```
# Set seeds for reproducibility
import random
random.seed(0)

import numpy as np
np.random.seed(0)

import tensorflow as tf
tf.random.set_seed(0)
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
import os
import json
from zipfile import ZipFile
from PIL import Image

import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras import layers, models
```

```
!pip install kaggle
```

```
Requirement already satisfied: kaggle in /usr/local/lib/python3.10/dist-packages (1.5.16)
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.10/dist-packages (from kaggle) (1.16.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from kaggle) (2024.2.2)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.10/dist-packages (from kaggle) (2.8.2)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from kaggle) (2.31.0)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from kaggle) (4.66.2)
Requirement already satisfied: python-slugify in /usr/local/lib/python3.10/dist-packages (from kaggle) (8.0.4)
Requirement already satisfied: urllib3 in /usr/local/lib/python3.10/dist-packages (from kaggle) (2.0.7)
Requirement already satisfied: bleach in /usr/local/lib/python3.10/dist-packages (from kaggle) (6.1.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-packages (from bleach->kaggle) (0.5.1)
Requirement already satisfied: text-unidecode>=1.3 in /usr/local/lib/python3.10/dist-packages (from python-slugify->kaggle) (1.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->kaggle) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->kaggle) (3.6)
```

```
!ls -l kaggle.json
```

```
ls: cannot access 'kaggle.json': No such file or directory
```

```
!pip install kaggle
```

```
Requirement already satisfied: kaggle in /usr/local/lib/python3.10/dist-packages (1.5.16)
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.10/dist-packages (from kaggle) (1.16.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from kaggle) (2024.2.2)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.10/dist-packages (from kaggle) (2.8.2)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from kaggle) (2.31.0)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from kaggle) (4.66.2)
Requirement already satisfied: python-slugify in /usr/local/lib/python3.10/dist-packages (from kaggle) (8.0.4)
Requirement already satisfied: urllib3 in /usr/local/lib/python3.10/dist-packages (from kaggle) (2.0.7)
Requirement already satisfied: bleach in /usr/local/lib/python3.10/dist-packages (from kaggle) (6.1.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-packages (from bleach->kaggle) (0.5.1)
Requirement already satisfied: text-unidecode>=1.3 in /usr/local/lib/python3.10/dist-packages (from python-slugify->kaggle) (1.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->kaggle) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->kaggle) (3.6)
```

```
!ls ~/.kaggle
```

```
!kaggle config create
```

```
Traceback (most recent call last):
  File "/usr/local/bin/kaggle", line 5, in <module>
    from kaggle.cli import main
  File "/usr/local/lib/python3.10/dist-packages/kaggle/__init__.py", line 23, in <module>
    api.authenticate()
  File "/usr/local/lib/python3.10/dist-packages/kaggle/api/kaggle_api_extended.py", line 403, in authenticate
    raise IOError('Could not find {}. Make sure it\'s located in'
OSError: Could not find kaggle.json. Make sure it's located in /root/.kaggle. Or use the environment method.
```

```
kaggle_credentails = json.load(open("kaggle.json"))
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-7-8f659ac12673> in <cell line: 1>()
----> 1 kaggle_credentails = json.load(open("kaggle.json"))

FileNotFoundError: [Errno 2] No such file or directory: 'kaggle.json'
```

```
# setup Kaggle API key as environment variables
os.environ['KAGGLE_USERNAME'] = kaggle_credentails["username"]
os.environ['KAGGLE_KEY'] = kaggle_credentails["key"]
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-16-cb6f8e6b6342> in <cell line: 2>()
      1 # setup Kaggle API key as environment variables
----> 2 os.environ['KAGGLE_USERNAME'] = kaggle_credentails["username"]
      3 os.environ['KAGGLE_KEY'] = kaggle_credentails["key"]

NameError: name 'kaggle_credentails' is not defined
```

```
!kaggle datasets download -d abdallahalidev/plantvillage-dataset
```

```
Traceback (most recent call last):
  File "/usr/local/bin/kaggle", line 5, in <module>
    from kaggle.cli import main
  File "/usr/local/lib/python3.10/dist-packages/kaggle/__init__.py", line 23, in <module>
    api.authenticate()
  File "/usr/local/lib/python3.10/dist-packages/kaggle/api/kaggle_api_extended.py", line 403, in authenticate
    raise IOError('Could not find {}. Make sure it\'s located in'
OSError: Could not find kaggle.json. Make sure it's located in /root/.kaggle. Or use the environment method.
```

```
!ls
```

```
sample_data
```

```
# Unzip the downloaded dataset
with ZipFile("C:\Users\prave\Downloads.zip", 'r') as zip_ref:
    zip_ref.extractall()
```

```
  File "<ipython-input-22-a9ed489a9792>", line 2
    with ZipFile("C:\Users\prave\Downloads.zip", 'r') as zip_ref:
                                                ^
SyntaxError: (unicode error) 'unicodeescape' codec can't decode bytes in position 2-3: truncated \UXXXXXXXX escape
```

```
print(os.listdir("plantvillage dataset"))
```

```
print(len(os.listdir("plantvillage dataset/segmented")))
print(os.listdir("plantvillage dataset/segmented")[:5])

print(len(os.listdir("plantvillage dataset/color")))
print(os.listdir("plantvillage dataset/color")[:5])

print(len(os.listdir("plantvillage dataset/grayscale")))
print(os.listdir("plantvillage dataset/grayscale")[:5])
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-23-64d83234f816> in <cell line: 1>()
----> 1 print(os.listdir("plantvillage dataset"))
      2
      3
      4 print(len(os.listdir("plantvillage dataset/segmented")))
      5 print(os.listdir("plantvillage dataset/segmented")[:5])

FileNotFoundError: [Errno 2] No such file or directory: 'plantvillage dataset'
```

```
print(len(os.listdir("plantvillage dataset/color/Grape___healthy")))
print(os.listdir("plantvillage dataset/color/Grape___healthy")[:5])
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-24-09dce15089cc> in <cell line: 1>()
----> 1 print(len(os.listdir("plantvillage dataset/color/Grape___healthy")))
      2 print(os.listdir("plantvillage dataset/color/Grape___healthy")[:5])
      3 

FileNotFoundError: [Errno 2] No such file or directory: 'plantvillage dataset/color/Grape___healthy'
```

```
# Dataset Path
base_dir = 'plantvillage dataset/color'
```

```
image_path = '/content/plantvillage dataset/color/Apple___Cedar_apple_rust/025b2b9a-0ec4-4132-96ac-7f2832d0db4a___FREC_C.Rust 3655.JPG'

# Read the image
img = mpimg.imread(image_path)

print(img.shape)
# Display the image
plt.imshow(img)
plt.axis('off')  # Turn off axis numbers
plt.show()
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-26-51252b96d046> in <cell line: 4>()
      2 
      3 # Read the image
----> 4 img = mpimg.imread(image_path)
      5 
      6 print(img.shape)

 ──────────────────────  1 frames  ──────────────────────
/usr/local/lib/python3.10/dist-packages/PIL/Image.py in open(fp, mode, formats)
   3225 
   3226     if filename:
-> 3227         fp = builtins.open(filename, "rb")
   3228         exclusive_fp = True
   3229 

FileNotFoundError: [Errno 2] No such file or directory: '/content/plantvillage dataset/color/Apple___Cedar_apple_rust/025b2b9a-
0ec4-4132-96ac-7f2832d0db4a___FREC_C.Rust 3655.JPG'
```

```
image_path = '/content/plantvillage dataset/color/Apple___Cedar_apple_rust/025b2b9a-0ec4-4132-96ac-7f2832d0db4a___FREC_C.Rust 3655.JPG'

# Read the image
img = mpimg.imread(image_path)

print(img)
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-27-75985c3f1ba8> in <cell line: 4>()
      2 
      3 # Read the image
----> 4 img = mpimg.imread(image_path)
      5 
      6 print(img)

 ──────────────────────  1 frames  ──────────────────────
/usr/local/lib/python3.10/dist-packages/PIL/Image.py in open(fp, mode, formats)
   3225 
   3226     if filename:
-> 3227         fp = builtins.open(filename, "rb")
   3228         exclusive_fp = True
   3229 

FileNotFoundError: [Errno 2] No such file or directory: '/content/plantvillage dataset/color/Apple___Cedar_apple_rust/025b2b9a-
0ec4-4132-96ac-7f2832d0db4a___FREC_C.Rust 3655.JPG'
```

```
# Image Parameters
img_size = 224
batch_size = 32
```

```
# Image Data Generators
data_gen = ImageDataGenerator(
    rescale=1./255,
    validation_split=0.2  # Use 20% of data for validation
)
```

```
# Train Generator
train_generator = data_gen.flow_from_directory(
    base_dir,
    target_size=(img_size, img_size),
    batch_size=batch_size,
    subset='training',
    class_mode='categorical'
)
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-30-c7eee91d6a01> in <cell line: 2>()
      1 # Train Generator
----> 2 train_generator = data_gen.flow_from_directory(
      3     base_dir,
      4     target_size=(img_size, img_size),
      5     batch_size=batch_size,

                                  1 frames

/usr/local/lib/python3.10/dist-packages/keras/src/preprocessing/image.py in __init__(self, directory, image_data_generator,
target_size, color_mode, classes, class_mode, batch_size, shuffle, seed, data_format, save_to_dir, save_prefix, save_format,
follow_links, subset, interpolation, keep_aspect_ratio, dtype)
    561         if not classes:
    562             classes = []
--> 563             for subdir in sorted(os.listdir(directory)):
    564                 if os.path.isdir(os.path.join(directory, subdir)):
    565                     classes.append(subdir)

FileNotFoundError: [Errno 2] No such file or directory: 'plantvillage dataset/color'
```

```
# Validation Generator
validation_generator = data_gen.flow_from_directory(
    base_dir,
    target_size=(img_size, img_size),
    batch_size=batch_size,
    subset='validation',
    class_mode='categorical'
)
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-31-072c50c1608b> in <cell line: 2>()
      1 # Validation Generator
----> 2 validation_generator = data_gen.flow_from_directory(
      3     base_dir,
      4     target_size=(img_size, img_size),
      5     batch_size=batch_size,

                                  1 frames

/usr/local/lib/python3.10/dist-packages/keras/src/preprocessing/image.py in __init__(self, directory, image_data_generator,
target_size, color_mode, classes, class_mode, batch_size, shuffle, seed, data_format, save_to_dir, save_prefix, save_format,
follow_links, subset, interpolation, keep_aspect_ratio, dtype)
    561         if not classes:
    562             classes = []
--> 563             for subdir in sorted(os.listdir(directory)):
    564                 if os.path.isdir(os.path.join(directory, subdir)):
    565                     classes.append(subdir)

FileNotFoundError: [Errno 2] No such file or directory: 'plantvillage dataset/color'
```

```
# Model Definition
model = models.Sequential()

model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(img_size, img_size, 3)))
model.add(layers.MaxPooling2D(2, 2))

model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D(2, 2))


model.add(layers.Flatten())
model.add(layers.Dense(256, activation='relu'))
model.add(layers.Dense(train_generator.num_classes, activation='softmax'))
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-32-1fd37a0ee242> in <cell line: 13>()
     11 model.add(layers.Flatten())
     12 model.add(layers.Dense(256, activation='relu'))
---> 13 model.add(layers.Dense(train_generator.num_classes, activation='softmax'))

NameError: name 'train_generator' is not defined
```

```
# model summary
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d (Conv2D)             (None, 222, 222, 32)      896       
                                                                 
 max_pooling2d (MaxPooling2  (None, 111, 111, 32)      0         
 D)                                                              
                                                                 
 conv2d_1 (Conv2D)           (None, 109, 109, 64)      18496     
                                                                 
 max_pooling2d_1 (MaxPoolin  (None, 54, 54, 64)        0         
 g2D)                                                            
                                                                 
 flatten (Flatten)           (None, 186624)            0         
                                                                 
 dense (Dense)               (None, 256)               47776000  
                                                                 
=================================================================
Total params: 47795392 (182.32 MB)
Trainable params: 47795392 (182.32 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

```
# Compile the Model
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

```
# Training the Model
history = model.fit(
    train_generator,
    steps_per_epoch=train_generator.samples // batch_size,  # Number of steps per epoch
    epochs=5,  # Number of epochs
    validation_data=validation_generator,
    validation_steps=validation_generator.samples // batch_size  # Validation steps
)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-35-1e78b9779117> in <cell line: 2>()
      1 # Training the Model
      2 history = model.fit(
----> 3     train_generator,
      4     steps_per_epoch=train_generator.samples // batch_size,  # Number of steps per epoch
      5     epochs=5,  # Number of epochs

NameError: name 'train_generator' is not defined
```

```
# Model Evaluation
print("Evaluating model...")
val_loss, val_accuracy = model.evaluate(validation_generator, steps=validation_generator.samples // batch_size)
print(f"Validation Accuracy: {val_accuracy * 100:.2f}%")
```

```
Evaluating model...
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-36-63593e401f84> in <cell line: 3>()
      1 # Model Evaluation
      2 print("Evaluating model...")
----> 3 val_loss, val_accuracy = model.evaluate(validation_generator, steps=validation_generator.samples // batch_size)
      4 print(f"Validation Accuracy: {val_accuracy * 100:.2f}%")

NameError: name 'validation_generator' is not defined
```

```
# Plot training & validation accuracy values
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()

# Plot training & validation loss values
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-37-3ffca6ab203e> in <cell line: 2>()
      1 # Plot training & validation accuracy values
----> 2 plt.plot(history.history['accuracy'])
      3 plt.plot(history.history['val_accuracy'])
      4 plt.title('Model accuracy')
      5 plt.ylabel('Accuracy')

NameError: name 'history' is not defined
```

```
# Function to Load and Preprocess the Image using Pillow
def load_and_preprocess_image(image_path, target_size=(224, 224)):
    # Load the image
    img = Image.open(image_path)
    # Resize the image
    img = img.resize(target_size)
    # Convert the image to a numpy array
    img_array = np.array(img)
    # Add batch dimension
    img_array = np.expand_dims(img_array, axis=0)
    # Scale the image values to [0, 1]
    img_array = img_array.astype('float32') / 255.
    return img_array

# Function to Predict the Class of an Image
def predict_image_class(model, image_path, class_indices):
    preprocessed_img = load_and_preprocess_image(image_path)
    predictions = model.predict(preprocessed_img)
    predicted_class_index = np.argmax(predictions, axis=1)[0]
    predicted_class_name = class_indices[predicted_class_index]
    return predicted_class_name


# Create a mapping from class indices to class names
class_indices = {v: k for k, v in train_generator.class_indices.items()}
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-39-2dc2b5043d34> in <cell line: 2>()
      1 # Create a mapping from class indices to class names
----> 2 class_indices = {v: k for k, v in train_generator.class_indices.items()}
      3

NameError: name 'train_generator' is not defined
```

```
class_indices
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-40-1182e89b957a> in <cell line: 1>()
----> 1 class_indices
      2

NameError: name 'class_indices' is not defined
```

```
# saving the class names as json file
json.dump(class_indices, open('class_indices.json', 'w'))
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-41-edc2dd7ca138> in <cell line: 2>()
      1 # saving the class names as json file
----> 2 json.dump(class_indices, open('class_indices.json', 'w'))

NameError: name 'class_indices' is not defined
```

```
# Example Usage
image_path = '/content/test_apple_black_rot.JPG'
#image_path = '/content/test_blueberry_healthy.jpg'
#image_path = '/content/test_potato_early_blight.jpg'
predicted_class_name = predict_image_class(model, image_path, class_indices)

# Output the result
print("Predicted Class Name:", predicted_class_name)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-42-25f7ee1fac37> in <cell line: 5>()
      3 #image_path = '/content/test_blueberry_healthy.jpg'
      4 #image_path = '/content/test_potato_early_blight.jpg'
----> 5 predicted_class_name = predict_image_class(model, image_path, class_indices)
      6 
      7 # Output the result

NameError: name 'class_indices' is not defined
```

```
model.save('drive/MyDrive/Youtube/trained_models/plant_disease_prediction_model.h5')
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103: UserWarning: You are saving your model as an HDF5 file v
  saving_api.save_model(
```

```
model.save('plant_disease_prediction_model.h5')
```