidad de imágenes, las cuales no son las suficientes para que el algoritmo elabore un modelo con más precisión, además de que la calidad de las fotos también pudo influir.

5. Conclusiones

Es útil tener una clasificación de enfermedades en la planta del café y más aún cuando al visualizar las plantas parece ser la misma enfermedad, por ello al tener la clasificación se puede tratar, estudiar de manera adecuada la enfermedad de la planta sin tener el error de pensar que se trata de lo mismo, así como también es necesario saber cuándo la planta está sana para no afectar el resto del cultivo.

References

[1] Wikipedia, «Países por producción de café,» 03 Abril 2021. [En línea]. Available: https://es.wikipedia.org/wiki/Anexo:Pa%C3%ADses_por_producci%C3%B3n_de_caf%C3%A9. [Último acceso: 12 Octubre 2021].

[2] CropLife, «Roya del cafeto,» [En línea]. Available: <https://www.croplifela.org/es/plagas/listado-de-plagas/roya-del-cafeto> . [Último acceso: 13 Octubre 2021].

[3] P. Arcila, V. Farfan, B. Moreno, G. Salazar y G. Hincapie, «Sistemas de producción de café en Colombia,» Mayo 2007. [En línea]. Available: <https://biblioteca.cenicafe.org/handle/10778/720>. [Último acceso: 12 Octubre 2021].

[4] J. Robledo, «Clima andino y café en Colombia,» 2005. [En línea]. Available: <http://www.sidalc.net/cgi-bin/wxis.exe/?IsisScript=bac.xis&method=post&formato=2&cantidad=1&expresion=mfn=043826>. [Último acceso: 12 Octubre 2021].

[5] J. Zapata, «La variedad Colombia: selección de un cultivar compuesto resistente a la roya del cafeto.,» 1988. [En línea]. Available: <http://www.sidalc.net/cgi-bin/wxis.exe/?IsisScript=bac.xis&method=post&formato=2&cantidad=1&expresion=mfn=015183>. [Último acceso: 12 Octubre 2021].

[6] AMP Tech, «Cómo hacer un clasificador de imagenes desde cero con Tensorflow» 19 Julio 2018. [En línea]. Available: https://www.youtube.com/watch?v=EAqb20_4Rdg. [Último acceso: 20 Octubre 2021].