```
Python 3.9.16 (main, Mar  1 2023, 18:30:21) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.10.0 -- An enhanced Interactive Python.

In [1]: import os, glob
   ...: import numpy as np
   ...: import seaborn as sns
   ...: import matplotlib.pyplot as plt
   ...: import pandas as pd
   ...: import tensorflow as tf
   ...: from sklearn.model_selection import train_test_split
   ...: from tensorflow.keras.preprocessing.image import ImageDataGenerator
   ...: from tensorflow.keras.models import Model
   ...: from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
   ...: from tensorflow.keras.callbacks import Callback,EarlyStopping
   ...: from tensorflow.keras.applications import VGG19
   ...: from tensorflow.keras.applications.vgg19 import preprocess_input
   ...: from sklearn.metrics import  classification_report
```

```
In [2]: file_path = r"C:\Users\Admin\Desktop\Tomatod"
   ...: name_class = os.listdir(file_path)
   ...: print(name_class)
['Tomato_Bacterial_spot', 'Tomato_Late_blight', 'Tomato_Septoria_leaf_spot',
'Tomato__Tomato_YellowLeaf__Curl_Virus']

In [3]: filepaths = list(glob.glob(file_path+'/**/*.*'))
   ...:
   ...: labels = list(map(lambda x: os.path.split(os.path.split(x)[0])[1], filepaths))
   ...: filepath = pd.Series(filepaths, name='Filepath').astype(str)
   ...: labels = pd.Series(labels, name='Label')
   ...: data = pd.concat([filepath, labels], axis=1)
   ...: data = data.sample(frac=1).reset_index(drop=True)
   ...: data.head()
Out[3]:
                                            Filepath                                      Label
0  C:\Users\Admin\Desktop\Tomatod\Tomato__Tomato_...  Tomato__Tomato_YellowLeaf__Curl_Virus
1  C:\Users\Admin\Desktop\Tomatod\Tomato_Bacteria...                 Tomato_Bacterial_spot
2  C:\Users\Admin\Desktop\Tomatod\Tomato_Late_bli...                   Tomato_Late_blight
3  C:\Users\Admin\Desktop\Tomatod\Tomato__Tomato_...  Tomato__Tomato_YellowLeaf__Curl_Virus
4  C:\Users\Admin\Desktop\Tomatod\Tomato_Septoria...               Tomato_Septoria_leaf_spot
```

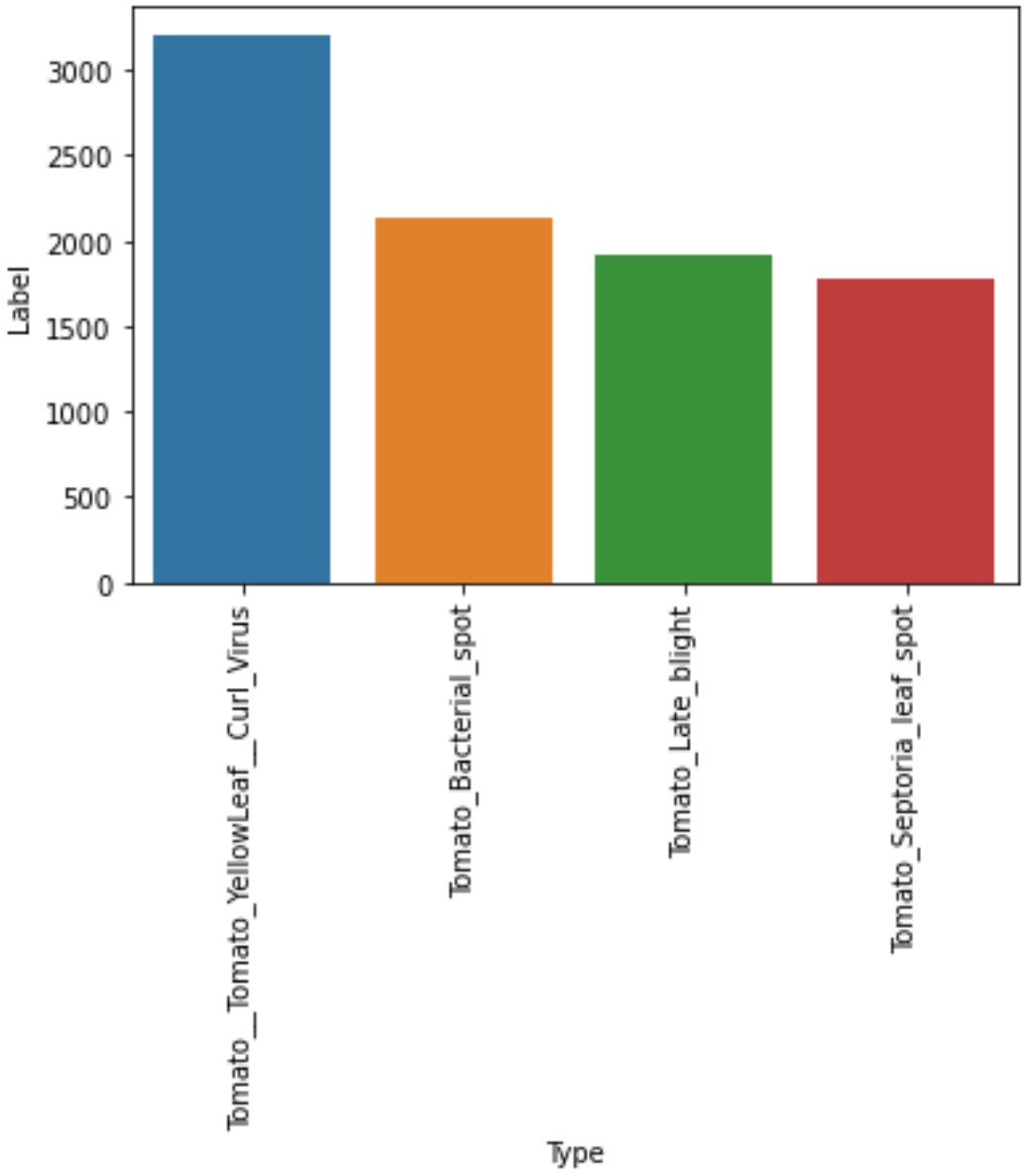
```
In [4]: data.describe
Out[4]:
<bound method NDFrame.describe of                                              Filepath
Label
0     C:\Users\Admin\Desktop\Tomatod\Tomato__Tomato_...  Tomato__Tomato_YellowLeaf__Curl_Virus
1     C:\Users\Admin\Desktop\Tomatod\Tomato_Bacteria...                 Tomato_Bacterial_spot
2     C:\Users\Admin\Desktop\Tomatod\Tomato_Late_bli...                   Tomato_Late_blight
3     C:\Users\Admin\Desktop\Tomatod\Tomato__Tomato_...  Tomato__Tomato_YellowLeaf__Curl_Virus
4     C:\Users\Admin\Desktop\Tomatod\Tomato_Septoria...               Tomato_Septoria_leaf_spot
...                                                 ...                                     ...
9010  C:\Users\Admin\Desktop\Tomatod\Tomato_Bacteria...                 Tomato_Bacterial_spot
9011  C:\Users\Admin\Desktop\Tomatod\Tomato__Tomato_...  Tomato__Tomato_YellowLeaf__Curl_Virus
9012  C:\Users\Admin\Desktop\Tomatod\Tomato__Tomato_...  Tomato__Tomato_YellowLeaf__Curl_Virus
9013  C:\Users\Admin\Desktop\Tomatod\Tomato_Late_bli...                   Tomato_Late_blight
9014  C:\Users\Admin\Desktop\Tomatod\Tomato_Late_bli...                   Tomato_Late_blight
```

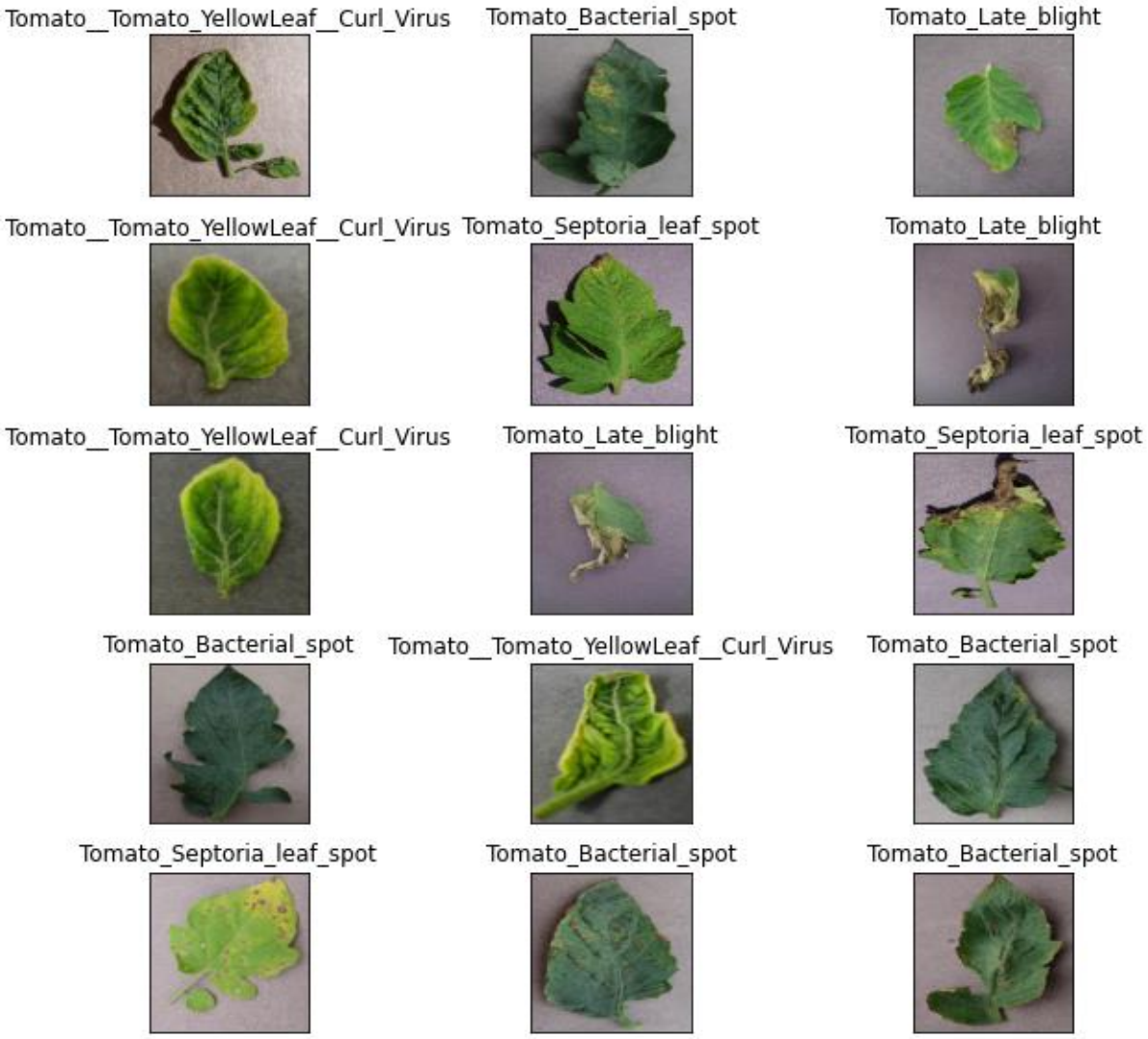
```
[9015 rows x 2 columns]>

In [5]: counts = data.Label.value_counts()
   ...: sns.barplot(x=counts.index, y=counts)
   ...: plt.xlabel('Type')
   ...: plt.xticks(rotation=90);
```

Warning

Figures now render in the Plots pane by default. To make them also appear inline in the Console, uncheck "Mute Inline Plotting" under the Plots pane options menu.

```
In [6]: train, test = train_test_split(data, test_size=0.25, random_state=42)
   ...:
   ...: fig, axes = plt.subplots(nrows=5, ncols=3, figsize=(10,8), subplot_kw={'xticks':
[],'yticks':[]})
   ...: for i, ax in enumerate(axes.flat):
   ...:     ax.imshow(plt.imread(data.Filepath[i]))
   ...:     ax.set_title(data.Label[i])
   ...: plt.tight_layout()
   ...: plt.show()
```


3000
2500
2000
1500
1000
500
0
Label
Tomato__Tomato_YellowLeaf__Curl_Virus
Tomato_Bacterial_spot
Tomato_Late_blight
Tomato_Septoria_leaf_spot
Type



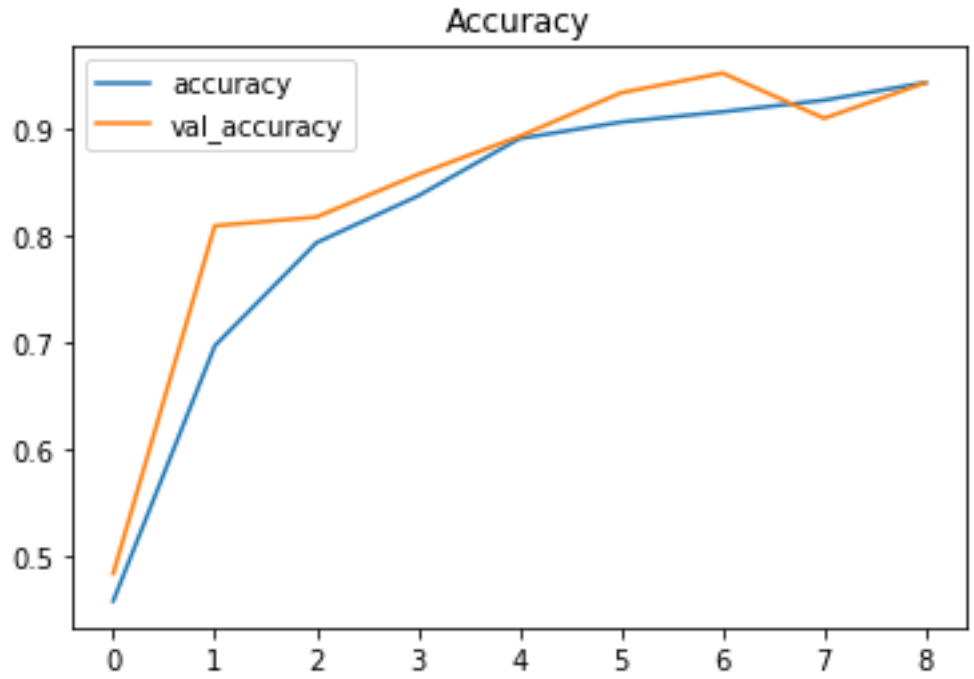

```
In [7]: train_datagen = ImageDataGenerator(preprocessing_function=preprocess_input,
   ...:                                    validation_split=0.2)
   ...: test_datagen = ImageDataGenerator(preprocessing_function=preprocess_input)
```

```
In [8]: train_gen = train_datagen.flow_from_dataframe(
   ...:     dataframe=train,
   ...:     x_col='Filepath',
   ...:     y_col='Label',
   ...:     target_size=(100,100),
   ...:     class_mode='categorical',
   ...:     batch_size=32,
   ...:     shuffle=True,
   ...:     seed=42
   ...: )
   ...: valid_gen = train_datagen.flow_from_dataframe(
   ...:     dataframe=train,
   ...:     x_col='Filepath',
   ...:     y_col='Label',
   ...:     target_size=(100,100),
   ...:     class_mode='categorical',
   ...:     batch_size=32,
   ...:     shuffle=False,
   ...:     seed=42
   ...: )
   ...: test_gen = test_datagen.flow_from_dataframe(
   ...:     dataframe=test,
   ...:     x_col='Filepath',
   ...:     y_col='Label',
   ...:     target_size=(100,100),
   ...:     class_mode='categorical',
   ...:     batch_size=32,
   ...:     shuffle=False
   ...: )
   ...:
   ...: pretrained_model = VGG19(
   ...:     input_shape=(100,100, 3),
   ...:     include_top=False,
   ...:     weights='imagenet',
   ...:     pooling='avg'
   ...: )
```

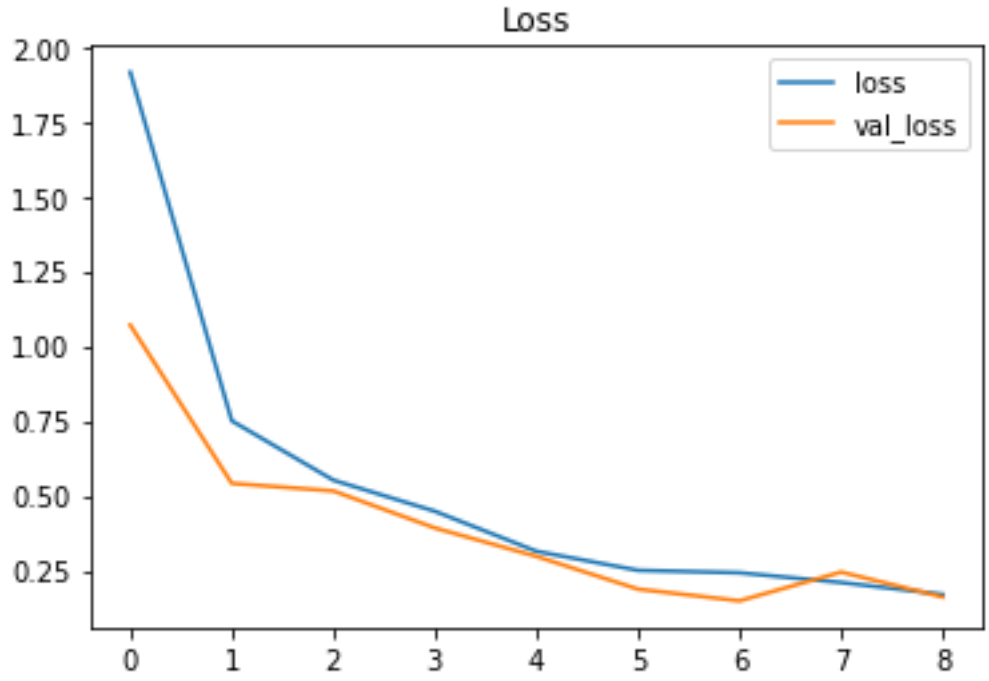
```
Found 6761 validated image filenames belonging to 4 classes.
Found 6761 validated image filenames belonging to 4 classes.
Found 2254 validated image filenames belonging to 4 classes.

2023-03-17 17:38:28.564133: I tensorflow/core/platform/cpu_feature_guard.cc:193] This
TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the
following CPU instructions in performance-critical operations:  AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.

In [9]: inputs = pretrained_model.input
   ...:
   ...: x = Dense(128, activation='relu')(pretrained_model.output)
   ...: x = Dense(128, activation='relu')(x)
   ...:
   ...: outputs = Dense(4, activation='softmax')(x)
   ...:
   ...: model = Model(inputs=inputs, outputs=outputs)
   ...:
   ...: model.compile(
   ...:     optimizer='adam',
   ...:     loss='categorical_crossentropy',
   ...:     metrics=['accuracy']
   ...: )
   ...:
   ...: my_callbacks  = [EarlyStopping(monitor='val_accuracy',
   ...:                                min_delta=0,
   ...:                                patience=2,
   ...:                                mode='auto')]
```

```
In [10]: history = model.fit(
    ...:     train_gen,
    ...:     validation_data=valid_gen,
    ...:     epochs=10,
    ...:     callbacks=my_callbacks
    ...: )
Epoch 1/10
212/212 [==============================] - 1433s 7s/step - loss: 1.9148 - accuracy: 0.4585 -
val_loss: 1.0700 - val_accuracy: 0.4841
Epoch 2/10
212/212 [==============================] - 1420s 7s/step - loss: 0.7498 - accuracy: 0.6965 -
val_loss: 0.5408 - val_accuracy: 0.8085
Epoch 3/10
212/212 [==============================] - 1236s 6s/step - loss: 0.5515 - accuracy: 0.7926 -
val_loss: 0.5155 - val_accuracy: 0.8166
Epoch 4/10
212/212 [==============================] - 1202s 6s/step - loss: 0.4476 - accuracy: 0.8363 -
val_loss: 0.3917 - val_accuracy: 0.8564
Epoch 5/10
212/212 [==============================] - 1188s 6s/step - loss: 0.3138 - accuracy: 0.8898 -
val_loss: 0.2968 - val_accuracy: 0.8919
Epoch 6/10
212/212 [==============================] - 1197s 6s/step - loss: 0.2502 - accuracy: 0.9052 -
val_loss: 0.1873 - val_accuracy: 0.9326
Epoch 7/10
212/212 [==============================] - 1208s 6s/step - loss: 0.2420 - accuracy: 0.9148 -
val_loss: 0.1482 - val_accuracy: 0.9509
Epoch 8/10
212/212 [==============================] - 1198s 6s/step - loss: 0.2095 - accuracy: 0.9256 -
val_loss: 0.2444 - val_accuracy: 0.9087
Epoch 9/10
212/212 [==============================] - 1211s 6s/step - loss: 0.1697 - accuracy: 0.9420 -
val_loss: 0.1610 - val_accuracy: 0.9420
```

```
In [11]: pd.DataFrame(history.history)[['accuracy','val_accuracy']].plot()
    ...: plt.title("Accuracy")
    ...: plt.show()
```




```
In [12]: pd.DataFrame(history.history)[['loss','val_loss']].plot()
    ...: plt.title("Loss")
    ...: plt.show()
```



```
In [13]: results = model.evaluate(test_gen, verbose=0)
    ...:
    ...: print("    Test Loss: {:.5f}".format(results[0]))
    ...: print("Test Accuracy: {:.2f}%".format(results[1] * 100))
    Test Loss: 0.23190
Test Accuracy: 92.10%
```

```
In [15]:
    ...: pred = model.predict(test_gen)
    ...: pred = np.argmax(pred,axis=1)
71/71 [==============================] - 72s 1s/step

In [16]:
    ...: labels = (train_gen.class_indices)
    ...: labels = dict((v,k) for k,v in labels.items())
    ...: pred = [labels[k] for k in pred]
```

```
In [17]: y_test = list(test.Label)
    ...: print(classification_report(y_test, pred))
                                    precision    recall  f1-score   support

               Tomato_Bacterial_spot       0.83      1.00      0.90       515
                  Tomato_Late_blight       0.90      0.93      0.92       496
            Tomato_Septoria_leaf_spot       0.98      0.84      0.90       438
Tomato__Tomato_YellowLeaf__Curl_Virus       0.99      0.91      0.95       805

                            accuracy                           0.92      2254
                           macro avg       0.92      0.92      0.92      2254
                        weighted avg       0.93      0.92      0.92      2254
```

```
In [18]: fig, axes = plt.subplots(nrows=5, ncols=2, figsize=(12, 8),
    ...:                           subplot_kw={'xticks': [], 'yticks': []})
    ...:
    ...: for i, ax in enumerate(axes.flat):
    ...:     ax.imshow(plt.imread(test.Filepath.iloc[i]))
    ...:     ax.set_title(f"True: {test.Label.iloc[i]}\nPredicted: {pred[i]}")
    ...: plt.tight_layout()
    ...: plt.show()
```

True outputs vs Predicted outputs:

True: Tomato__Tomato_YellowLeaf__Curl_Virus
Predicted: Tomato__Tomato_YellowLeaf__Curl_Virus

True: Tomato_Bacterial_spot
Predicted: Tomato_Bacterial_spot

True: Tomato__Tomato_YellowLeaf__Curl_Virus
Predicted: Tomato__Tomato_YellowLeaf__Curl_Virus

True: Tomato_Bacterial_spot
Predicted: Tomato_Bacterial_spot

True: Tomato_Late_blight
Predicted: Tomato_Late_blight

True: Tomato_Bacterial_spot
Predicted: Tomato_Bacterial_spot

True: Tomato_Late_blight
Predicted: Tomato_Late_blight

True: Tomato__Tomato_YellowLeaf__Curl_Virus
Predicted: Tomato__Tomato_YellowLeaf__Curl_Virus

True: Tomato_Late_blight
Predicted: Tomato_Late_blight

True: Tomato_Late_blight
Predicted: Tomato_Late_blight