

In [1]: pip install tensorflow

```
Note: you may need to restart the kernel to use updated packages. Collecting tensorflow
  Downloading tensorflow-2.10.0-cp39-cp39-win_amd64.whl (455.9 MB)
    ----- 455.9/455.9 MB 1.8 MB/s eta 0:00:00
Collecting google-pasta>=0.1.1
  Downloading google_pasta-0.2.0-py3-none-any.whl (57 kB)
    ----- 57.5/57.5 kB 3.1 MB/s eta 0:00:00
Requirement already satisfied: packaging in c:\users\hp\anaconda3\lib\site-packages (from tensorflow) (21.3)
Collecting libclang>=13.0.0
  Downloading libclang-14.0.6-py2.py3-none-win_amd64.whl (14.2 MB)
    ----- 14.2/14.2 MB 5.8 MB/s eta 0:00:00
Collecting tensorflow-estimator<2.11,>=2.10.0
  Downloading tensorflow_estimator-2.10.0-py2.py3-none-any.whl (438 kB)
    ----- 438.7/438.7 kB 6.8 MB/s eta 0:00:00
Collecting tensorboard<2.11,>=2.10
  Downloading tensorboard-2.10.1-py3-none-any.whl (5.9 MB)
    ----- 5.9/5.9 MB 6.1 MB/s eta 0:00:00
Collecting keras<2.11,>=2.10.0
  Downloading keras-2.10.0-py2.py3-none-any.whl (1.7 MB)
    ----- 1.7/1.7 MB 5.6 MB/s eta 0:00:00
Collecting protobuf<3.20,>=3.9.2
  Downloading protobuf-3.19.6-cp39-cp39-win_amd64.whl (895 kB)
    ----- 895.9/895.9 kB 7.1 MB/s eta 0:00:00
Collecting absl-py>=1.0.0
  Downloading absl_py-1.3.0-py3-none-any.whl (124 kB)
    ----- 124.6/124.6 kB 3.6 MB/s eta 0:00:00
Collecting termcolor>=1.1.0
  Downloading termcolor-2.1.0-py3-none-any.whl (5.8 kB)
Requirement already satisfied: typing-extensions>=3.6.6 in c:\users\hp\anaconda3\lib\site-packages (from tensorflow) (4.3.0)
Collecting tensorflow-io-gcs-filesystem>=0.23.1
  Downloading tensorflow_io_gcs_filesystem-0.27.0-cp39-cp39-win_amd64.whl (1.5 MB)
    ----- 1.5/1.5 MB 6.7 MB/s eta 0:00:00
Requirement already satisfied: wrapt>=1.11.0 in c:\users\hp\anaconda3\lib\site-packages (from tensorflow) (1.14.1)
Collecting flatbuffers>=2.0
  Downloading flatbuffers-22.10.26-py2.py3-none-any.whl (26 kB)
Requirement already satisfied: setuptools in c:\users\hp\anaconda3\lib\site-packages (from tensorflow) (63.4.1)
Requirement already satisfied: numpy>=1.20 in c:\users\hp\anaconda3\lib\site-packages (from tensorflow) (1.21.5)
Requirement already satisfied: six>=1.12.0 in c:\users\hp\anaconda3\lib\site-packages (from tensorflow) (1.16.0)
Collecting gast<=0.4.0,>=0.2.1
  Downloading gast-0.4.0-py3-none-any.whl (9.8 kB)
Collecting astunparse>=1.6.0
```

```
        Downloading astunparse-1.6.3-py2.py3-none-any.whl (12 kB)
Collecting keras-preprocessing>=1.1.1
    Downloading Keras_Preprocessing-1.1.2-py2.py3-none-any.whl (42 kB)
      ----- 42.6/42.6 kB ? eta 0:00:00
Requirement already satisfied: h5py>=2.9.0 in c:\users\hp\anaconda3\lib\site-packages (from tensorflow) (3.7.0)
Collecting grpcio<2.0,>=1.24.3
    Downloading grpcio-1.50.0-cp39-cp39-win_amd64.whl (3.7 MB)
      ----- 3.7/3.7 MB 6.1 MB/s eta 0:00:00
Collecting opt-einsum>=2.3.2
    Downloading opt_einsum-3.3.0-py3-none-any.whl (65 kB)
      ----- 65.5/65.5 kB 3.5 MB/s eta 0:00:00
Requirement already satisfied: wheel<1.0,>=0.23.0 in c:\users\hp\anaconda3\lib\site-packages (from astunparse>=1.6.0->tensorflow) (0.37.1)
Requirement already satisfied: requests<3,>=2.21.0 in c:\users\hp\anaconda3\lib\site-packages (from tensorboard<2.11,>=2.10->tensorflow) (2.28.1)
Collecting tensorboard-data-server<0.7.0,>=0.6.0
    Downloading tensorboard_data_server-0.6.1-py3-none-any.whl (2.4 kB)
Collecting google-auth-oauthlib<0.5,>=0.4.1
    Downloading google_auth_oauthlib-0.4.6-py2.py3-none-any.whl (18 kB)
Collecting google-auth<3,>=1.6.3
    Downloading google_auth-2.14.0-py2.py3-none-any.whl (175 kB)
      ----- 175.0/175.0 kB 11.0 MB/s eta 0:00:00
Requirement already satisfied: markdown>=2.6.8 in c:\users\hp\anaconda3\lib\site-packages (from tensorboard<2.11,>=2.10->tensorflow) (3.3.4)
Requirement already satisfied: werkzeug>=1.0.1 in c:\users\hp\anaconda3\lib\site-packages (from tensorboard<2.11,>=2.10->tensorflow) (2.0.3)
Collecting tensorboard-plugin-wit>=1.6.0
    Downloading tensorboard_plugin_wit-1.8.1-py3-none-any.whl (781 kB)
      ----- 781.3/781.3 kB 2.2 MB/s eta 0:00:00
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in c:\users\hp\anaconda3\lib\site-packages (from packaging->tensorflow) (3.0.9)
Collecting cachetools<6.0,>=2.0.0
    Downloading cachetools-5.2.0-py3-none-any.whl (9.3 kB)
Requirement already satisfied: pyasn1-modules>=0.2.1 in c:\users\hp\anaconda3\lib\site-packages (from google-auth<3,>=1.6.3->tensorboard<2.11,>=2.10->tens orflow) (0.2.8)
Collecting rsa<5,>=3.1.4
    Downloading rsa-4.9-py3-none-any.whl (34 kB)
Collecting requests-oauthlib>=0.7.0
    Downloading requests_oauthlib-1.3.1-py2.py3-none-any.whl (23 kB)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\hp\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorboard<2.11,>=2.10->ten sorflow) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\hp\anaconda3\lib\site -packages (from requests<3,>=2.21.0->tensorboard<2.11,>=2.10->tensorflow) (3. 3)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\hp\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorboard<2.11,>=2.10->tensorflow) (2022.9.14)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\hp\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorboard<2.11,>=2.10->tensorflow) (1.26.11)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in c:\users\hp\anaconda3
```

```
\lib\site-packages (from pyasn1-modules>=0.2.1->google-auth<3,>=1.6.3->tensor  
board<2.11,>=2.10->tensorflow) (0.4.8)  
Collecting oauthlib>=3.0.0  
    Downloading oauthlib-3.2.2-py3-none-any.whl (151 kB)  
       151.7/151.7 kB 2.3 MB/s eta 0:00:  
 00  
Installing collected packages: tensorflow-plugin-wit, libclang, keras, flatb  
uffers, termcolor, tensorflow-io-gcs-filesystem, tensorflow-estimator, tensor  
board-data-server, rsa, protobuf, opt-einsum, oauthlib, keras-preprocessing,  
grpcio, google-pasta, gast, cachetools, astunparse, absl-py, requests-oauthli  
b, google-auth, google-auth-oauthlib, tensorflow, tensorflow  
Successfully installed absl-py-1.3.0 astunparse-1.6.3 cachetools-5.2.0 flatbu  
ffers-22.10.26 gast-0.4.0 google-auth-2.14.0 google-auth-oauthlib-0.4.6 googl  
e-pasta-0.2.0 grpcio-1.50.0 keras-2.10.0 keras-preprocessing-1.1.2 libclang-1  
4.0.6 oauthlib-3.2.2 opt-einsum-3.3.0 protobuf-3.19.6 requests-oauthlib-1.3.1  
rsa-4.9 tensorflow-2.10.1 tensorflow-data-server-0.6.1 tensorflow-plugin-w  
it-1.8.1 tensorflow-2.10.0 tensorflow-estimator-2.10.0 tensorflow-io-gcs-file  
system-0.27.0 termcolor-2.1.0
```

Import all the Dependencies

```
In [2]: import tensorflow as tf  
from tensorflow.keras import models, layers  
import matplotlib.pyplot as plt  
from IPython.display import HTML
```

Set all the Constants

```
In [3]: BATCH_SIZE = 32  
IMAGE_SIZE = 256  
CHANNELS=3  
EPOCHS=50
```

Import data into tensorflow dataset object

```
In [8]: dataset = tf.keras.preprocessing.image_dataset_from_directory("D:\\PlantVillage"  
,  
    seed=123,  
    shuffle=True,  
    image_size=(IMAGE_SIZE,IMAGE_SIZE),  
    batch_size=BATCH_SIZE  
)
```

Found 2152 files belonging to 3 classes.

```
In [9]: class_names = dataset.class_names  
class_names
```

```
Out[9]: ['Potato__Early_blight', 'Potato__Late_blight', 'Potato__healthy']
```

```
In [10]: for image_batch, labels_batch in dataset.take(1):  
    print(image_batch.shape)  
    print(labels_batch.numpy())
```

```
(32, 256, 256, 3)  
[1 1 1 0 0 0 0 0 1 1 1 1 0 1 0 1 1 1 0 1 0 1 0 0 1 0 0 1 1 2 0 0]
```

Visualize some of the images from our dataset

```
In [11]: plt.figure(figsize=(10, 10))
for image_batch, labels_batch in dataset.take(1):
    for i in range(12):
        ax = plt.subplot(3, 4, i + 1)
        plt.imshow(image_batch[i].numpy().astype("uint8"))
        plt.title(class_names[labels_batch[i]])
        plt.axis("off")
```

Potato__Early_blight



Potato__Early_blight



Potato__Early_blight



Potato__Late_blight



Potato__Early_blight



Potato__Early_blight



Potato__Late_blight



Potato__Early_blight



Potato__Late_blight



Potato__Early_blight



Potato__Early_blight



Potato__Early_blight



```
In [12]: def get_dataset_partitions_tf(ds, train_split=0.8, val_split=0.1, test_split=0.1,
    assert (train_split + test_split + val_split) == 1

    ds_size = len(ds)

    if shuffle:
        ds = ds.shuffle(shuffle_size, seed=12)

    train_size = int(train_split * ds_size)
    val_size = int(val_split * ds_size)

    train_ds = ds.take(train_size)
    val_ds = ds.skip(train_size).take(val_size)
    test_ds = ds.skip(train_size).skip(val_size)

    return train_ds, val_ds, test_ds
```

```
In [13]: train_ds, val_ds, test_ds = get_dataset_partitions_tf(dataset)
```

```
In [14]: len(train_ds)
```

Out[14]: 54

```
In [15]: len(val_ds)
```

Out[15]: 6

```
In [16]: len(test_ds)
```

Out[16]: 8

Cache, Shuffle, and Prefetch the Dataset

```
In [17]: train_ds = train_ds.cache().shuffle(1000).prefetch(buffer_size=tf.data.AUTOTUNE)
val_ds = val_ds.cache().shuffle(1000).prefetch(buffer_size=tf.data.AUTOTUNE)
test_ds = test_ds.cache().shuffle(1000).prefetch(buffer_size=tf.data.AUTOTUNE)
```

Building the Model

Creating a Layer for Resizing and Normalization

```
In [18]: resize_and_rescale = tf.keras.Sequential([
    layers.experimental.preprocessing.Resizing(IMAGE_SIZE, IMAGE_SIZE),
    layers.experimental.preprocessing.Rescaling(1./255),
])
```

Data Augmentation

```
In [19]: data_augmentation = tf.keras.Sequential([
    layers.experimental.preprocessing.RandomFlip("horizontal_and_vertical"),
    layers.experimental.preprocessing.RandomRotation(0.2),
])
```

Applying Data Augmentation to Train Dataset

```
In [20]: train_ds = train_ds.map(
    lambda x, y: (data_augmentation(x, training=True), y)
).prefetch(buffer_size=tf.data.AUTOTUNE)
```

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

Model Architecture

```
In [21]: input_shape = (BATCH_SIZE, IMAGE_SIZE, IMAGE_SIZE, CHANNELS)
n_classes = 3

model = models.Sequential([
    resize_and_rescale,
    layers.Conv2D(32, kernel_size = (3,3), activation='relu', input_shape=input_s
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, kernel_size = (3,3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, kernel_size = (3,3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Flatten(),
    layers.Dense(64, activation='relu'),
    layers.Dense(n_classes, activation='softmax'),
])

model.build(input_shape=input_shape)
```

```
In [25]: model.summary()
```

Model: "sequential_2"

Layer (type)	Output Shape	Param #
<hr/>		
sequential (Sequential)	(None, 256, 256, 3)	0
conv2d (Conv2D)	(None, 254, 254, 32)	896
max_pooling2d (MaxPooling2D)	(None, 127, 127, 32)	0
conv2d_1 (Conv2D)	(None, 125, 125, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 62, 62, 64)	0
conv2d_2 (Conv2D)	(None, 60, 60, 64)	36928
max_pooling2d_2 (MaxPooling2D)	(None, 30, 30, 64)	0
conv2d_3 (Conv2D)	(None, 28, 28, 64)	36928
max_pooling2d_3 (MaxPooling2D)	(None, 14, 14, 64)	0
conv2d_4 (Conv2D)	(None, 12, 12, 64)	36928
max_pooling2d_4 (MaxPooling2D)	(None, 6, 6, 64)	0
conv2d_5 (Conv2D)	(None, 4, 4, 64)	36928
max_pooling2d_5 (MaxPooling2D)	(None, 2, 2, 64)	0
flatten (Flatten)	(None, 256)	0
dense (Dense)	(None, 64)	16448
dense_1 (Dense)	(None, 3)	195
<hr/>		
Total params: 183,747		
Trainable params: 183,747		
Non-trainable params: 0		

Compiling the Model

```
In [23]: model.compile(  
    optimizer='adam',  
    loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=False),  
    metrics=['accuracy'])
```

In [24]: history = model.fit(

```
    train_ds,  
    batch_size=BATCH_SIZE,  
    validation_data=val_ds,  
    verbose=1,  
    epochs=50,  
)
```

```
Epoch 1/50  
54/54 [=====] - 64s 1s/step - loss: 0.9033 - accuracy: 0.4902 - val_loss: 0.8519 - val_accuracy: 0.4896  
Epoch 2/50  
54/54 [=====] - 66s 1s/step - loss: 0.6651 - accuracy: 0.7205 - val_loss: 0.5658 - val_accuracy: 0.7760  
Epoch 3/50  
54/54 [=====] - 67s 1s/step - loss: 0.4275 - accuracy: 0.8466 - val_loss: 0.4906 - val_accuracy: 0.7708  
Epoch 4/50  
54/54 [=====] - 75s 1s/step - loss: 0.3099 - accuracy: 0.8762 - val_loss: 0.2711 - val_accuracy: 0.8854  
Epoch 5/50  
54/54 [=====] - 76s 1s/step - loss: 0.2212 - accuracy: 0.9184 - val_loss: 0.3150 - val_accuracy: 0.8750  
Epoch 6/50  
54/54 [=====] - 72s 1s/step - loss: 0.1878 - accuracy: 0.9277 - val_loss: 0.2308 - val_accuracy: 0.9219  
Epoch 7/50  
54/54 [=====] - 70s 1s/step - loss: 0.1740 - accuracy: 0.9323 - val_loss: 0.2449 - val_accuracy: 0.9062  
Epoch 8/50  
54/54 [=====] - 73s 1s/step - loss: 0.1714 - accuracy: 0.9346 - val_loss: 0.2229 - val_accuracy: 0.9062  
Epoch 9/50  
54/54 [=====] - 73s 1s/step - loss: 0.1397 - accuracy: 0.9456 - val_loss: 0.2009 - val_accuracy: 0.9167  
Epoch 10/50  
54/54 [=====] - 72s 1s/step - loss: 0.1279 - accuracy: 0.9543 - val_loss: 0.1362 - val_accuracy: 0.9479  
Epoch 11/50  
54/54 [=====] - 72s 1s/step - loss: 0.1155 - accuracy: 0.9566 - val_loss: 0.1447 - val_accuracy: 0.9271  
Epoch 12/50  
54/54 [=====] - 74s 1s/step - loss: 0.1059 - accuracy: 0.9578 - val_loss: 0.1528 - val_accuracy: 0.9271  
Epoch 13/50  
54/54 [=====] - 72s 1s/step - loss: 0.1268 - accuracy: 0.9514 - val_loss: 0.1871 - val_accuracy: 0.9323  
Epoch 14/50  
54/54 [=====] - 71s 1s/step - loss: 0.1103 - accuracy: 0.9537 - val_loss: 0.1811 - val_accuracy: 0.9219  
Epoch 15/50  
54/54 [=====] - 71s 1s/step - loss: 0.0860 - accuracy: 0.9676 - val_loss: 0.4553 - val_accuracy: 0.8854  
Epoch 16/50  
54/54 [=====] - 73s 1s/step - loss: 0.1096 - accuracy: 0.9589 - val_loss: 0.1370 - val_accuracy: 0.9479
```

```
Epoch 17/50
54/54 [=====] - 72s 1s/step - loss: 0.1132 - accuracy: 0.9589 - val_loss: 0.1642 - val_accuracy: 0.9375
Epoch 18/50
54/54 [=====] - 72s 1s/step - loss: 0.0824 - accuracy: 0.9728 - val_loss: 0.1411 - val_accuracy: 0.9583
Epoch 19/50
54/54 [=====] - 71s 1s/step - loss: 0.0811 - accuracy: 0.9734 - val_loss: 0.4807 - val_accuracy: 0.8646
Epoch 20/50
54/54 [=====] - 69s 1s/step - loss: 0.0886 - accuracy: 0.9635 - val_loss: 0.2646 - val_accuracy: 0.9115
Epoch 21/50
54/54 [=====] - 70s 1s/step - loss: 0.0795 - accuracy: 0.9688 - val_loss: 0.1649 - val_accuracy: 0.9479
Epoch 22/50
54/54 [=====] - 71s 1s/step - loss: 0.0669 - accuracy: 0.9786 - val_loss: 0.1153 - val_accuracy: 0.9427
Epoch 23/50
54/54 [=====] - 72s 1s/step - loss: 0.0654 - accuracy: 0.9705 - val_loss: 0.1701 - val_accuracy: 0.9375
Epoch 24/50
54/54 [=====] - 74s 1s/step - loss: 0.0629 - accuracy: 0.9769 - val_loss: 0.1090 - val_accuracy: 0.9531
Epoch 25/50
54/54 [=====] - 75s 1s/step - loss: 0.0699 - accuracy: 0.9688 - val_loss: 0.2560 - val_accuracy: 0.9062
Epoch 26/50
54/54 [=====] - 81s 1s/step - loss: 0.0791 - accuracy: 0.9728 - val_loss: 0.4425 - val_accuracy: 0.8646
Epoch 27/50
54/54 [=====] - 79s 1s/step - loss: 0.0927 - accuracy: 0.9641 - val_loss: 0.1354 - val_accuracy: 0.9583
Epoch 28/50
54/54 [=====] - 76s 1s/step - loss: 0.0576 - accuracy: 0.9803 - val_loss: 0.2572 - val_accuracy: 0.9375
Epoch 29/50
54/54 [=====] - 77s 1s/step - loss: 0.0532 - accuracy: 0.9826 - val_loss: 0.1488 - val_accuracy: 0.9427
Epoch 30/50
54/54 [=====] - 79s 1s/step - loss: 0.0738 - accuracy: 0.9728 - val_loss: 0.1353 - val_accuracy: 0.9531
Epoch 31/50
54/54 [=====] - 79s 1s/step - loss: 0.0431 - accuracy: 0.9850 - val_loss: 0.3989 - val_accuracy: 0.8854
Epoch 32/50
54/54 [=====] - 78s 1s/step - loss: 0.0384 - accuracy: 0.9838 - val_loss: 0.3001 - val_accuracy: 0.9010
Epoch 33/50
54/54 [=====] - 75s 1s/step - loss: 0.0745 - accuracy: 0.9740 - val_loss: 0.1572 - val_accuracy: 0.9375
Epoch 34/50
54/54 [=====] - 76s 1s/step - loss: 0.0543 - accuracy: 0.9797 - val_loss: 0.0975 - val_accuracy: 0.9688
Epoch 35/50
54/54 [=====] - 73s 1s/step - loss: 0.0386 - accuracy: 0.9844 - val_loss: 0.1779 - val_accuracy: 0.9427
```

```
Epoch 36/50
54/54 [=====] - 79s 1s/step - loss: 0.0282 - accuracy: 0.9919 - val_loss: 0.1215 - val_accuracy: 0.9583
Epoch 37/50
54/54 [=====] - 91s 2s/step - loss: 0.0443 - accuracy: 0.9861 - val_loss: 0.1595 - val_accuracy: 0.9531
Epoch 38/50
54/54 [=====] - 88s 2s/step - loss: 0.0658 - accuracy: 0.9780 - val_loss: 0.1559 - val_accuracy: 0.9375
Epoch 39/50
54/54 [=====] - 89s 2s/step - loss: 0.0470 - accuracy: 0.9809 - val_loss: 0.4642 - val_accuracy: 0.8750
Epoch 40/50
54/54 [=====] - 93s 2s/step - loss: 0.0323 - accuracy: 0.9878 - val_loss: 0.1109 - val_accuracy: 0.9531
Epoch 41/50
54/54 [=====] - 63s 1s/step - loss: 0.0319 - accuracy: 0.9884 - val_loss: 0.1865 - val_accuracy: 0.9427
Epoch 42/50
54/54 [=====] - 62s 1s/step - loss: 0.0217 - accuracy: 0.9925 - val_loss: 0.1886 - val_accuracy: 0.9479
Epoch 43/50
54/54 [=====] - 63s 1s/step - loss: 0.0444 - accuracy: 0.9878 - val_loss: 0.3748 - val_accuracy: 0.8958
Epoch 44/50
54/54 [=====] - 62s 1s/step - loss: 0.0269 - accuracy: 0.9878 - val_loss: 0.2848 - val_accuracy: 0.9219
Epoch 45/50
54/54 [=====] - 66s 1s/step - loss: 0.1015 - accuracy: 0.9688 - val_loss: 0.4152 - val_accuracy: 0.8125
Epoch 46/50
54/54 [=====] - 72s 1s/step - loss: 0.0928 - accuracy: 0.9641 - val_loss: 0.2293 - val_accuracy: 0.9323
Epoch 47/50
54/54 [=====] - 70s 1s/step - loss: 0.0450 - accuracy: 0.9826 - val_loss: 0.2478 - val_accuracy: 0.9219
Epoch 48/50
54/54 [=====] - 69s 1s/step - loss: 0.0205 - accuracy: 0.9925 - val_loss: 0.2131 - val_accuracy: 0.9479
Epoch 49/50
54/54 [=====] - 71s 1s/step - loss: 0.0247 - accuracy: 0.9913 - val_loss: 0.4754 - val_accuracy: 0.8802
Epoch 50/50
54/54 [=====] - 72s 1s/step - loss: 0.0383 - accuracy: 0.9832 - val_loss: 0.2935 - val_accuracy: 0.9323
```

In [26]: `scores = model.evaluate(test_ds)`

```
8/8 [=====] - 3s 223ms/step - loss: 0.2410 - accuracy: 0.9180
```

In [27]: `scores`

Out[27]: [0.24096325039863586, 0.91796875]

Plotting the Accuracy and Loss Curves

```
In [28]: history
```

```
Out[28]: <keras.callbacks.History at 0x2829e558d00>
```

```
In [29]: history.params
```

```
Out[29]: {'verbose': 1, 'epochs': 50, 'steps': 54}
```

```
In [30]: history.history.keys()
```

```
Out[30]: dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])
```

```
In [31]: type(history.history['loss'])
```

```
Out[31]: list
```

```
In [32]: len(history.history['loss'])
```

```
Out[32]: 50
```

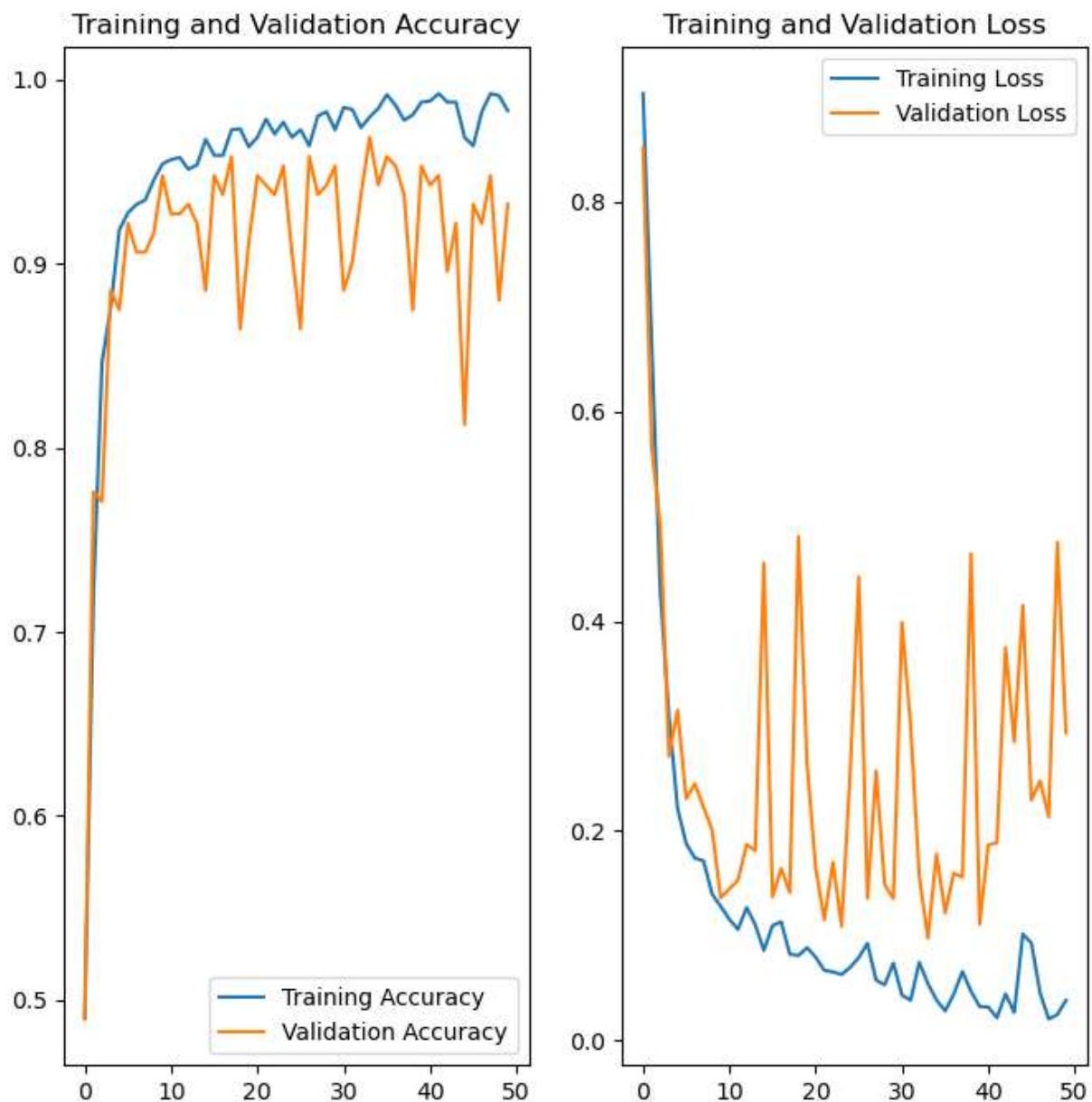
```
In [33]: history.history['loss'][:5] # show Loss for first 5 epochs
```

```
Out[33]: [0.9032962322235107,  
          0.665100634098053,  
          0.42751604318618774,  
          0.3098973035812378,  
          0.2211935669183731]
```

```
In [35]: acc = history.history['accuracy']  
val_acc = history.history['val_accuracy']  
  
loss = history.history['loss']  
val_loss = history.history['val_loss']
```

```
In [36]: plt.figure(figsize=(8, 8))
plt.subplot(1, 2, 1)
plt.plot(range(EPOCHS), acc, label='Training Accuracy')
plt.plot(range(EPOCHS), val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')

plt.subplot(1, 2, 2)
plt.plot(range(EPOCHS), loss, label='Training Loss')
plt.plot(range(EPOCHS), val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.title('Training and Validation Loss')
plt.show()
```



Run prediction on a sample image

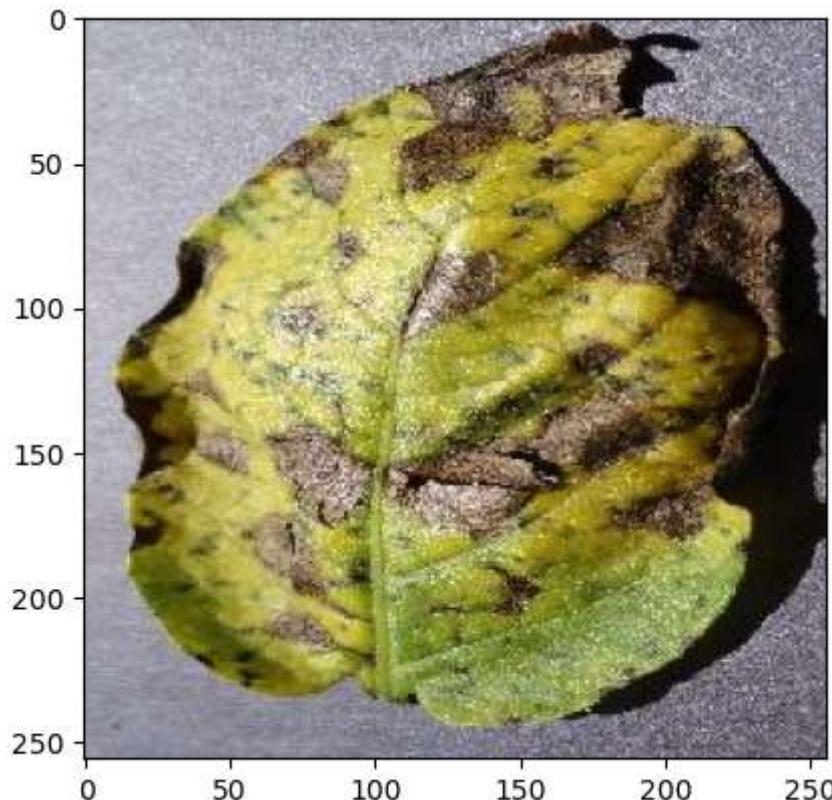
```
In [37]: import numpy as np
for images_batch, labels_batch in test_ds.take(1):

    first_image = images_batch[0].numpy().astype('uint8')
    first_label = labels_batch[0].numpy()

    print("first image to predict")
    plt.imshow(first_image)
    print("actual label:", class_names[first_label])

    batch_prediction = model.predict(images_batch)
    print("predicted label:", class_names[np.argmax(batch_prediction[0])])
```

```
first image to predict
actual label: Potato_Early_blight
1/1 [=====] - 1s 729ms/step
predicted label: Potato_Early_blight
```



Write a function for inference

```
In [38]: def predict(model, img):
    img_array = tf.keras.preprocessing.image.img_to_array(images[i].numpy())
    img_array = tf.expand_dims(img_array, 0)

    predictions = model.predict(img_array)

    predicted_class = class_names[np.argmax(predictions[0])]
    confidence = round(100 * (np.max(predictions[0])), 2)
    return predicted_class, confidence
```

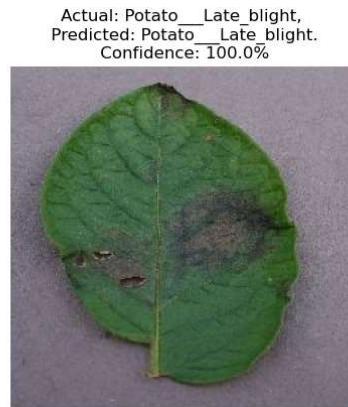
Now run inference on few sample images

```
In [39]: plt.figure(figsize=(15, 15))
for images, labels in test_ds.take(1):
    for i in range(9):
        ax = plt.subplot(3, 3, i + 1)
        plt.imshow(images[i].numpy().astype("uint8"))

    predicted_class, confidence = predict(model, images[i].numpy())
    actual_class = class_names[labels[i]]

    plt.title(f"Actual: {actual_class},\n Predicted: {predicted_class}.\n Cor
    plt.axis("off")
```

```
1/1 [=====] - 0s 267ms/step
1/1 [=====] - 0s 79ms/step
1/1 [=====] - 0s 77ms/step
1/1 [=====] - 0s 51ms/step
1/1 [=====] - 0s 67ms/step
1/1 [=====] - 0s 66ms/step
1/1 [=====] - 0s 67ms/step
1/1 [=====] - 0s 51ms/step
1/1 [=====] - 0s 63ms/step
```



Saving the Model

```
In [44]: import os
model_version=max([int(i) for i in os.listdir("../models") + [0]])+1
model.save(f"../models/{model_version}")
```

WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showing 5 of 6). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: ../models/2\assets

INFO:tensorflow:Assets written to: ../models/2\assets

```
In [45]: model.save("../potatoes.h5")
```

