Liliana Solorio Cázares
Axel Isidro Quiroz Avalos

Proyecto
Visión Computacional

Tabla de contenidos

01

Base de datos

Link:
https://www.kaggle.com/datasets/farukalam/tomato-leaf-diseases-detection-computer-vision

Nombre de la base de datos

Tomato Leaf Diseases Detection Computer Vision

Características

La base de datos consiste de una base de imágenes pre-procesadas con:

- Auto-orientación de los datos de píxeles (con eliminación de la orientación EXIF)
- Redimensión a 640x640 (Strech)

Separadas en 3 carpetas:

- train
- test
- valid

Características

Cada imagen una relacionada con una con una etiqueta que indicaba las enfermedades detectadas en la hoja fotografiada:

- Bacterial Spot
- Early Blight
- Healthy
- Late Blight
- Leaf Mold
- Target Spot
- Black Spot

Escritas en formato YOLO v5

02

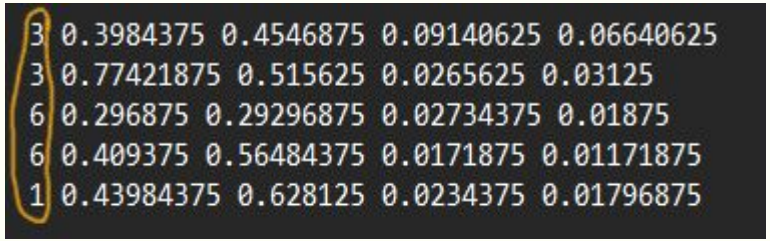
Objetivo

Crear un modelo para evaluar el estado de salud de una hoja de tomate, identificando si está sana o enferma.

Primer reto

Encontrar una manera de crear una tabla csv que indique a qué clase pertenece cada imagen. Esto con el objetivo de hacer más rápida y sencilla la lectura de los datos

El dataset contaba con documentos .txt en dónde se indicaba a qué clases pertenecía cada imagen (Indicada con amarillo)



Se logró obtener un archivo .csv que nos da información completa de a qué clases pertenece cada imagen

ID	Bacterial_spot	Early_blight	Healt	Late_blight	Leaf_mold	Target_spot	Black_spot
IMG_1021_JPG.rf.5(	0	0	1	0	0	0	0
IMG_0341_JPG.rf.c3	0	1	0	1	0	0	0
IMG_1036_JPG.rf.d7	1	1	0	0	0	0	0
IMG_0362_JPG.rf.2c	0	0	0	0	0	1	0
IMG_1116_JPG.rf.36	0	0	1	0	0	0	0

03

CNN

Modelo de la CNN

Para la creación de este modelo se utilizaron:

- 2 capas de convolución (Conv2D)
- 4 capas de activación (Activation)
- 1 capa de pooling (MaxPooling2D)
- 1 capa de aplanamiento (Flatten)
- 2 capas densas (Dense)
- 1 capa de regularización (Dropout)

Layer (type)	Output Shape	Param #
conv2d_2 (Conv2D)	(None, 62, 62, 16)	160
activation_4 (Activation)	(None, 62, 62, 16)	0
max_pooling2d_1 (MaxPooling2D)	(None, 31, 31, 16)	0
conv2d_3 (Conv2D)	(None, 29, 29, 8)	1160
activation_5 (Activation)	(None, 29, 29, 8)	0
flatten_1 (Flatten)	(None, 6728)	0
dense_2 (Dense)	(None, 32)	215328
activation_6 (Activation)	(None, 32)	0
dropout_1 (Dropout)	(None, 32)	0
dense_3 (Dense)	(None, 2)	66
activation_7 (Activation)	(None, 2)	0

Total params: 216714 (846.54 KB)
Trainable params: 216714 (846.54 KB)
Non-trainable params: 0 (0.00 Byte)

Entrenamiento de la CNN

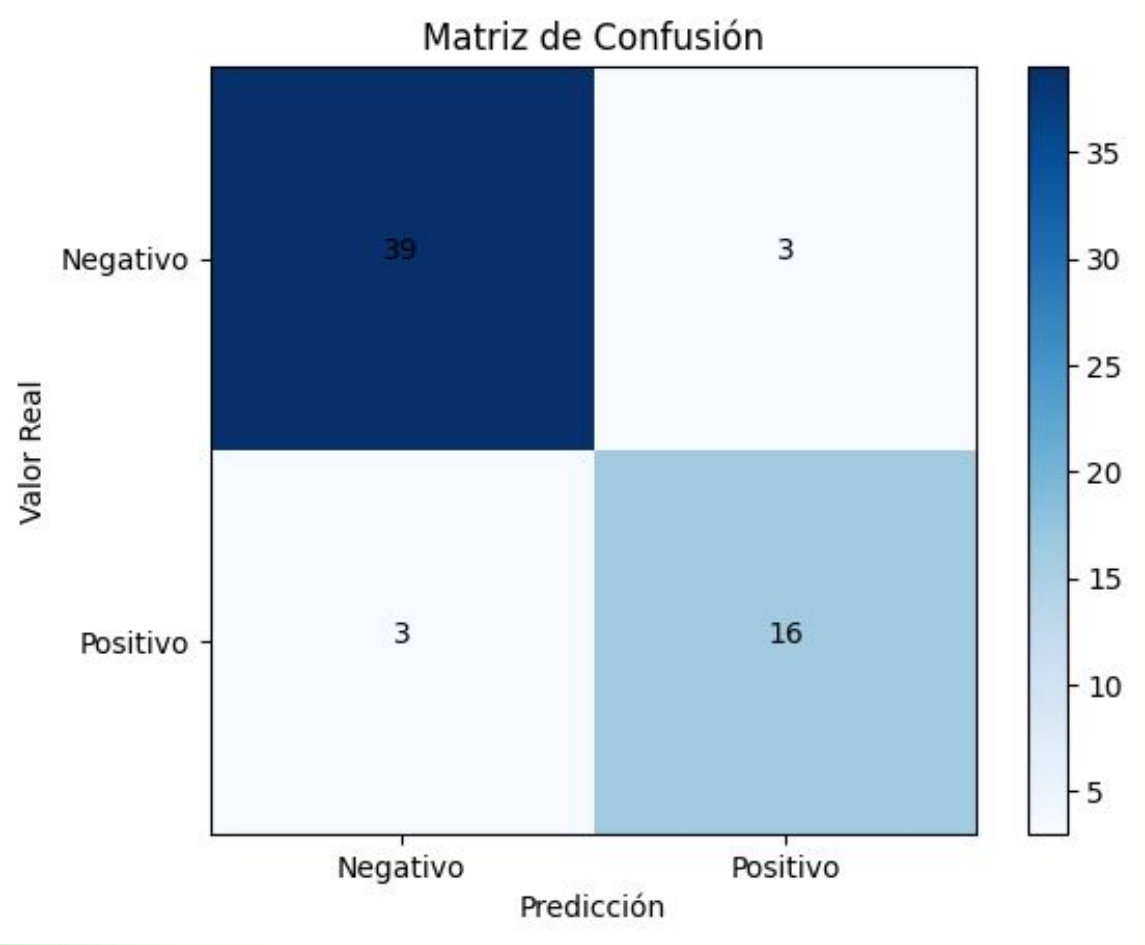
Se realizó un entrenamiento con los datos crudos (en las proporciones que se encontraban 483 imagenes de plantas enfermas y 162 imagenes de plantas sanas).

Entrenamiento con datos crudos

10 epocas

batch size = 15

```
loss: 0.2095 - accuracy: 0.9147 - val_loss: 0.2372 - val_accuracy: 0.9032
```

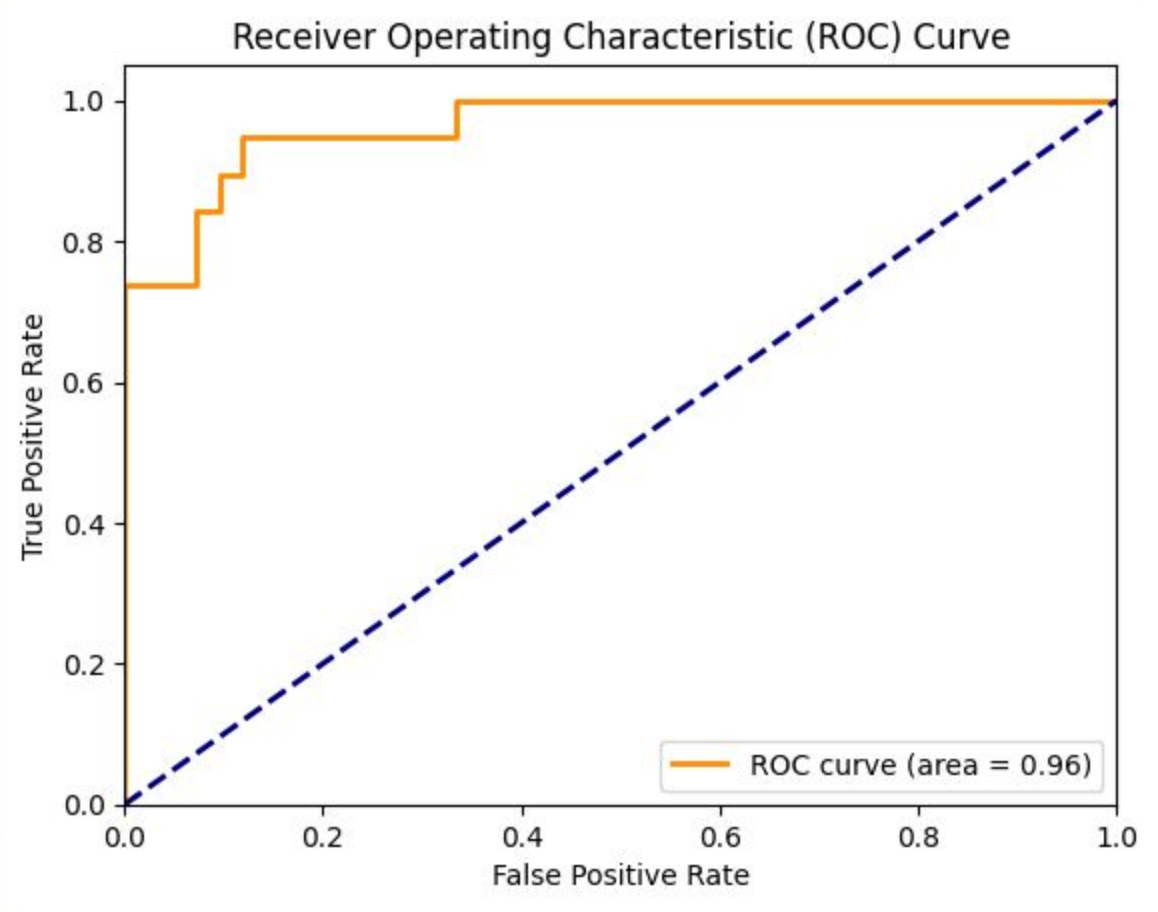


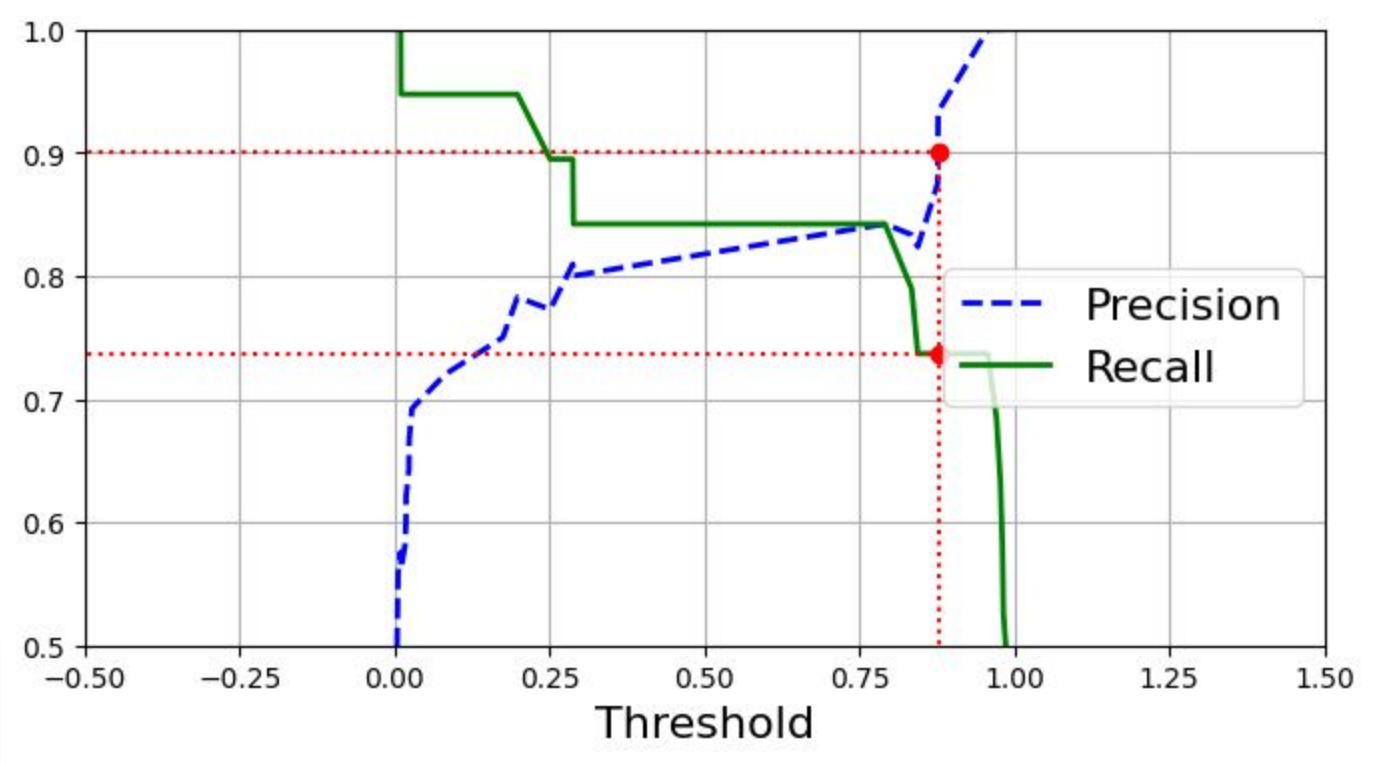
Accuracy: 0.9016

Recall: 0.8421

Precision: 0.8421

F1 Score: 0.8421


Receiver Operating Characteristic (ROC) Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (area = 0.96)


1.0
0.9
0.8
0.7
0.6
0.5
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
1.25
1.50
Threshold
Precision
Recall

04

Modelo pre-entrenado

Modelo de la CNN

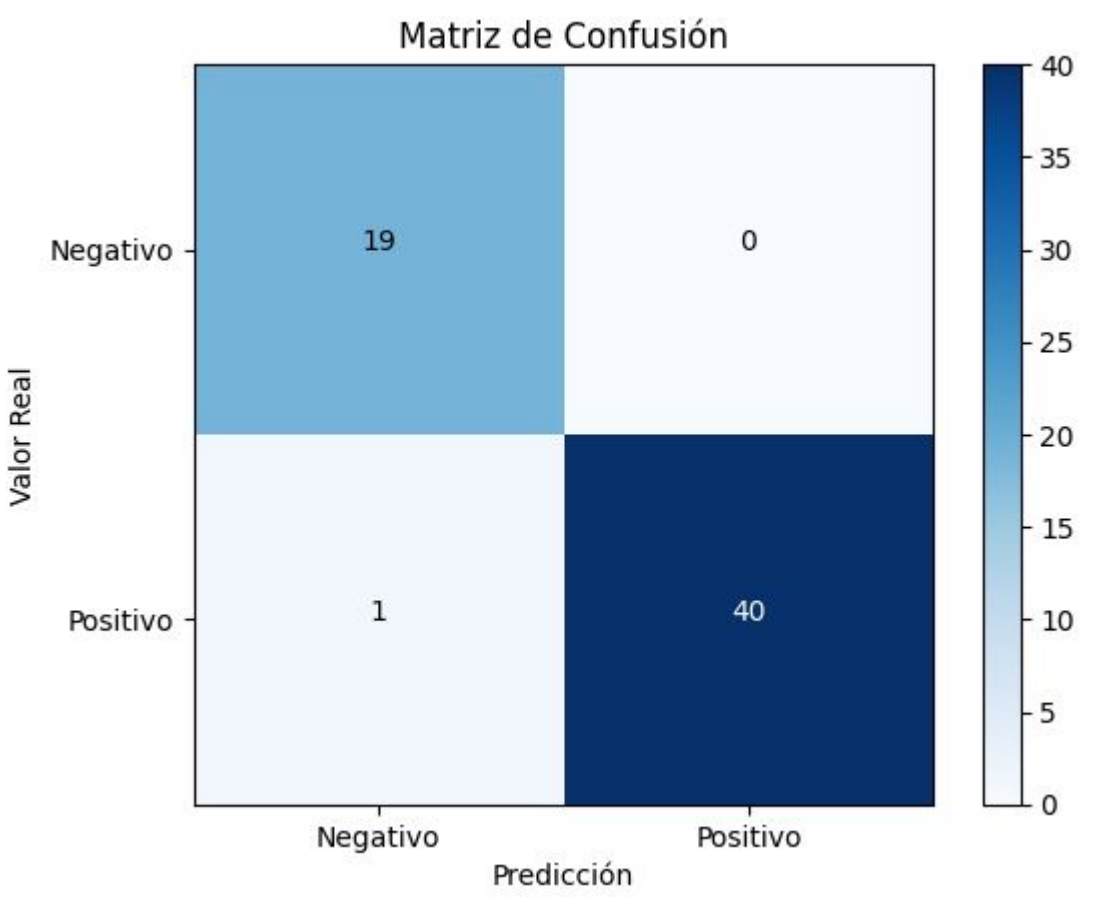
Para la creación de este modelo se utilizaron:

- Base Convolucional (VGG16)
- Extracción de características
- Aplanado de características
- Una capa densa con 256 unidades y activación ReLU.
- Una capa de dropout con una tasa del 50% para prevenir el sobreajuste.
- Una capa de salida con una única unidad y activación sigmoide para la clasificación binaria.

Entrenamiento de la CNN

- **Optimización**: optimizador RMSprop con una tasa de aprendizaje muy baja (2e-5).
- **Pérdida**: La función de pérdida es la entropía cruzada binaria.
- **Métricas**: Se utiliza la precisión (accuracy) como métrica para evaluar el rendimiento del modelo.
- El modelo se entrena durante 30 épocas con un tamaño de lote de 20.

```
loss: 0.0198 - acc: 0.9953 - val_loss: 0.0215 - val_acc: 0.9833
```

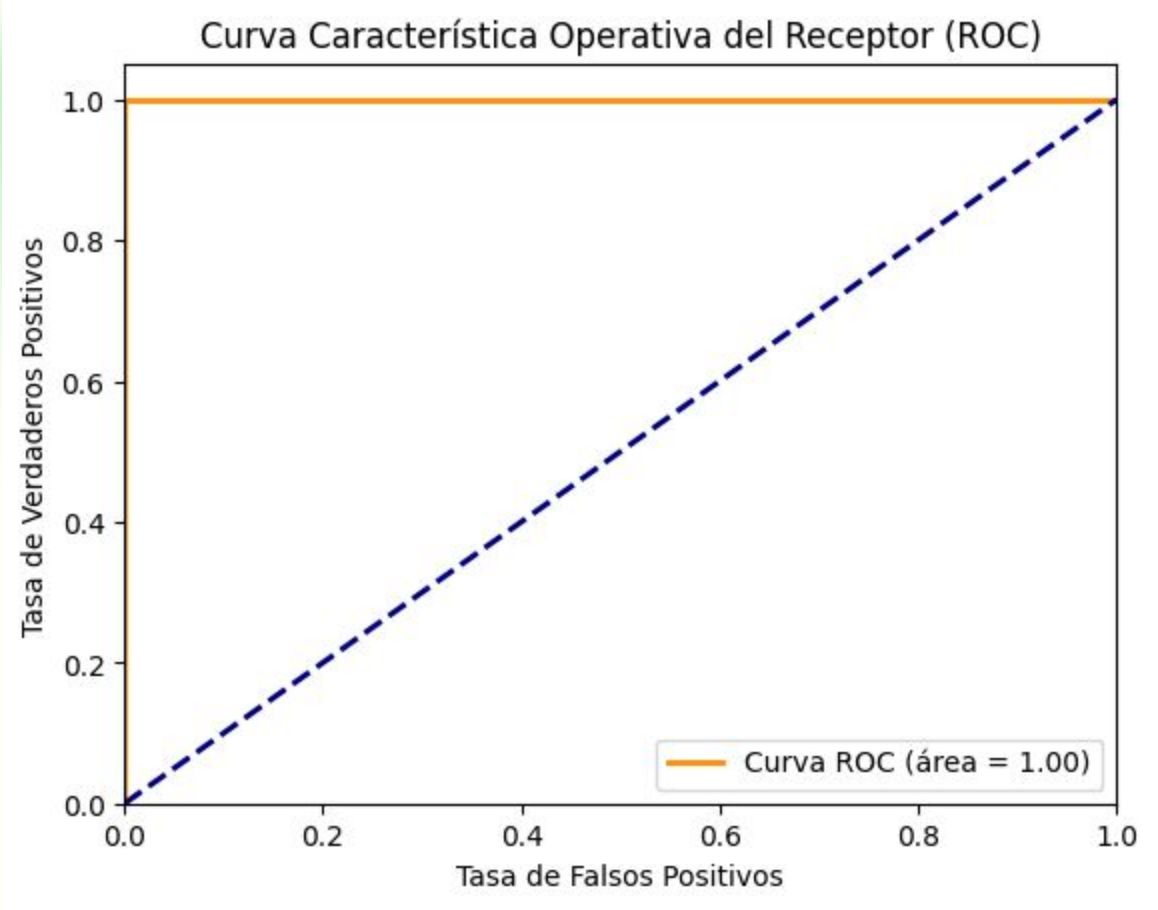


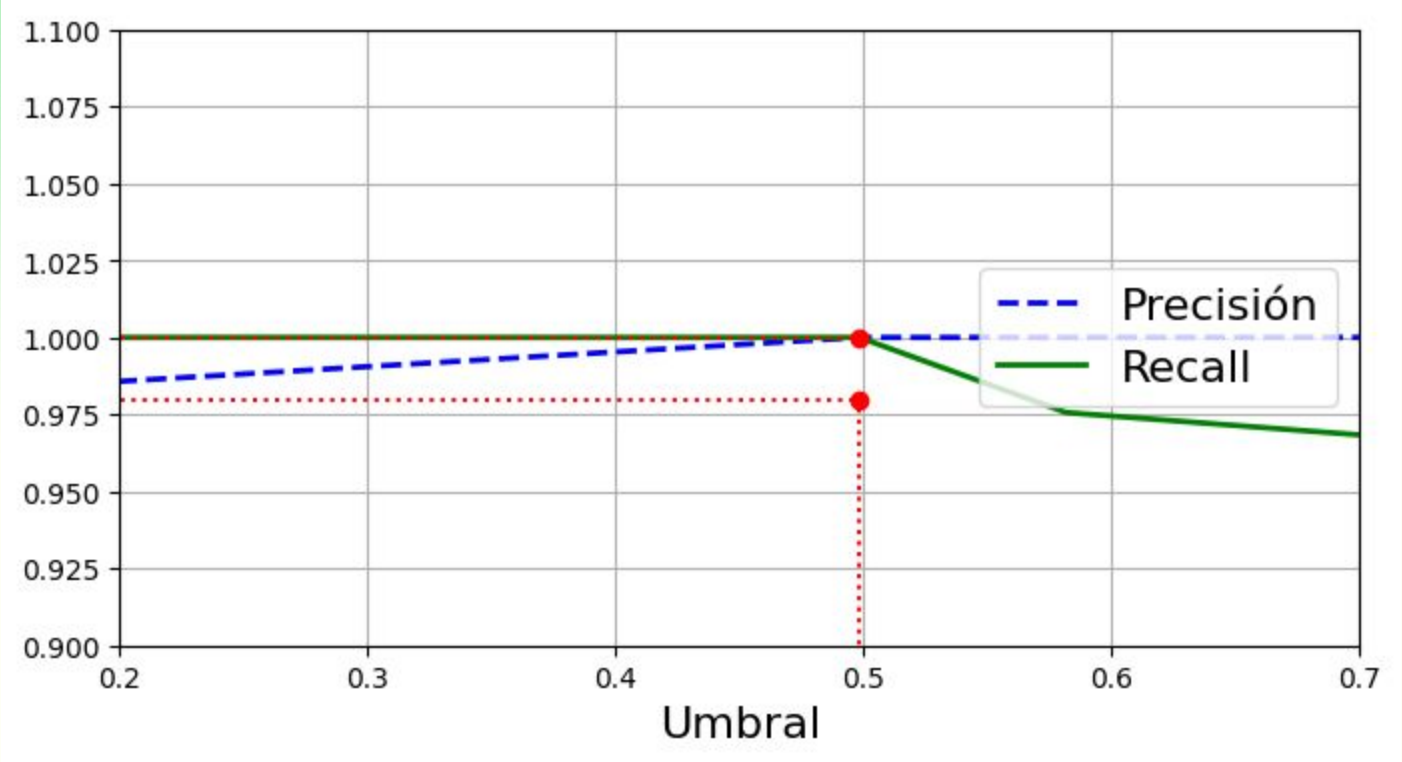
Accuracy: 0.9833

Recall: 0.95

Precision: 1

F1 Score: 0.9744


Curva Característica Operativa del Receptor (ROC)
Tasa de Verdaderos Positivos
Tasa de Falsos Positivos
0.0
0.2
0.4
0.6
0.8
1.0
Curva ROC (área = 1.00)


1.100
1.075
1.050
1.025
1.000
0.975
0.950
0.925
0.900
0.2
0.3
0.4
0.5
0.6
0.7
Umbral
Precisión
Recall

MODELO 2

¿Por qué otro modelo pre-entrenado?

VGG16 es un modelo pre-entrenado que al tratar de aplicarlo por primera vez, ocupaba mucho espacio en la memoria RAM.

Así que nos dimos en la tarea de encontrar una alternativa mucho más ligera. Encontramos EfficientNet-b0, que b0 es su versión más ligera. Es un red neuronal convolucional entrenada con datos de ImageNet y proviene de la librería Pytorch.
Una ventaja de trabajar con Pytorch es lo fácil que es indicar que el modelo trabaje con GPU para así acortar tiempos que con solo CPU serían muy tardados.

Modelo de la CNN

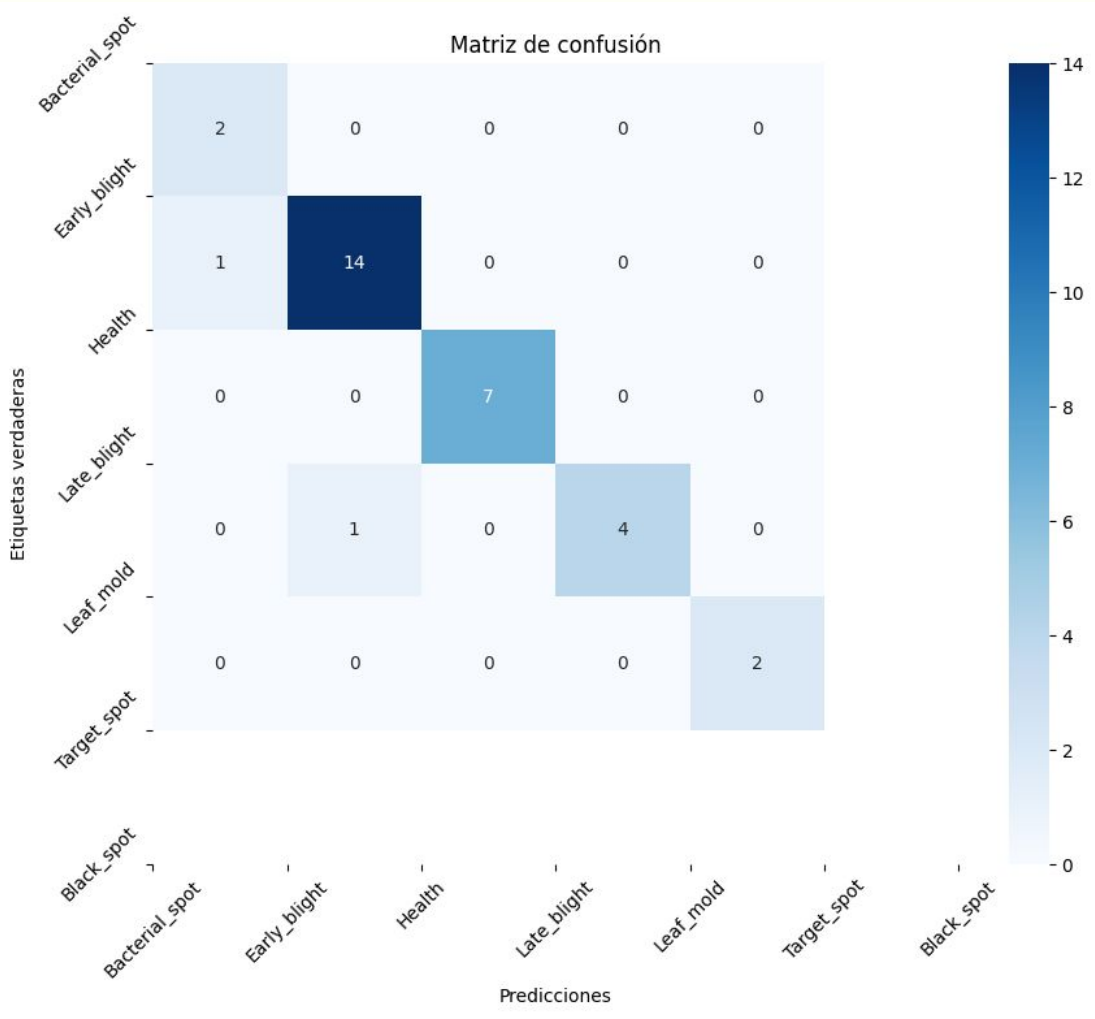
Para la creación de este modelo se utilizaron:

- Base Convolucional (EfficientNet-b0)
- Se usaron transformaciones en las imágenes de rotaciones verticales y horizontales. Cambios de saturación, brillo, contraste.
- Un realizó un resize a las imágenes de 256px
- Dropout de 20%
- Una capa densa con 512 unidades y activación ReLU
- Se modificó la última capa para tener 7 neuronas de salida

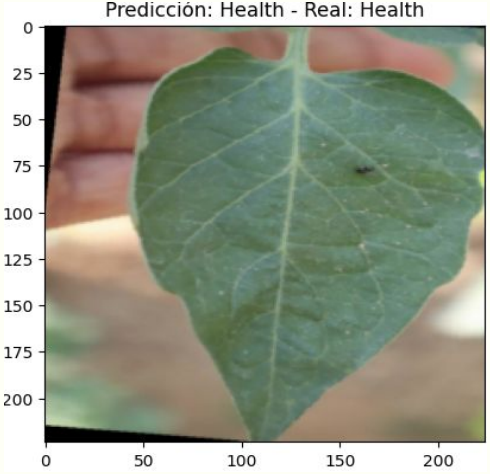
Entrenamiento de la CNN

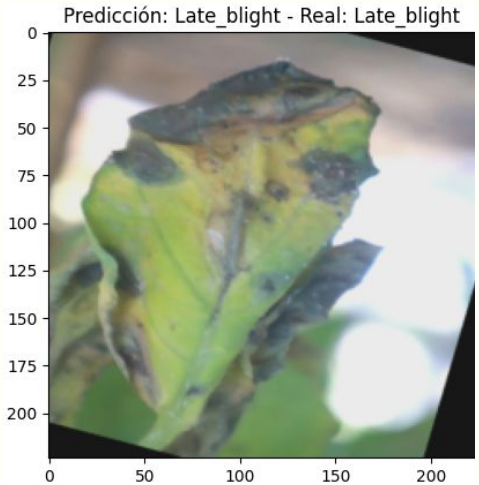
- **Optimización**: optimizador Adam con una tasa de aprendizaje de 0.001
- **Pérdida**: La función de pérdida es la entropía cruzada binaria.
- **Métricas**: Se utiliza la precisión (accuracy) como métrica para evaluar el rendimiento del modelo.
- El modelo se entrena durante 14 épocas

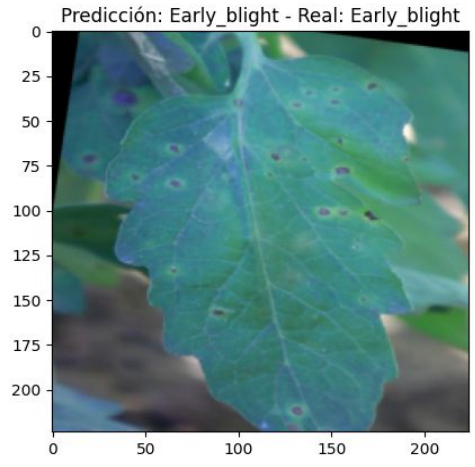
```
Epoch 1/14, Loss: 0.4112, Accuracy: 0.8410
Epoch 2/14, Loss: 0.2287, Accuracy: 0.9278
Epoch 3/14, Loss: 0.1804, Accuracy: 0.9610
Epoch 4/14, Loss: 0.1116, Accuracy: 0.9703
Epoch 5/14, Loss: 0.0976, Accuracy: 0.9734
Epoch 6/14, Loss: 0.1153, Accuracy: 0.9482
Epoch 7/14, Loss: 0.1213, Accuracy: 0.9663
Epoch 8/14, Loss: 0.1016, Accuracy: 0.9823
Epoch 9/14, Loss: 0.0705, Accuracy: 0.9781
Epoch 10/14, Loss: 0.0763, Accuracy: 0.9829
Epoch 11/14, Loss: 0.0575, Accuracy: 0.9845
Epoch 12/14, Loss: 0.0672, Accuracy: 0.9816
Epoch 13/14, Loss: 0.0659, Accuracy: 0.9779
Epoch 14/14, Loss: 0.0761, Accuracy: 0.9796
Entrenamiento completado
```

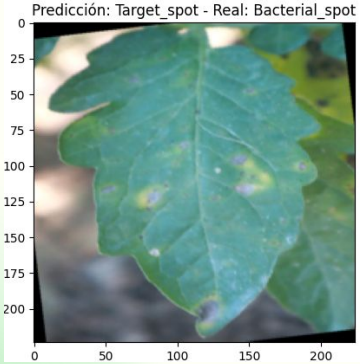

Matriz de confusión
Etiquetas verdaderas
Predicciones
Bacterial_spot
Early_blight
Health
Late_blight
Leaf_mold
Target_spot
Black_spot
2 0 0 0 0
1 14 0 0 0
0 0 7 0 0
0 1 0 4 0
0 0 0 0 2
14
12
10
8
6
4
2
0

Ejemplos Visuales



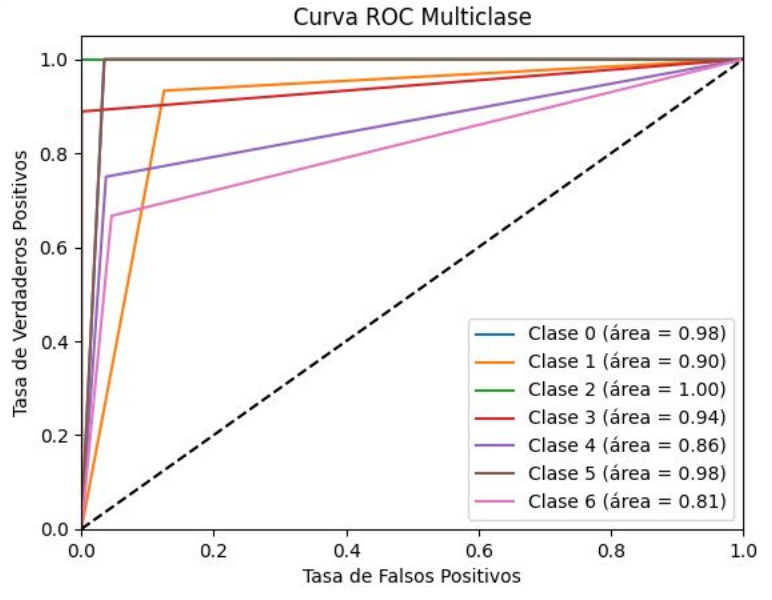


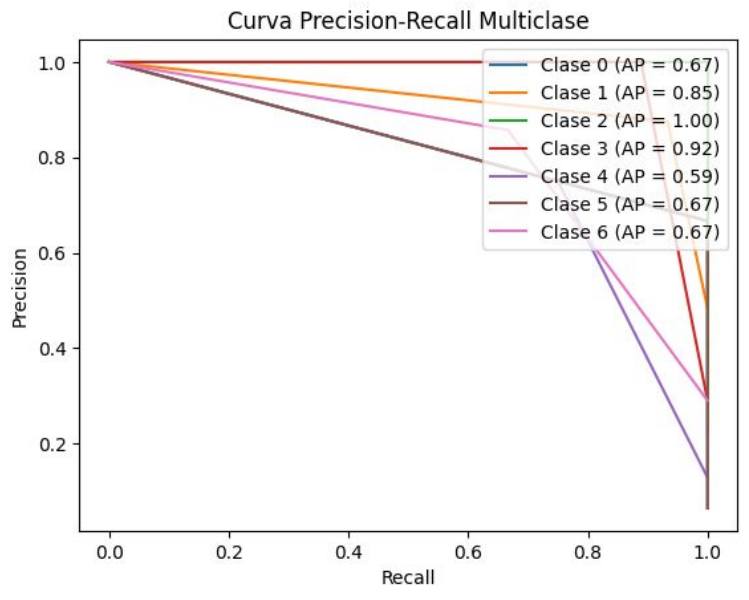




Ejemplo de predicción incorrecta

Es importante analizar visualmente las predicciones erróneas del modelo, en este caso confundió Target_spot (que se refiere a que la hoja tiene hongos), con bacterial_spot.


Curva ROC Multiclase
Tasa de Verdaderos Positivos
Tasa de Falsos Positivos
0.0
0.2
0.4
0.6
0.8
1.0
Clase 0 (área = 0.98)
Clase 1 (área = 0.90)
Clase 2 (área = 1.00)
Clase 3 (área = 0.94)
Clase 4 (área = 0.86)
Clase 5 (área = 0.98)
Clase 6 (área = 0.81)


Curva Precision-Recall Multiclase
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0
Clase 0 (AP = 0.67)
Clase 1 (AP = 0.85)
Clase 2 (AP = 1.00)
Clase 3 (AP = 0.92)
Clase 4 (AP = 0.59)
Clase 5 (AP = 0.67)
Clase 6 (AP = 0.67)

05

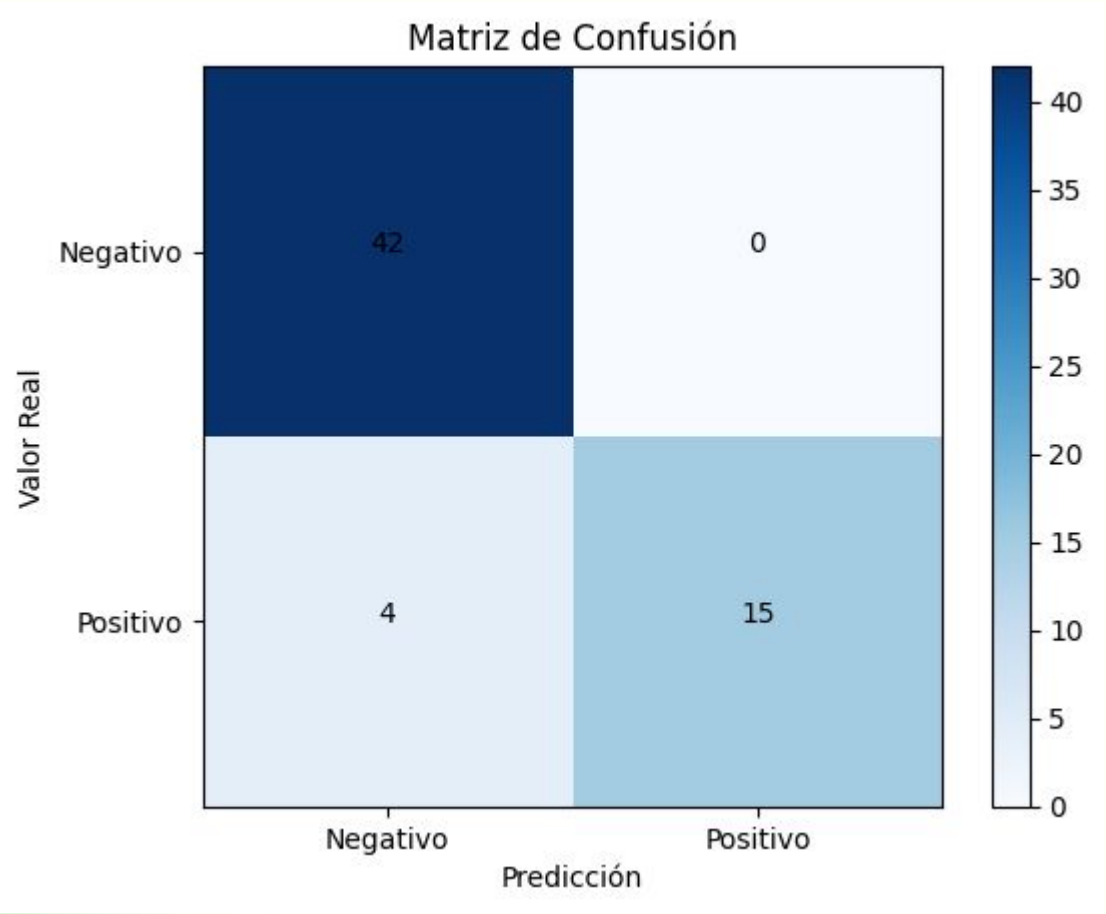
Data augmentation

Data augmentation

Como una manera de mejorar los resultados de nuestro modelo de CNN, se generaron nuevas imágenes para obtener un entrenamiento con clases balanceadas.

Se generaron 321 imágenes de hojas sanas lo que nos permitió tener 483 imágenes de hojas sanas y 483 imágenes de hojas con enfermedades.

Se generaron imágenes a partir de rotación usando ángulos de entre –10 y 10 grados.

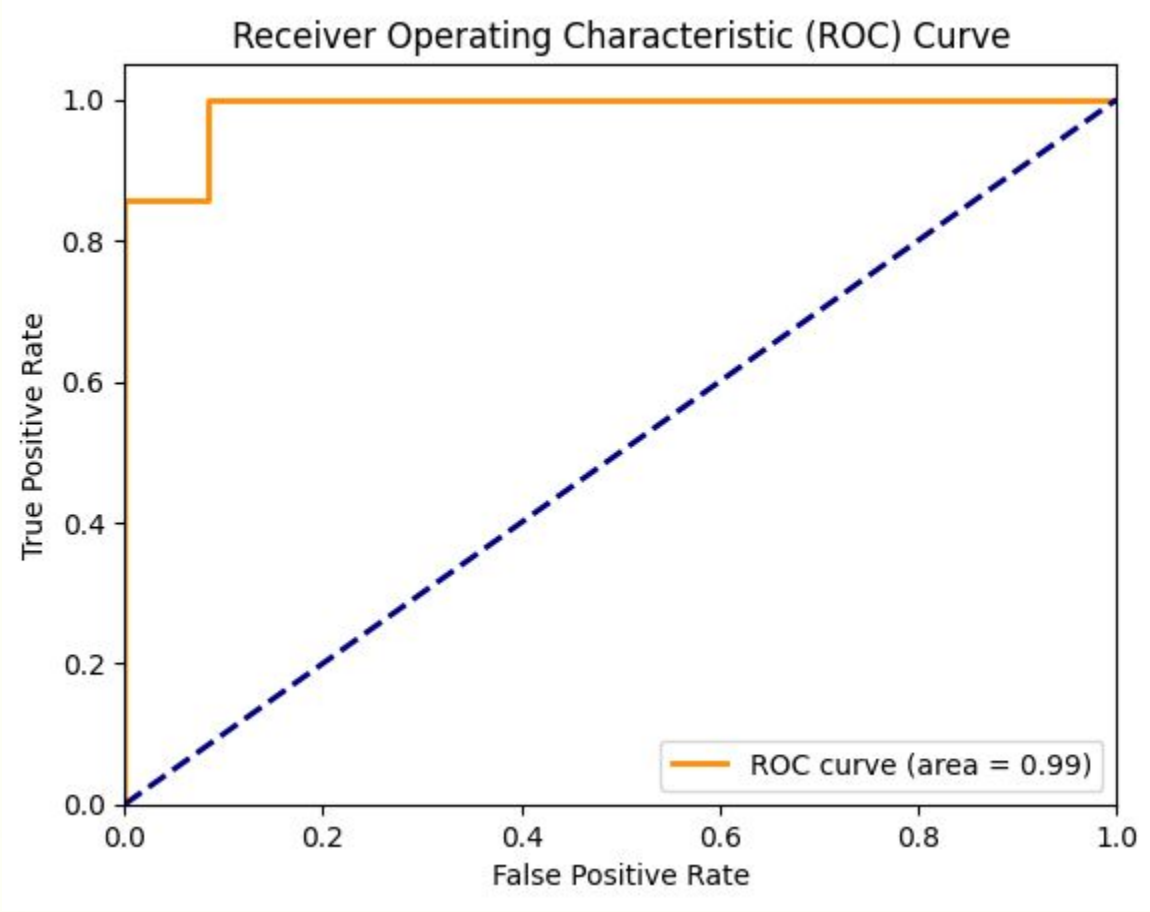


Accuracy: 0.9672

Recall: 0.8947

Precision: 1

F1 Score: 0.9444


Receiver Operating Characteristic (ROC) Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (area = 0.99)

Data augmentation

Evaluando los resultados de ambos casos, todas las métricas muestran una mejora significativa, con la precisión (precision) mostrando la mayor mejora porcentual (18.75%), seguida por el F1 Score (12.15%), el accuracy (7.28%) y el recall (6.25%).

06

Conclusiones