

notebook1e3c17a8c1-1

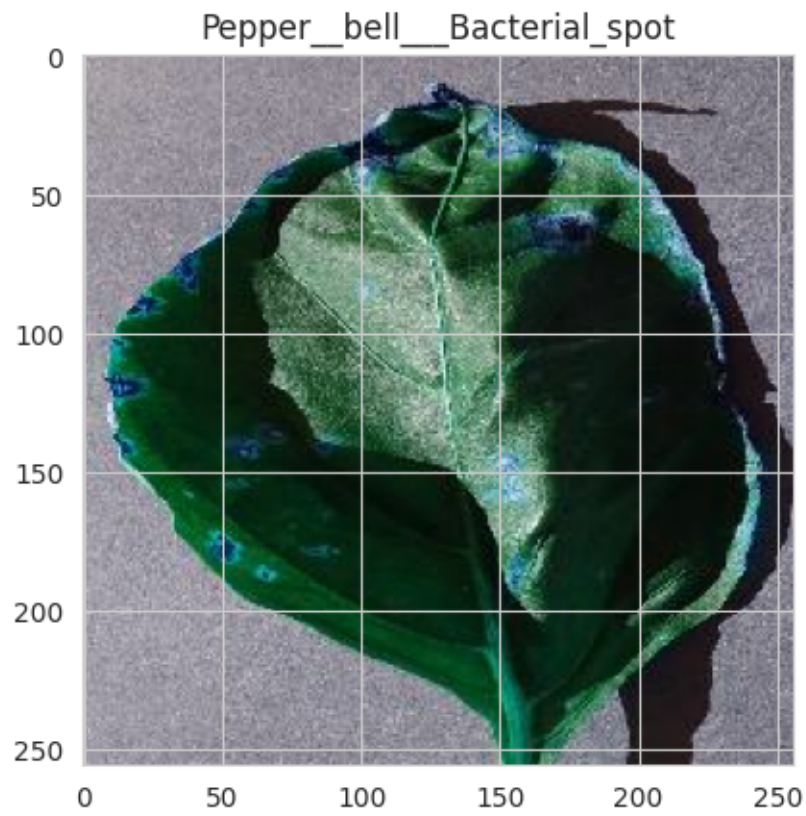
December 2, 2023

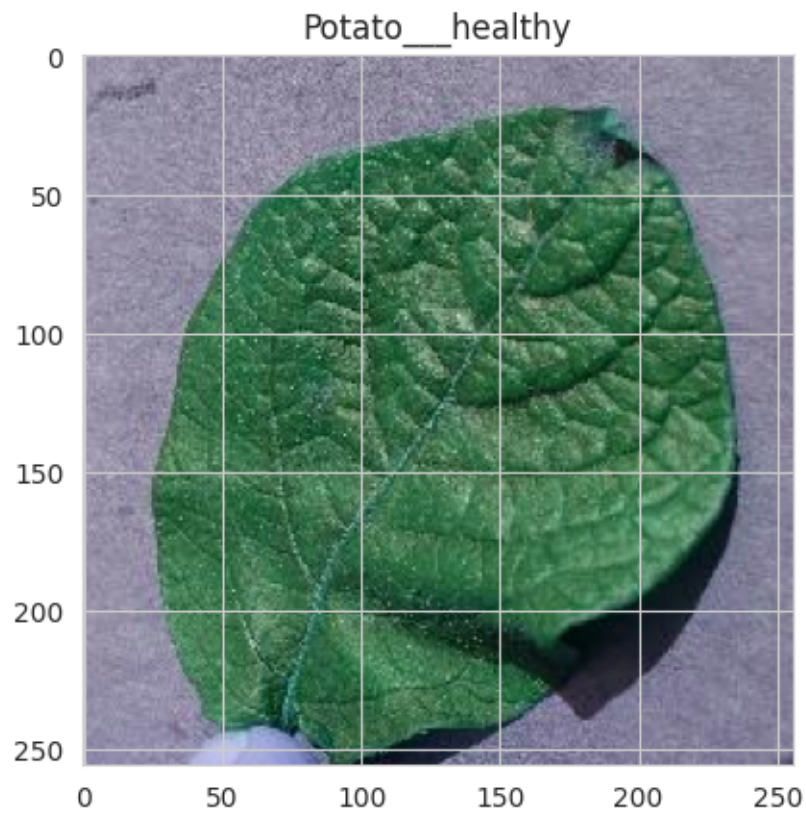
```
[9]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import cv2
import os
import seaborn as sns

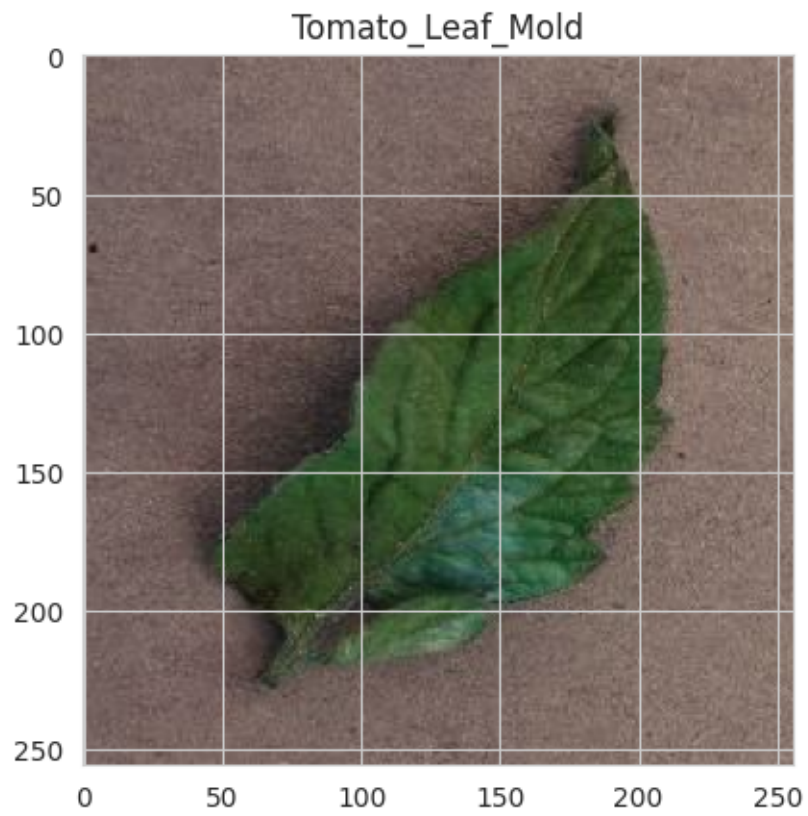
import tensorflow as tf
from tensorflow.keras.models import *
from tensorflow.keras.layers import *
from tensorflow.keras.preprocessing.image import ImageDataGenerator, \
    img_to_array
```

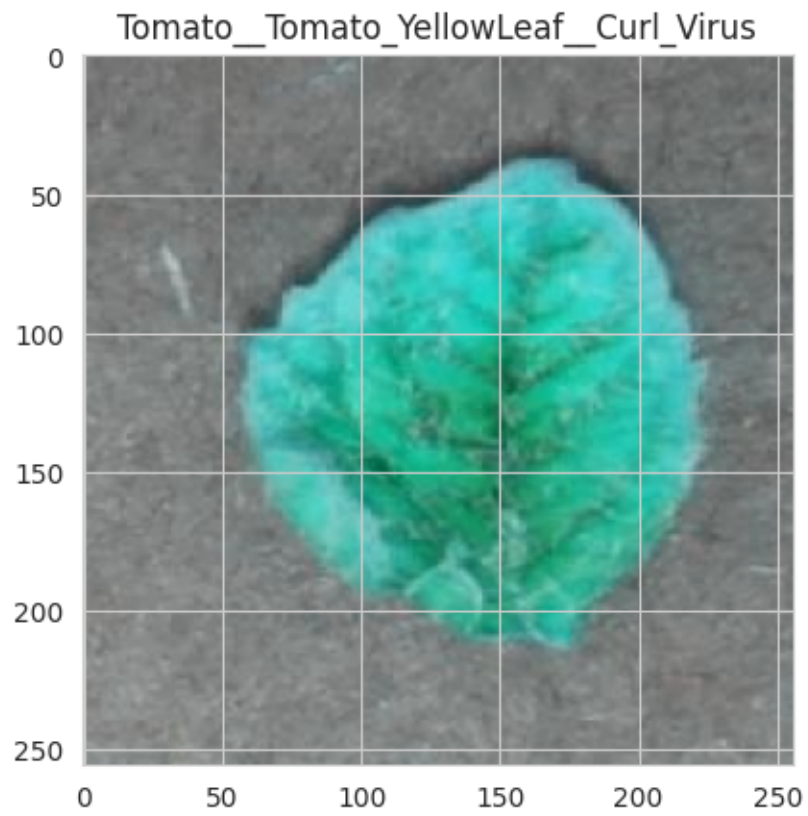
```
[10]: DATA_PATH = '../input/plantdisease/PlantVillage'

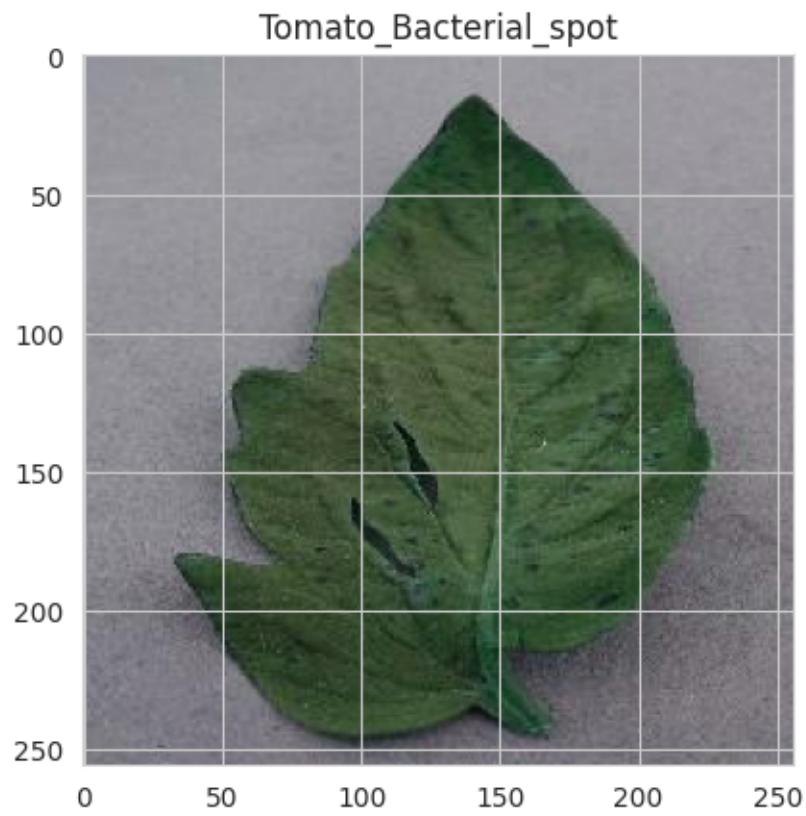
for cat in os.listdir(DATA_PATH):
    path = os.path.join(DATA_PATH, cat)
    for img in os.listdir(path):
        image = cv2.imread(os.path.join(path, img), cv2.IMREAD_UNCHANGED)
        plt.imshow(image)
        plt.title(f'{cat}')
        plt.show()
    break
```

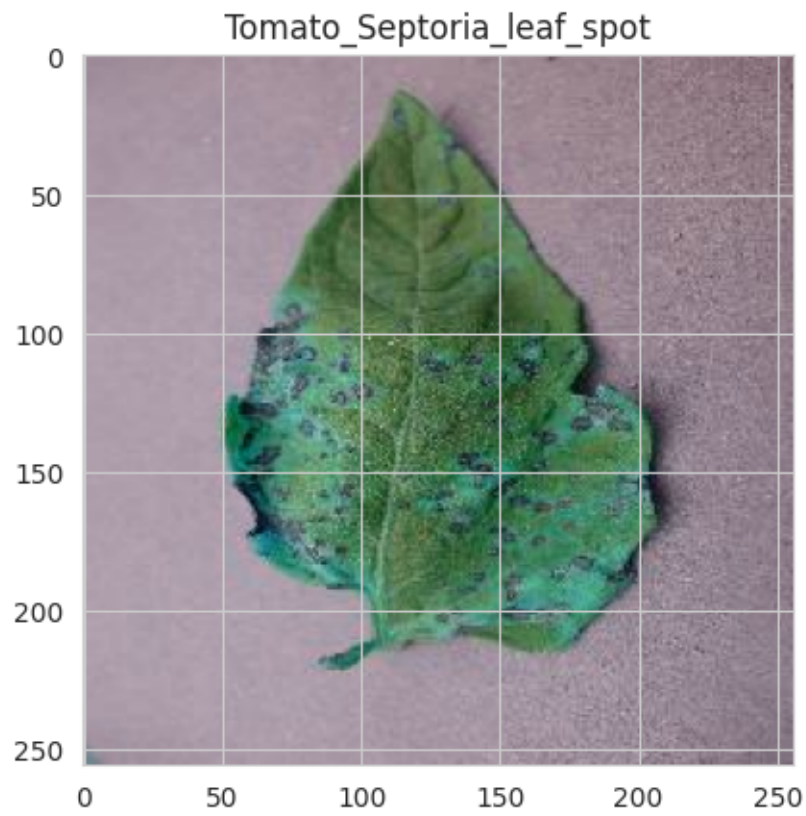


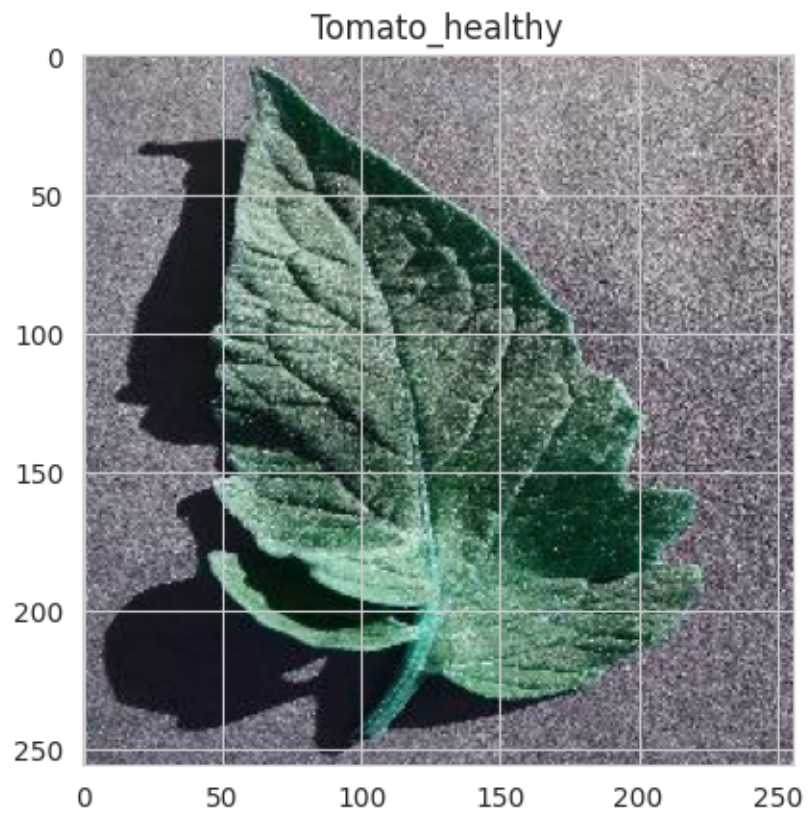


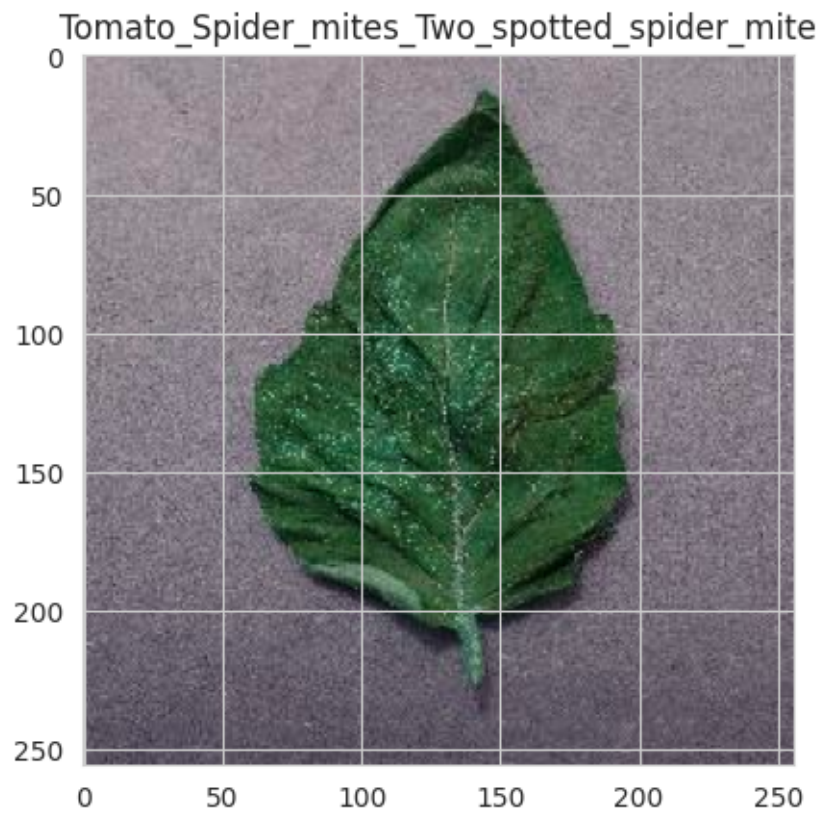


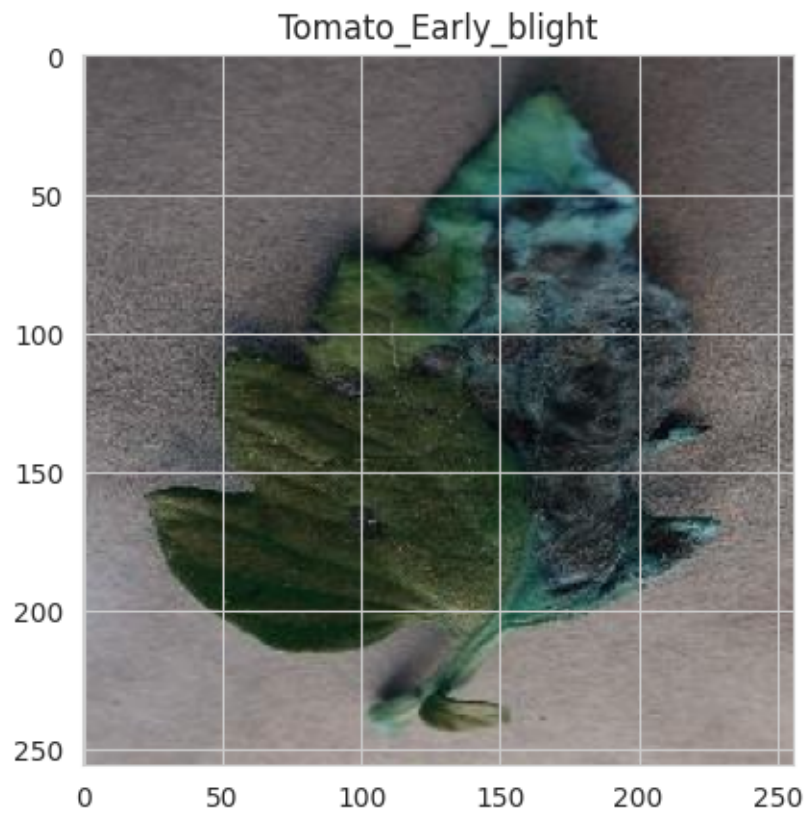


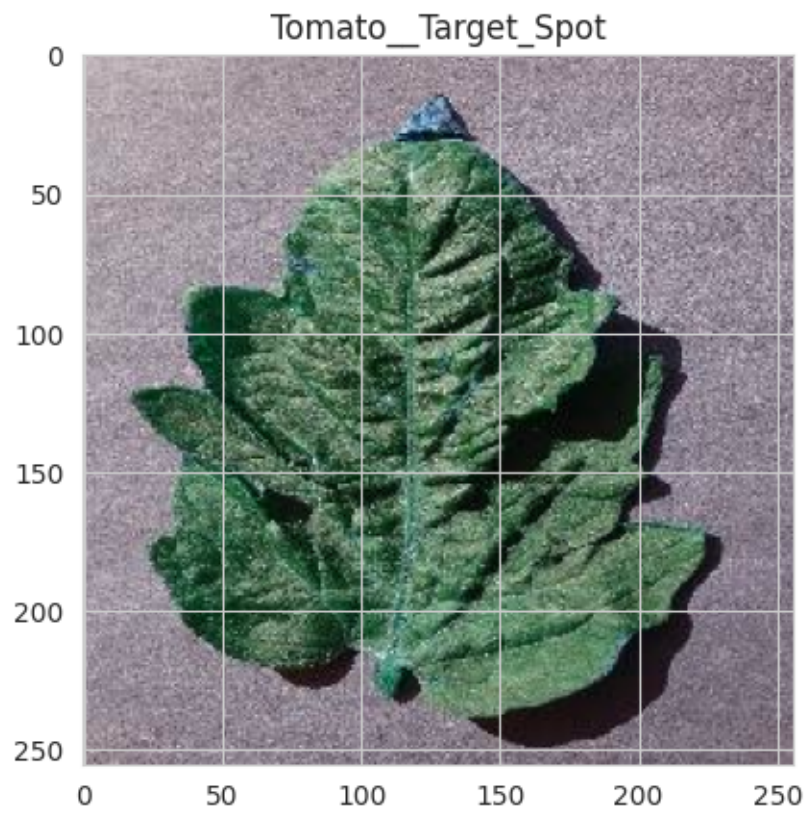


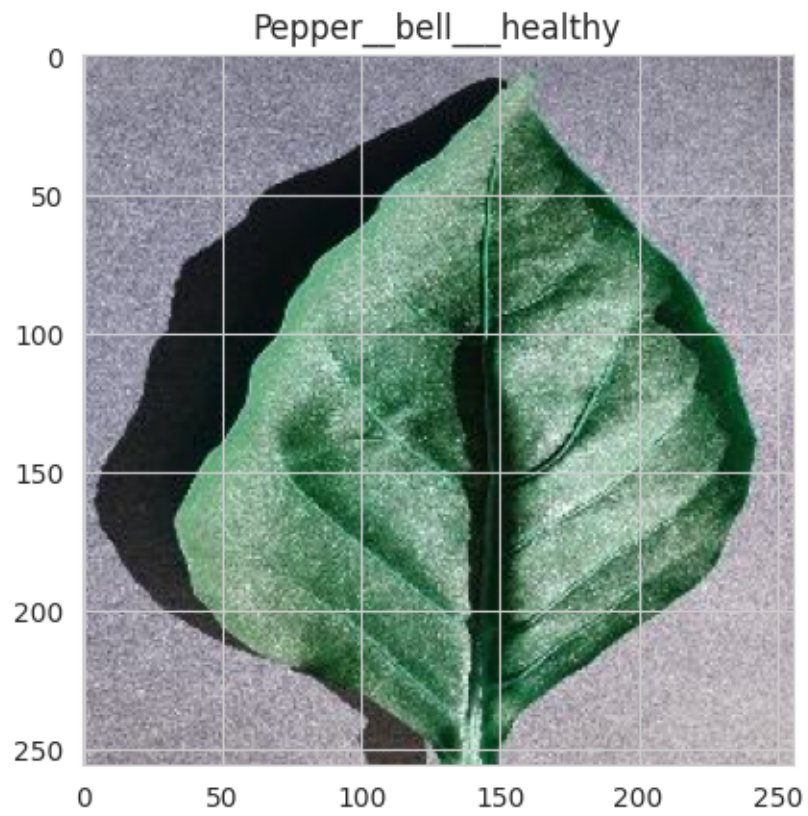


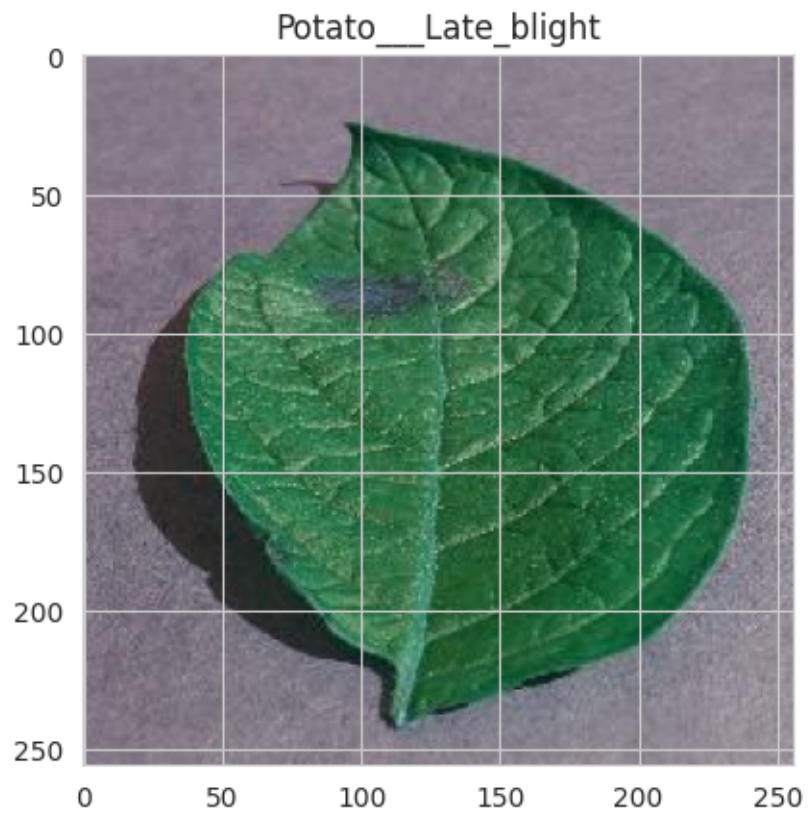


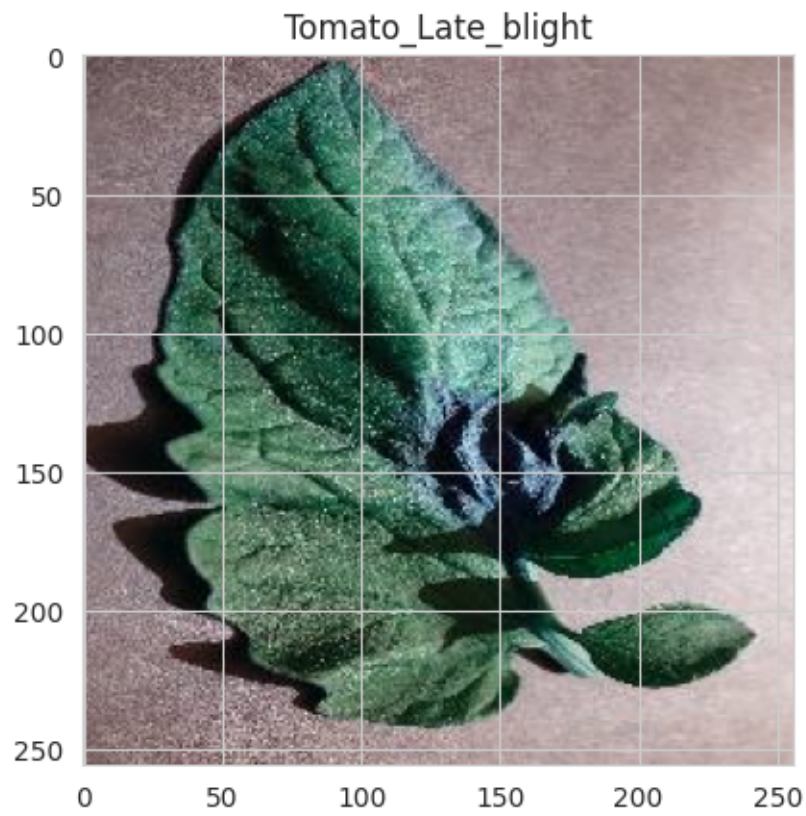


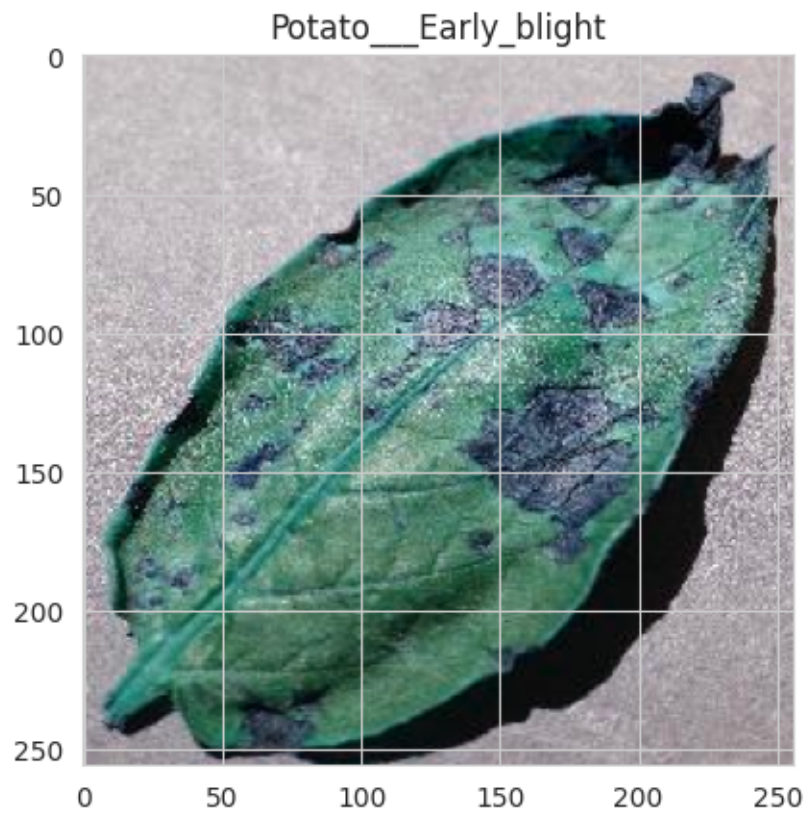


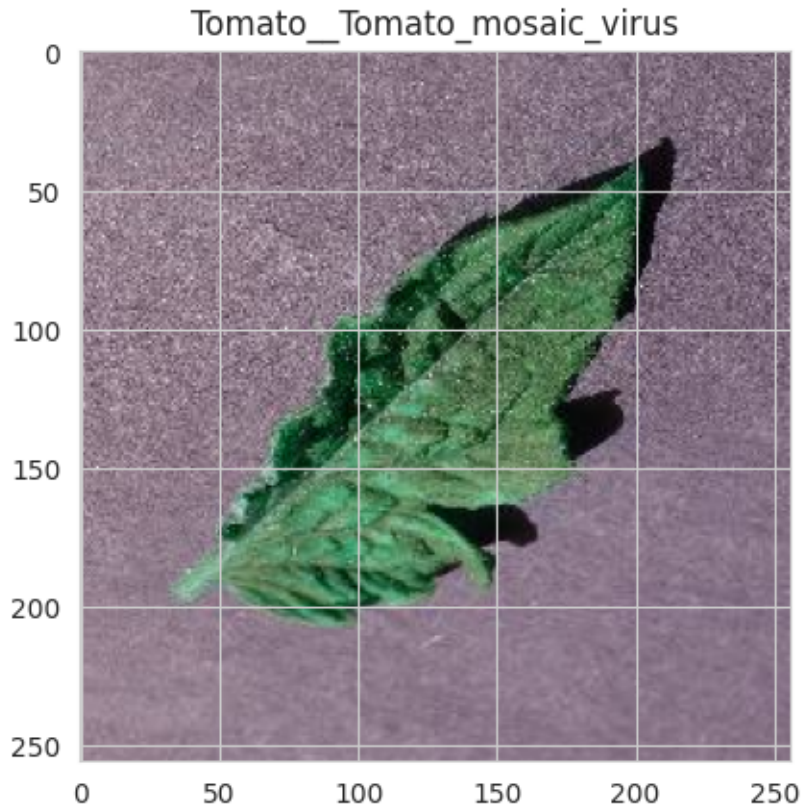












```
[11]: IMG_SHAPE = (224, 224)
INPUT_SHAPE = [224, 224, 3]
EPOCHS = 50
BS = 32
img_data_gen = ImageDataGenerator(rescale=1./255, rotation_range=0.2,
    ↪horizontal_flip=True, vertical_flip=True,
    ↪shear_range=0.2, validation_split=0.25)

train_data_gen = img_data_gen.flow_from_directory(DATA_PATH, batch_size=BS,
    ↪subset='training',
    ↪shuffle=True,
    class_mode='categorical',

val_data_gen = img_data_gen.flow_from_directory(DATA_PATH, batch_size=BS,
    ↪subset='validation',
    ↪shuffle=True,
    class_mode='categorical',
```

Found 15482 images belonging to 15 classes.

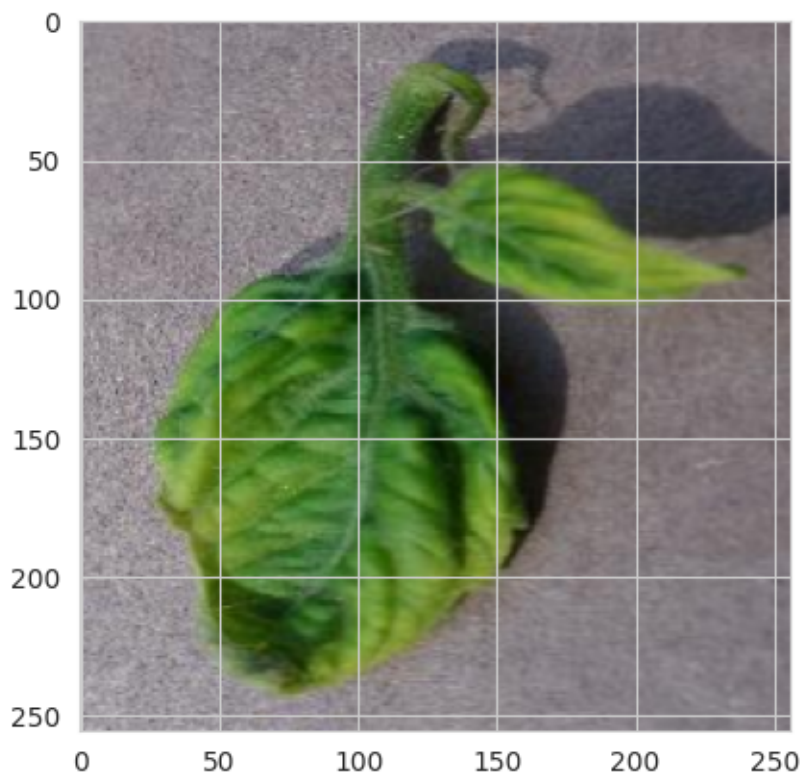
Found 5156 images belonging to 15 classes.


```
[12]: label = train_data_gen.class_indices
label
```

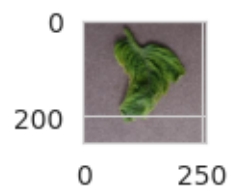
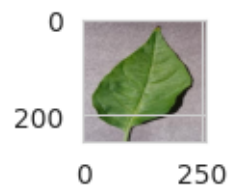
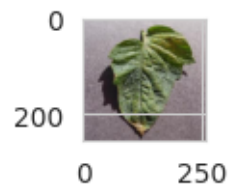
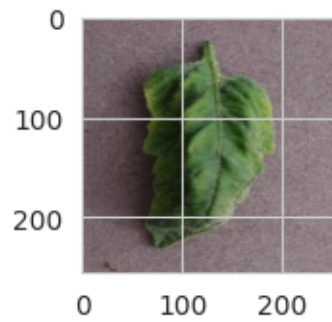
```
[12]: {'Pepper__bell__Bacterial_spot': 0,
      'Pepper__bell__healthy': 1,
      'Potato__Early_blight': 2,
      'Potato__Late_blight': 3,
      'Potato__healthy': 4,
      'Tomato_Bacterial_spot': 5,
      'Tomato_Early_blight': 6,
      'Tomato_Late_blight': 7,
      'Tomato_Leaf_Mold': 8,
      'Tomato_Septoria_leaf_spot': 9,
      'Tomato_Spider_mites_Two_spotted_spider_mite': 10,
      'Tomato__Target_Spot': 11,
      'Tomato__Tomato_YellowLeaf__Curl_Virus': 12,
      'Tomato__Tomato_mosaic_virus': 13,
      'Tomato_healthy': 14}
```

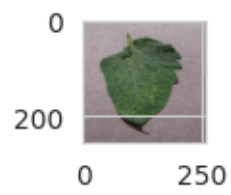
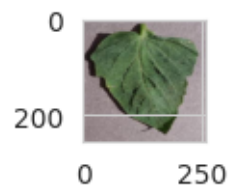
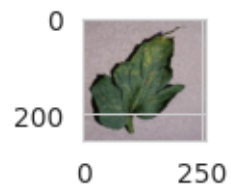
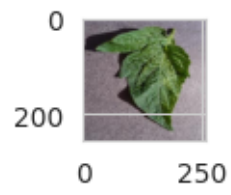
```
[13]: img = train_data_gen.__getitem__(11)[0]
plt.imshow(img[0])
#plt.title(label[11])
```

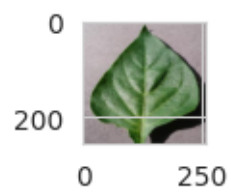
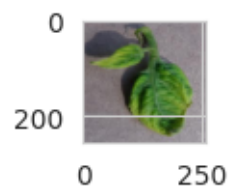
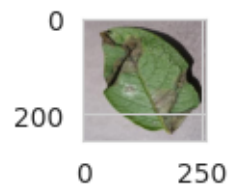
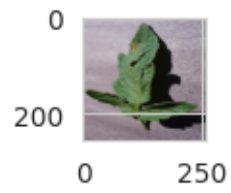
```
[13]: <matplotlib.image.AxesImage at 0x7e032dfd6320>
```



```
[14]: plt.figure(figsize=(16,10))
      for i in range(15):
          plt.subplot(5, 3, i+1)
          img = train_data_gen.__getitem__(i)[0]
          plt.imshow(img[0])
          plt.xticks()
          plt.show()
```









```
[17]: def model_building(model_name, INPUT_SHAPE=INPUT_SHAPE):
    print('Model Initialization started')
    base_model = model_name(include_top=False, weights='imagenet',
    ↪ input_shape=INPUT_SHAPE)

    for layers in base_model.layers:
        layers.trainable = False
    print('Model Initialization finished')

    #model creation
    print('Model creation started')
    inp_model = base_model.output

    x = GlobalAveragePooling2D()(inp_model)
    x = Dense(128, activation = 'relu')(x)
    x = Dense(15, activation = 'sigmoid')(x)

    model = Model(inputs = base_model.input, outputs = x)

    #model summary
    print('Model summary')
    #model.summary()

    #model compilation
    model.compile(optimizer = 'adam', metrics=['accuracy'], loss =
    ↪ 'categorical_crossentropy')

    history = model.fit(train_data_gen, validation_data=val_data_gen,
                        validation_steps=len(val_data_gen)//BS,
                        steps_per_epoch=len(train_data_gen)//BS,
                        batch_size=BS,
                        epochs=EPOCHS)

    print('Model Building Finished')

    !mkdir -p saved_model
    model.save(f'saved_model/{model_name}_1.h5')
    print('Model was saved')
```

```
return history
```

```
[18]: def evaluation_plot(model):  
    sns.set_style('whitegrid')  
  
    plt.figure(figsize=(10, 8))  
    plt.plot(model['loss'], label = 'loss')  
    plt.plot(model['accuracy'], label = 'accuracy')  
    plt.plot(model['val_loss'], label = 'val_loss')  
    plt.plot(model['val_accuracy'], label = 'val_accuracy')  
    plt.legend()  
    plt.title('Model Evaluation')  
    plt.show()
```

VGG16 model

```
[19]: from tensorflow.keras.applications.vgg16 import VGG16
```

```
[20]: vgg16_hist = model_building(VGG16)
```

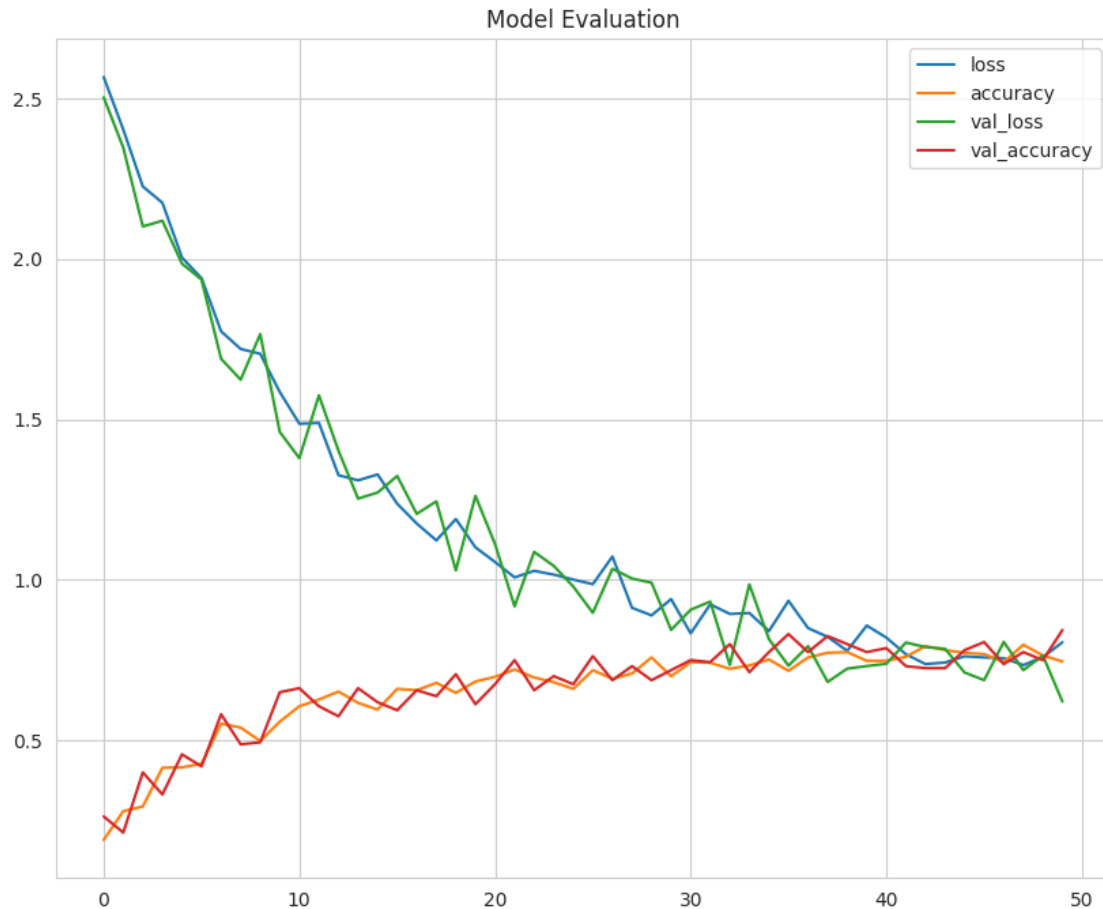
```
Model Initialization started  
Downloading data from https://storage.googleapis.com/tensorflow/keras-  
applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5  
58889256/58889256 [=====] - 3s 0us/step  
Model Initialization finished  
Model creation started  
Model summary  
Epoch 1/50  
15/15 [=====] - 14s 662ms/step - loss: 2.5674 -  
accuracy: 0.1896 - val_loss: 2.5041 - val_accuracy: 0.2625  
Epoch 2/50  
15/15 [=====] - 10s 698ms/step - loss: 2.4041 -  
accuracy: 0.2792 - val_loss: 2.3477 - val_accuracy: 0.2125  
Epoch 3/50  
15/15 [=====] - 10s 697ms/step - loss: 2.2263 -  
accuracy: 0.2937 - val_loss: 2.1019 - val_accuracy: 0.4000  
Epoch 4/50  
15/15 [=====] - 10s 683ms/step - loss: 2.1757 -  
accuracy: 0.4146 - val_loss: 2.1196 - val_accuracy: 0.3313  
Epoch 5/50  
15/15 [=====] - 11s 721ms/step - loss: 2.0050 -  
accuracy: 0.4156 - val_loss: 1.9860 - val_accuracy: 0.4563  
Epoch 6/50  
15/15 [=====] - 10s 683ms/step - loss: 1.9406 -  
accuracy: 0.4271 - val_loss: 1.9364 - val_accuracy: 0.4187  
Epoch 7/50  
15/15 [=====] - 10s 650ms/step - loss: 1.7750 -
```

accuracy: 0.5521 - val_loss: 1.6896 - val_accuracy: 0.5813
 Epoch 8/50
 15/15 [=====] - 10s 699ms/step - loss: 1.7201 -
 accuracy: 0.5396 - val_loss: 1.6242 - val_accuracy: 0.4875
 Epoch 9/50
 15/15 [=====] - 10s 659ms/step - loss: 1.7049 -
 accuracy: 0.4979 - val_loss: 1.7659 - val_accuracy: 0.4938
 Epoch 10/50
 15/15 [=====] - 10s 664ms/step - loss: 1.5857 -
 accuracy: 0.5583 - val_loss: 1.4612 - val_accuracy: 0.6500
 Epoch 11/50
 15/15 [=====] - 10s 680ms/step - loss: 1.4870 -
 accuracy: 0.6062 - val_loss: 1.3795 - val_accuracy: 0.6625
 Epoch 12/50
 15/15 [=====] - 10s 658ms/step - loss: 1.4895 -
 accuracy: 0.6271 - val_loss: 1.5754 - val_accuracy: 0.6062
 Epoch 13/50
 15/15 [=====] - 10s 684ms/step - loss: 1.3266 -
 accuracy: 0.6521 - val_loss: 1.4030 - val_accuracy: 0.5750
 Epoch 14/50
 15/15 [=====] - 10s 673ms/step - loss: 1.3105 -
 accuracy: 0.6167 - val_loss: 1.2535 - val_accuracy: 0.6625
 Epoch 15/50
 15/15 [=====] - 10s 656ms/step - loss: 1.3289 -
 accuracy: 0.5958 - val_loss: 1.2723 - val_accuracy: 0.6187
 Epoch 16/50
 15/15 [=====] - 12s 814ms/step - loss: 1.2377 -
 accuracy: 0.6604 - val_loss: 1.3240 - val_accuracy: 0.5938
 Epoch 17/50
 15/15 [=====] - 10s 668ms/step - loss: 1.1758 -
 accuracy: 0.6562 - val_loss: 1.2058 - val_accuracy: 0.6562
 Epoch 18/50
 15/15 [=====] - 10s 655ms/step - loss: 1.1233 -
 accuracy: 0.6792 - val_loss: 1.2449 - val_accuracy: 0.6375
 Epoch 19/50
 15/15 [=====] - 10s 673ms/step - loss: 1.1899 -
 accuracy: 0.6479 - val_loss: 1.0299 - val_accuracy: 0.7063
 Epoch 20/50
 15/15 [=====] - 10s 656ms/step - loss: 1.1015 -
 accuracy: 0.6833 - val_loss: 1.2620 - val_accuracy: 0.6125
 Epoch 21/50
 15/15 [=====] - 10s 658ms/step - loss: 1.0559 -
 accuracy: 0.6979 - val_loss: 1.1120 - val_accuracy: 0.6750
 Epoch 22/50
 15/15 [=====] - 10s 653ms/step - loss: 1.0080 -
 accuracy: 0.7208 - val_loss: 0.9177 - val_accuracy: 0.7500
 Epoch 23/50
 15/15 [=====] - 10s 683ms/step - loss: 1.0283 -

accuracy: 0.6958 - val_loss: 1.0874 - val_accuracy: 0.6562
 Epoch 24/50
 15/15 [=====] - 10s 655ms/step - loss: 1.0167 -
 accuracy: 0.6812 - val_loss: 1.0437 - val_accuracy: 0.7000
 Epoch 25/50
 15/15 [=====] - 10s 661ms/step - loss: 1.0010 -
 accuracy: 0.6604 - val_loss: 0.9794 - val_accuracy: 0.6750
 Epoch 26/50
 15/15 [=====] - 10s 668ms/step - loss: 0.9868 -
 accuracy: 0.7188 - val_loss: 0.8976 - val_accuracy: 0.7625
 Epoch 27/50
 15/15 [=====] - 10s 650ms/step - loss: 1.0730 -
 accuracy: 0.6917 - val_loss: 1.0348 - val_accuracy: 0.6875
 Epoch 28/50
 15/15 [=====] - 10s 666ms/step - loss: 0.9131 -
 accuracy: 0.7083 - val_loss: 1.0044 - val_accuracy: 0.7312
 Epoch 29/50
 15/15 [=====] - 10s 674ms/step - loss: 0.8895 -
 accuracy: 0.7583 - val_loss: 0.9913 - val_accuracy: 0.6875
 Epoch 30/50
 15/15 [=====] - 10s 656ms/step - loss: 0.9403 -
 accuracy: 0.7000 - val_loss: 0.8443 - val_accuracy: 0.7188
 Epoch 31/50
 15/15 [=====] - 10s 655ms/step - loss: 0.8334 -
 accuracy: 0.7437 - val_loss: 0.9070 - val_accuracy: 0.7500
 Epoch 32/50
 15/15 [=====] - 10s 658ms/step - loss: 0.9240 -
 accuracy: 0.7417 - val_loss: 0.9326 - val_accuracy: 0.7437
 Epoch 33/50
 15/15 [=====] - 10s 649ms/step - loss: 0.8940 -
 accuracy: 0.7229 - val_loss: 0.7353 - val_accuracy: 0.8000
 Epoch 34/50
 15/15 [=====] - 9s 641ms/step - loss: 0.8968 -
 accuracy: 0.7333 - val_loss: 0.9859 - val_accuracy: 0.7125
 Epoch 35/50
 15/15 [=====] - 10s 653ms/step - loss: 0.8406 -
 accuracy: 0.7521 - val_loss: 0.8156 - val_accuracy: 0.7750
 Epoch 36/50
 15/15 [=====] - 10s 650ms/step - loss: 0.9349 -
 accuracy: 0.7167 - val_loss: 0.7328 - val_accuracy: 0.8313
 Epoch 37/50
 15/15 [=====] - 10s 652ms/step - loss: 0.8499 -
 accuracy: 0.7583 - val_loss: 0.7938 - val_accuracy: 0.7750
 Epoch 38/50
 15/15 [=====] - 10s 644ms/step - loss: 0.8225 -
 accuracy: 0.7729 - val_loss: 0.6820 - val_accuracy: 0.8250
 Epoch 39/50
 15/15 [=====] - 10s 660ms/step - loss: 0.7791 -


```
accuracy: 0.7750 - val_loss: 0.7235 - val_accuracy: 0.8000
Epoch 40/50
15/15 [=====] - 9s 636ms/step - loss: 0.8577 -
accuracy: 0.7479 - val_loss: 0.7315 - val_accuracy: 0.7750
Epoch 41/50
15/15 [=====] - 10s 654ms/step - loss: 0.8202 -
accuracy: 0.7479 - val_loss: 0.7388 - val_accuracy: 0.7875
Epoch 42/50
15/15 [=====] - 10s 655ms/step - loss: 0.7683 -
accuracy: 0.7604 - val_loss: 0.8048 - val_accuracy: 0.7312
Epoch 43/50
15/15 [=====] - 10s 645ms/step - loss: 0.7378 -
accuracy: 0.7937 - val_loss: 0.7907 - val_accuracy: 0.7250
Epoch 44/50
15/15 [=====] - 10s 647ms/step - loss: 0.7426 -
accuracy: 0.7812 - val_loss: 0.7855 - val_accuracy: 0.7250
Epoch 45/50
15/15 [=====] - 10s 697ms/step - loss: 0.7620 -
accuracy: 0.7729 - val_loss: 0.7113 - val_accuracy: 0.7812
Epoch 46/50
15/15 [=====] - 10s 648ms/step - loss: 0.7583 -
accuracy: 0.7688 - val_loss: 0.6877 - val_accuracy: 0.8062
Epoch 47/50
15/15 [=====] - 10s 654ms/step - loss: 0.7560 -
accuracy: 0.7437 - val_loss: 0.8067 - val_accuracy: 0.7375
Epoch 48/50
15/15 [=====] - 9s 629ms/step - loss: 0.7346 -
accuracy: 0.7979 - val_loss: 0.7187 - val_accuracy: 0.7750
Epoch 49/50
15/15 [=====] - 10s 645ms/step - loss: 0.7594 -
accuracy: 0.7646 - val_loss: 0.7643 - val_accuracy: 0.7500
Epoch 50/50
15/15 [=====] - 10s 655ms/step - loss: 0.8056 -
accuracy: 0.7458 - val_loss: 0.6209 - val_accuracy: 0.8438
Model Building Finished
Model was saved
```

```
[21]: evaluation_plot(vgg16_hist.history)
```



InceptionV3 model

```
[22]: from tensorflow.keras.applications.inception_v3 import InceptionV3
```

```
[23]: inc_history = model_building(InceptionV3)
```

Model Initialization started

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/inception_v3/inception_v3_weights_tf_dim_ordering_tf_kernels_notop.h5

87910968/87910968 [=====] - 4s 0us/step

Model Initialization finished

Model creation started

Model summary

Epoch 1/50

15/15 [=====] - 16s 723ms/step - loss: 2.4546 - accuracy: 0.2229 - val_loss: 2.0601 - val_accuracy: 0.3750

Epoch 2/50

15/15 [=====] - 10s 663ms/step - loss: 1.8611 - accuracy: 0.4250 - val_loss: 1.7488 - val_accuracy: 0.4250

Epoch 3/50
15/15 [=====] - 10s 657ms/step - loss: 1.6116 - accuracy: 0.5271 - val_loss: 1.3987 - val_accuracy: 0.5312

Epoch 4/50
15/15 [=====] - 10s 655ms/step - loss: 1.3693 - accuracy: 0.5771 - val_loss: 1.4240 - val_accuracy: 0.5750

Epoch 5/50
15/15 [=====] - 9s 635ms/step - loss: 1.1986 - accuracy: 0.6229 - val_loss: 1.1571 - val_accuracy: 0.6187

Epoch 6/50
15/15 [=====] - 9s 639ms/step - loss: 1.1880 - accuracy: 0.6125 - val_loss: 1.1038 - val_accuracy: 0.6750

Epoch 7/50
15/15 [=====] - 10s 661ms/step - loss: 1.1647 - accuracy: 0.6292 - val_loss: 0.9240 - val_accuracy: 0.7375

Epoch 8/50
15/15 [=====] - 10s 645ms/step - loss: 0.9644 - accuracy: 0.6938 - val_loss: 0.8525 - val_accuracy: 0.7625

Epoch 9/50
15/15 [=====] - 9s 638ms/step - loss: 0.9107 - accuracy: 0.7063 - val_loss: 0.9045 - val_accuracy: 0.6812

Epoch 10/50
15/15 [=====] - 9s 642ms/step - loss: 0.9322 - accuracy: 0.7146 - val_loss: 0.8421 - val_accuracy: 0.7500

Epoch 11/50
15/15 [=====] - 9s 637ms/step - loss: 0.7951 - accuracy: 0.7479 - val_loss: 0.8100 - val_accuracy: 0.7688

Epoch 12/50
15/15 [=====] - 10s 650ms/step - loss: 0.8460 - accuracy: 0.7208 - val_loss: 0.7591 - val_accuracy: 0.7750

Epoch 13/50
15/15 [=====] - 9s 638ms/step - loss: 0.8904 - accuracy: 0.7104 - val_loss: 0.9435 - val_accuracy: 0.6250

Epoch 14/50
15/15 [=====] - 10s 656ms/step - loss: 0.7634 - accuracy: 0.7542 - val_loss: 0.9605 - val_accuracy: 0.6562

Epoch 15/50
15/15 [=====] - 9s 637ms/step - loss: 0.8157 - accuracy: 0.7312 - val_loss: 0.9037 - val_accuracy: 0.6812

Epoch 16/50
15/15 [=====] - 9s 634ms/step - loss: 0.7680 - accuracy: 0.7333 - val_loss: 0.8221 - val_accuracy: 0.7563

Epoch 17/50
15/15 [=====] - 9s 641ms/step - loss: 0.7592 - accuracy: 0.7479 - val_loss: 0.7181 - val_accuracy: 0.7750

Epoch 18/50
15/15 [=====] - 9s 634ms/step - loss: 0.7068 - accuracy: 0.7708 - val_loss: 0.7023 - val_accuracy: 0.7500

Epoch 19/50
15/15 [=====] - 9s 642ms/step - loss: 0.6843 - accuracy: 0.7833 - val_loss: 0.6944 - val_accuracy: 0.7750

Epoch 20/50
15/15 [=====] - 9s 632ms/step - loss: 0.6444 - accuracy: 0.8042 - val_loss: 0.7477 - val_accuracy: 0.7312

Epoch 21/50
15/15 [=====] - 9s 638ms/step - loss: 0.6837 - accuracy: 0.7722 - val_loss: 0.7101 - val_accuracy: 0.7500

Epoch 22/50
15/15 [=====] - 9s 635ms/step - loss: 0.6600 - accuracy: 0.7937 - val_loss: 0.7114 - val_accuracy: 0.7625

Epoch 23/50
15/15 [=====] - 9s 634ms/step - loss: 0.7580 - accuracy: 0.7437 - val_loss: 0.6265 - val_accuracy: 0.8250

Epoch 24/50
15/15 [=====] - 10s 657ms/step - loss: 0.6965 - accuracy: 0.7937 - val_loss: 0.7808 - val_accuracy: 0.7625

Epoch 25/50
15/15 [=====] - 9s 629ms/step - loss: 0.6485 - accuracy: 0.8042 - val_loss: 0.5483 - val_accuracy: 0.8438

Epoch 26/50
15/15 [=====] - 9s 629ms/step - loss: 0.6761 - accuracy: 0.7792 - val_loss: 0.7192 - val_accuracy: 0.7812

Epoch 27/50
15/15 [=====] - 10s 669ms/step - loss: 0.6264 - accuracy: 0.8000 - val_loss: 0.6566 - val_accuracy: 0.8000

Epoch 28/50
15/15 [=====] - 9s 627ms/step - loss: 0.6426 - accuracy: 0.7688 - val_loss: 0.6754 - val_accuracy: 0.7625

Epoch 29/50
15/15 [=====] - 9s 637ms/step - loss: 0.6012 - accuracy: 0.8000 - val_loss: 0.8215 - val_accuracy: 0.7500

Epoch 30/50
15/15 [=====] - 10s 651ms/step - loss: 0.6088 - accuracy: 0.8292 - val_loss: 0.7165 - val_accuracy: 0.7875

Epoch 31/50
15/15 [=====] - 9s 634ms/step - loss: 0.6331 - accuracy: 0.7917 - val_loss: 0.6029 - val_accuracy: 0.7688

Epoch 32/50
15/15 [=====] - 10s 645ms/step - loss: 0.5939 - accuracy: 0.8021 - val_loss: 0.7278 - val_accuracy: 0.7563

Epoch 33/50
15/15 [=====] - 9s 631ms/step - loss: 0.5579 - accuracy: 0.8104 - val_loss: 0.5333 - val_accuracy: 0.8125

Epoch 34/50
15/15 [=====] - 10s 644ms/step - loss: 0.6208 - accuracy: 0.7937 - val_loss: 0.6363 - val_accuracy: 0.7688

Epoch 35/50
15/15 [=====] - 9s 632ms/step - loss: 0.6775 - accuracy: 0.7708 - val_loss: 0.5711 - val_accuracy: 0.8313

Epoch 36/50
15/15 [=====] - 9s 632ms/step - loss: 0.5770 - accuracy: 0.8167 - val_loss: 0.6399 - val_accuracy: 0.7875

Epoch 37/50
15/15 [=====] - 9s 640ms/step - loss: 0.5655 - accuracy: 0.8104 - val_loss: 0.5913 - val_accuracy: 0.8375

Epoch 38/50
15/15 [=====] - 9s 631ms/step - loss: 0.5752 - accuracy: 0.8125 - val_loss: 0.5958 - val_accuracy: 0.8188

Epoch 39/50
15/15 [=====] - 9s 639ms/step - loss: 0.5477 - accuracy: 0.8333 - val_loss: 0.5630 - val_accuracy: 0.8250

Epoch 40/50
15/15 [=====] - 10s 658ms/step - loss: 0.4892 - accuracy: 0.8375 - val_loss: 0.5758 - val_accuracy: 0.7937

Epoch 41/50
15/15 [=====] - 10s 645ms/step - loss: 0.5548 - accuracy: 0.8250 - val_loss: 0.5366 - val_accuracy: 0.7937

Epoch 42/50
15/15 [=====] - 9s 631ms/step - loss: 0.6246 - accuracy: 0.7979 - val_loss: 0.5535 - val_accuracy: 0.8125

Epoch 43/50
15/15 [=====] - 9s 621ms/step - loss: 0.5788 - accuracy: 0.8080 - val_loss: 0.7116 - val_accuracy: 0.7688

Epoch 44/50
15/15 [=====] - 9s 632ms/step - loss: 0.5522 - accuracy: 0.8125 - val_loss: 0.4770 - val_accuracy: 0.8687

Epoch 45/50
15/15 [=====] - 9s 629ms/step - loss: 0.4644 - accuracy: 0.8458 - val_loss: 0.5166 - val_accuracy: 0.8500

Epoch 46/50
15/15 [=====] - 9s 636ms/step - loss: 0.5928 - accuracy: 0.7896 - val_loss: 0.5467 - val_accuracy: 0.8438

Epoch 47/50
15/15 [=====] - 9s 641ms/step - loss: 0.5857 - accuracy: 0.8000 - val_loss: 0.6648 - val_accuracy: 0.8500

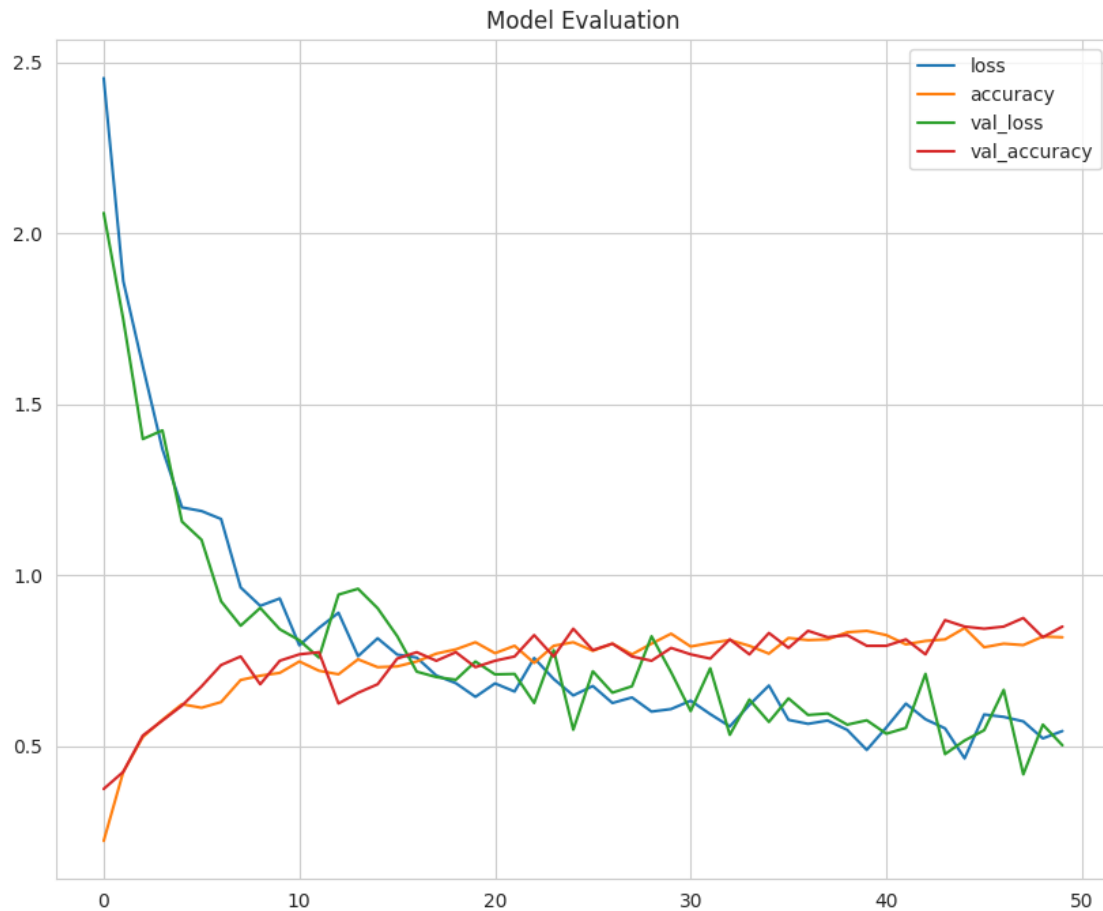
Epoch 48/50
15/15 [=====] - 9s 633ms/step - loss: 0.5727 - accuracy: 0.7958 - val_loss: 0.4178 - val_accuracy: 0.8750

Epoch 49/50
15/15 [=====] - 9s 625ms/step - loss: 0.5229 - accuracy: 0.8208 - val_loss: 0.5631 - val_accuracy: 0.8188

Epoch 50/50
15/15 [=====] - 9s 638ms/step - loss: 0.5445 - accuracy: 0.8188 - val_loss: 0.5026 - val_accuracy: 0.8500

Model Building Finished
Model was saved

```
[24]: evaluation_plot(inc_history.history)
```



Custom CNN model

```
[1]: import os
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

/opt/conda/lib/python3.10/site-packages/scipy/__init__.py:146: UserWarning: A

NumPy version $\geq 1.16.5$ and $< 1.23.0$ is required for this version of SciPy
(detected version 1.24.3
warnings.warn(f"A NumPy version $\geq \{np_minversion\}$ and $< \{np_maxversion\}$ ")

```
[2]: # Set the path to your dataset
DATA_PATH = '../input/plantdisease/PlantVillage'
```

```
[3]: # Define image dimensions and other parameters
IMG_SHAPE = (224, 224)
INPUT_SHAPE = (224, 224, 3)
EPOCHS = 50
BS = 32
```

```
[4]: # Create data generators
img_data_gen = ImageDataGenerator(rescale=1./255, rotation_range=0.2,
    ↪horizontal_flip=True,
                                vertical_flip=True, shear_range=0.2,
    ↪validation_split=0.25)

train_data_gen = img_data_gen.flow_from_directory(DATA_PATH, batch_size=BS,
    ↪subset='training',
                                class_mode='categorical',
    ↪shuffle=True,
                                target_size=IMG_SHAPE)

val_data_gen = img_data_gen.flow_from_directory(DATA_PATH, batch_size=BS,
    ↪subset='validation',
                                class_mode='categorical',
    ↪shuffle=True,
                                target_size=IMG_SHAPE)
```

Found 15482 images belonging to 15 classes.

Found 5156 images belonging to 15 classes.

```
[5]: # Define and compile the custom CNN model
def custom_cnn_model(input_shape=INPUT_SHAPE):
    model = Sequential()
    model.add(Conv2D(32, (3, 3), activation='relu', input_shape=input_shape))
    model.add(MaxPooling2D((2, 2)))
    model.add(Conv2D(64, (3, 3), activation='relu'))
    model.add(MaxPooling2D((2, 2)))
    model.add(Conv2D(128, (3, 3), activation='relu'))
    model.add(MaxPooling2D((2, 2)))
    model.add(Flatten())
    model.add(Dense(128, activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(15, activation='sigmoid'))
```

```

    model.compile(optimizer=Adam(), loss='categorical_crossentropy',
metrics=['accuracy'])
    return model

```

```

[6]: # Train the custom CNN model
custom_model = custom_cnn_model()
custom_history = custom_model.fit(train_data_gen, validation_data=val_data_gen,
                                validation_steps=len(val_data_gen)//BS,
                                steps_per_epoch=len(train_data_gen)//BS,
                                batch_size=BS, epochs=EPOCHS)

```

```

Epoch 1/50
15/15 [=====] - 20s 767ms/step - loss: 3.0831 -
accuracy: 0.0979 - val_loss: 2.6111 - val_accuracy: 0.1000
Epoch 2/50
15/15 [=====] - 12s 783ms/step - loss: 2.6459 -
accuracy: 0.1083 - val_loss: 2.6246 - val_accuracy: 0.1625
Epoch 3/50
15/15 [=====] - 11s 747ms/step - loss: 2.6469 -
accuracy: 0.1104 - val_loss: 2.5676 - val_accuracy: 0.1125
Epoch 4/50
15/15 [=====] - 13s 884ms/step - loss: 2.5958 -
accuracy: 0.1542 - val_loss: 2.4928 - val_accuracy: 0.2125
Epoch 5/50
15/15 [=====] - 11s 738ms/step - loss: 2.3982 -
accuracy: 0.2188 - val_loss: 2.2037 - val_accuracy: 0.2750
Epoch 6/50
15/15 [=====] - 11s 750ms/step - loss: 2.4061 -
accuracy: 0.2521 - val_loss: 2.2416 - val_accuracy: 0.3562
Epoch 7/50
15/15 [=====] - 10s 702ms/step - loss: 2.2812 -
accuracy: 0.2729 - val_loss: 2.1194 - val_accuracy: 0.3500
Epoch 8/50
15/15 [=====] - 11s 719ms/step - loss: 2.1160 -
accuracy: 0.3396 - val_loss: 1.9522 - val_accuracy: 0.4187
Epoch 9/50
15/15 [=====] - 10s 701ms/step - loss: 2.0677 -
accuracy: 0.3833 - val_loss: 1.7979 - val_accuracy: 0.4313
Epoch 10/50
15/15 [=====] - 10s 696ms/step - loss: 1.9491 -
accuracy: 0.4167 - val_loss: 1.7276 - val_accuracy: 0.5312
Epoch 11/50
15/15 [=====] - 12s 839ms/step - loss: 1.8949 -
accuracy: 0.4313 - val_loss: 1.7519 - val_accuracy: 0.4375
Epoch 12/50
15/15 [=====] - 10s 662ms/step - loss: 1.7873 -

```


accuracy: 0.4354 - val_loss: 1.5831 - val_accuracy: 0.5125
 Epoch 13/50
 15/15 [=====] - 10s 673ms/step - loss: 1.7892 -
 accuracy: 0.4125 - val_loss: 1.6103 - val_accuracy: 0.5750
 Epoch 14/50
 15/15 [=====] - 10s 669ms/step - loss: 1.7889 -
 accuracy: 0.4271 - val_loss: 1.5864 - val_accuracy: 0.5125
 Epoch 15/50
 15/15 [=====] - 9s 637ms/step - loss: 1.7616 -
 accuracy: 0.4167 - val_loss: 1.5090 - val_accuracy: 0.5188
 Epoch 16/50
 15/15 [=====] - 10s 657ms/step - loss: 1.7239 -
 accuracy: 0.4479 - val_loss: 1.4001 - val_accuracy: 0.5813
 Epoch 17/50
 15/15 [=====] - 9s 635ms/step - loss: 1.7264 -
 accuracy: 0.4396 - val_loss: 1.4430 - val_accuracy: 0.5437
 Epoch 18/50
 15/15 [=====] - 10s 656ms/step - loss: 1.8682 -
 accuracy: 0.4021 - val_loss: 1.4888 - val_accuracy: 0.5375
 Epoch 19/50
 15/15 [=====] - 9s 631ms/step - loss: 1.7362 -
 accuracy: 0.4604 - val_loss: 1.5013 - val_accuracy: 0.5125
 Epoch 20/50
 15/15 [=====] - 9s 630ms/step - loss: 1.6942 -
 accuracy: 0.4563 - val_loss: 1.2944 - val_accuracy: 0.6000
 Epoch 21/50
 15/15 [=====] - 9s 623ms/step - loss: 1.6214 -
 accuracy: 0.4979 - val_loss: 1.3394 - val_accuracy: 0.5813
 Epoch 22/50
 15/15 [=====] - 9s 629ms/step - loss: 1.5583 -
 accuracy: 0.4833 - val_loss: 1.2369 - val_accuracy: 0.6562
 Epoch 23/50
 15/15 [=====] - 9s 613ms/step - loss: 1.5915 -
 accuracy: 0.4854 - val_loss: 1.2576 - val_accuracy: 0.6187
 Epoch 24/50
 15/15 [=====] - 9s 605ms/step - loss: 1.4655 -
 accuracy: 0.5542 - val_loss: 1.1766 - val_accuracy: 0.6250
 Epoch 25/50
 15/15 [=====] - 9s 598ms/step - loss: 1.5210 -
 accuracy: 0.5042 - val_loss: 1.3705 - val_accuracy: 0.5188
 Epoch 26/50
 15/15 [=====] - 9s 599ms/step - loss: 1.4767 -
 accuracy: 0.5417 - val_loss: 1.2502 - val_accuracy: 0.5938
 Epoch 27/50
 15/15 [=====] - 9s 587ms/step - loss: 1.4938 -
 accuracy: 0.5250 - val_loss: 1.2080 - val_accuracy: 0.5938
 Epoch 28/50
 15/15 [=====] - 9s 630ms/step - loss: 1.3830 -

accuracy: 0.5667 - val_loss: 1.0708 - val_accuracy: 0.6812
 Epoch 29/50
 15/15 [=====] - 9s 599ms/step - loss: 1.4360 -
 accuracy: 0.5500 - val_loss: 1.1830 - val_accuracy: 0.6062
 Epoch 30/50
 15/15 [=====] - 9s 625ms/step - loss: 1.4258 -
 accuracy: 0.5417 - val_loss: 1.2099 - val_accuracy: 0.6187
 Epoch 31/50
 15/15 [=====] - 8s 566ms/step - loss: 1.4774 -
 accuracy: 0.5104 - val_loss: 1.1276 - val_accuracy: 0.6438
 Epoch 32/50
 15/15 [=====] - 10s 662ms/step - loss: 1.3788 -
 accuracy: 0.5375 - val_loss: 1.0424 - val_accuracy: 0.6250
 Epoch 33/50
 15/15 [=====] - 8s 565ms/step - loss: 1.3661 -
 accuracy: 0.5437 - val_loss: 0.9643 - val_accuracy: 0.6687
 Epoch 34/50
 15/15 [=====] - 8s 576ms/step - loss: 1.5027 -
 accuracy: 0.5375 - val_loss: 1.0816 - val_accuracy: 0.6625
 Epoch 35/50
 15/15 [=====] - 9s 617ms/step - loss: 1.4415 -
 accuracy: 0.5250 - val_loss: 1.2852 - val_accuracy: 0.5688
 Epoch 36/50
 15/15 [=====] - 9s 619ms/step - loss: 1.3056 -
 accuracy: 0.5688 - val_loss: 1.0786 - val_accuracy: 0.7125
 Epoch 37/50
 15/15 [=====] - 9s 583ms/step - loss: 1.4542 -
 accuracy: 0.5125 - val_loss: 1.0864 - val_accuracy: 0.6438
 Epoch 38/50
 15/15 [=====] - 9s 581ms/step - loss: 1.2930 -
 accuracy: 0.5854 - val_loss: 1.1641 - val_accuracy: 0.6125
 Epoch 39/50
 15/15 [=====] - 9s 587ms/step - loss: 1.2778 -
 accuracy: 0.6000 - val_loss: 0.8779 - val_accuracy: 0.6938
 Epoch 40/50
 15/15 [=====] - 9s 582ms/step - loss: 1.3245 -
 accuracy: 0.5708 - val_loss: 1.0039 - val_accuracy: 0.6187
 Epoch 41/50
 15/15 [=====] - 9s 600ms/step - loss: 1.1876 -
 accuracy: 0.5896 - val_loss: 1.0016 - val_accuracy: 0.6625
 Epoch 42/50
 15/15 [=====] - 9s 597ms/step - loss: 1.2433 -
 accuracy: 0.5854 - val_loss: 0.9973 - val_accuracy: 0.6625
 Epoch 43/50
 15/15 [=====] - 8s 566ms/step - loss: 1.4139 -
 accuracy: 0.5312 - val_loss: 0.9658 - val_accuracy: 0.6562
 Epoch 44/50
 15/15 [=====] - 8s 550ms/step - loss: 1.4006 -

```

accuracy: 0.5312 - val_loss: 1.1332 - val_accuracy: 0.6625
Epoch 45/50
15/15 [=====] - 9s 594ms/step - loss: 1.3958 -
accuracy: 0.5417 - val_loss: 1.0671 - val_accuracy: 0.7188
Epoch 46/50
15/15 [=====] - 8s 561ms/step - loss: 1.3121 -
accuracy: 0.5708 - val_loss: 0.9558 - val_accuracy: 0.7000
Epoch 47/50
15/15 [=====] - 8s 568ms/step - loss: 1.2333 -
accuracy: 0.6104 - val_loss: 0.8406 - val_accuracy: 0.7375
Epoch 48/50
15/15 [=====] - 8s 551ms/step - loss: 1.2288 -
accuracy: 0.5604 - val_loss: 1.0217 - val_accuracy: 0.6500
Epoch 49/50
15/15 [=====] - 8s 561ms/step - loss: 1.2248 -
accuracy: 0.5833 - val_loss: 1.0180 - val_accuracy: 0.6500
Epoch 50/50
15/15 [=====] - 8s 549ms/step - loss: 1.2494 -
accuracy: 0.5729 - val_loss: 0.9060 - val_accuracy: 0.7375

```

```

[8]: # Save the trained model
model_save_path = 'saved_model/custom_cnn_model.h5'
custom_model.save(model_save_path)
print(f"Custom CNN model saved at: {model_save_path}")

```

```

/opt/conda/lib/python3.10/site-packages/keras/src/engine/training.py:3000:
UserWarning: You are saving your model as an HDF5 file via `model.save()`. This
file format is considered legacy. We recommend using instead the native Keras
format, e.g. `model.save('my_model.keras')`.

```

```

    saving_api.save_model(

```

```

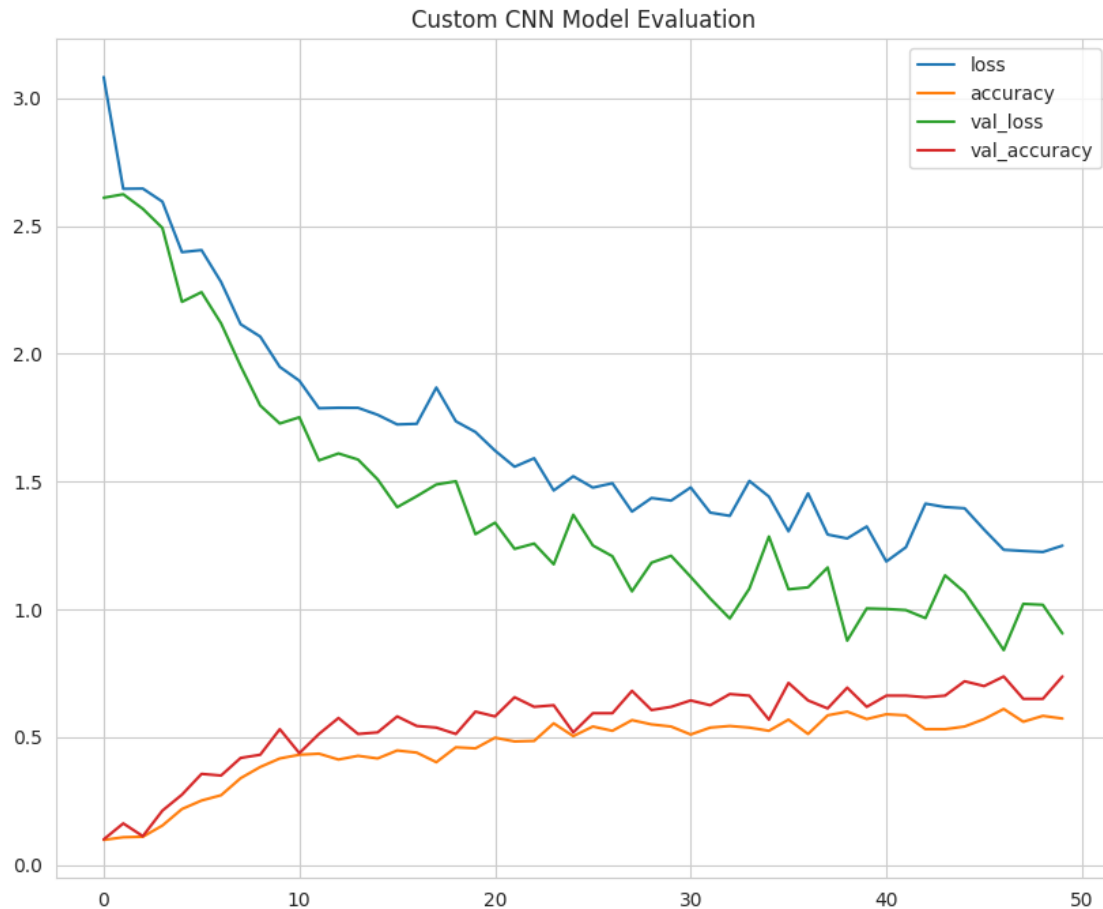
Custom CNN model saved at: saved_model/custom_cnn_model.h5

```

```

[7]: # Plot the evaluation metrics
sns.set_style('whitegrid')
plt.figure(figsize=(10, 8))
plt.plot(custom_history.history['loss'], label='loss')
plt.plot(custom_history.history['accuracy'], label='accuracy')
plt.plot(custom_history.history['val_loss'], label='val_loss')
plt.plot(custom_history.history['val_accuracy'], label='val_accuracy')
plt.legend()
plt.title('Custom CNN Model Evaluation')
plt.show()

```



Evaluation

```
[28]: from sklearn.metrics import classification_report, confusion_matrix
from sklearn.metrics import accuracy_score, f1_score, precision_score,
      ↪ recall_score
import seaborn as sns
```

```
[32]: # Load VGG16 model
vgg16_model = load_model('/kaggle/working/saved_model/<function VGG16 at_
      ↪ 0x7e0346c37f40>_1.h5') # Update with the correct path

# Load InceptionV3 model
inception_model = load_model('/kaggle/working/saved_model/<function InceptionV3_
      ↪ at 0x7e0346c34c10>_1.h5') # Update with the correct path

# Load Custom CNN model
custom_model = load_model('/kaggle/working/saved_model/custom_cnn_model.h5') #_
      ↪ Update with the correct path
```

```
[33]: def evaluate_model(model, data_gen):
    # Generate predictions
    predictions = model.predict(data_gen)
    y_pred = np.argmax(predictions, axis=1)
    y_true = data_gen.classes

    # Accuracy
    accuracy = accuracy_score(y_true, y_pred)
    print(f'Accuracy: {accuracy:.4f}')

    # F1 Score, Precision, Recall
    f1 = f1_score(y_true, y_pred, average='weighted')
    precision = precision_score(y_true, y_pred, average='weighted')
    recall = recall_score(y_true, y_pred, average='weighted')

    print(f'F1 Score: {f1:.4f}')
    print(f'Precision: {precision:.4f}')
    print(f'Recall: {recall:.4f}')

    # Confusion Matrix
    cm = confusion_matrix(y_true, y_pred)
    plt.figure(figsize=(10, 8))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', cbar=False,
                xticklabels=data_gen.class_indices.keys(),
                yticklabels=data_gen.class_indices.keys())
    plt.title('Confusion Matrix')
    plt.xlabel('Predicted')
    plt.ylabel('True')
    plt.show()
```

```
[34]: # Evaluate VGG16
vgg16_eval = vgg16_model.evaluate(val_data_gen)
print("Evaluation on VGG16 model:")
print(f"Loss: {vgg16_eval[0]:.4f}")
print(f"Accuracy: {vgg16_eval[1]:.4f}")

evaluate_model(vgg16_model, val_data_gen)
```

```
162/162 [=====] - 73s 452ms/step - loss: 37.0300 -
accuracy: 0.7915
Evaluation on VGG16 model:
Loss: 37.0300
Accuracy: 0.7915
162/162 [=====] - 67s 413ms/step
Accuracy: 0.0844
F1 Score: 0.0837
Precision: 0.0841
```

Recall: 0.0844

		Confusion Matrix														
True	Pepper_bell__Bacterial_spot	12	12	15	9	2	22	5	24	7	18	19	32	44	2	26
	Pepper_bell__healthy	26	21	21	22	3	34	12	37	10	27	32	38	55	7	24
	Potato__Early_blight	16	21	16	6	4	19	9	25	10	21	19	28	34	2	20
	Potato__Late_blight	14	21	12	11	0	23	9	25	12	22	14	22	43	4	18
	Potato__healthy	0	2	1	4	0	2	3	4	1	8	3	3	2	0	5
	Tomato_Bacterial_spot	28	32	25	16	2	60	20	67	15	41	38	51	85	7	44
	Tomato_Early_blight	14	16	24	16	1	19	7	22	11	11	23	28	40	6	12
	Tomato_Late_blight	22	36	23	23	0	46	15	49	14	31	36	54	68	12	48
	Tomato_Leaf_Mold	13	16	13	11	1	26	9	26	10	22	10	22	35	4	20
	Tomato_Septoria_leaf_spot	32	33	23	29	1	36	11	52	8	27	28	50	77	6	29
	Tomato_Spider_mites_Two_spotted_spider_mite	20	28	16	26	0	30	12	56	16	31	41	41	71	5	26
	Tomato_Target_Spot	17	23	12	19	0	36	12	40	16	16	26	36	66	1	31
	Tomato_Tomato_YellowLeaf_Curl_Virus	36	53	47	36	2	74	24	85	31	63	59	100	114	11	67
	Tomato_Tomato_mosaic_virus	3	9	3	8	0	7	3	7	2	7	10	11	10	3	10
	Tomato_healthy	29	18	21	28	1	42	13	38	13	22	31	28	75	10	28
	Pepper_bell__Bacterial_spot	Pepper_bell__healthy	Potato__Early_blight	Potato__Late_blight	Potato__healthy	Tomato_Bacterial_spot	Tomato_Early_blight	Tomato_Late_blight	Tomato_Leaf_Mold	Tomato_Septoria_leaf_spot	Tomato_Spider_mites_Two_spotted_spider_mite	Tomato_Target_Spot	Tomato_Tomato_YellowLeaf_Curl_Virus	Tomato_Tomato_mosaic_virus	Tomato_healthy	
	Predicted															

```
[35]: # Evaluate InceptionV3
inc_eval = inception_model.evaluate(val_data_gen)
print("\nEvaluation on InceptionV3 model:")
print(f"Loss: {inc_eval[0]:.4f}")
print(f"Accuracy: {inc_eval[1]:.4f}")

evaluate_model(inception_model, val_data_gen)
```

162/162 [=====] - 72s 434ms/step - loss: 0.5612 - accuracy: 0.8161

Evaluation on InceptionV3 model:

Loss: 0.5612

Accuracy: 0.8161

162/162 [=====] - 68s 415ms/step

Accuracy: 0.0820
 F1 Score: 0.0801
 Precision: 0.0792
 Recall: 0.0820

Confusion Matrix

True	Pepper__bell__Bacterial_spot	15	17	10	21	4	18	3	12	16	28	16	27	38	5	19
	Pepper__bell__healthy	28	15	22	14	5	40	11	41	15	39	21	29	59	9	21
	Potato__Early_blight	12	22	13	10	4	23	5	15	18	23	22	17	48	2	16
	Potato__Late_blight	16	12	10	10	1	40	3	14	12	31	14	24	46	2	15
	Potato__healthy	2	3	4	4	1	5	0	1	4	1	4	3	5	1	0
	Tomato_Bacterial_spot	32	38	29	24	6	54	10	30	26	46	52	42	97	13	32
	Tomato_Early_blight	13	18	9	17	0	28	1	23	15	21	11	22	53	3	16
	Tomato_Late_blight	24	18	19	33	4	62	9	32	40	49	25	34	85	11	32
	Tomato_Leaf_Mold	12	18	18	9	1	23	6	20	9	26	18	16	47	3	12
	Tomato_Septoria_leaf_spot	31	22	23	21	2	53	3	44	18	50	25	29	77	13	31
	Tomato_Spider_mites_Two_spotted_spider_mite	27	25	22	31	4	56	4	34	22	35	27	29	66	11	26
	Tomato__Target_Spot	20	28	17	21	2	32	8	33	22	31	19	26	70	9	13
	Tomato__Tomato_YellowLeaf__Curl_Virus	50	47	46	34	2	77	14	75	48	81	41	74	142	9	62
	Tomato__Tomato_mosaic_virus	5	5	2	2	0	5	1	18	5	10	6	10	14	2	8
	Tomato_healthy	22	32	13	36	1	52	6	26	25	38	23	25	59	13	26
		Pepper__bell__Bacterial_spot	Pepper__bell__healthy	Potato__Early_blight	Potato__Late_blight	Potato__healthy	Tomato_Bacterial_spot	Tomato_Early_blight	Tomato_Late_blight	Tomato_Leaf_Mold	Tomato_Septoria_leaf_spot	Tomato_Spider_mites_Two_spotted_spider_mite	Tomato__Target_Spot	Tomato__Tomato_YellowLeaf__Curl_Virus	Tomato__Tomato_mosaic_virus	Tomato_healthy
		Predicted														

```
[1]: # Evaluate Custom CNN
# custom_eval = custom_model.evaluate(val_data_gen)
print("\nEvaluation on Custom CNN model:")
print(f"Loss: {custom_history.history['loss']:.4f}")
print(f"Accuracy: {custom_history.history['accuracy']:.4f}")

# evaluate_model(custom_model, val_data_gen)
```

Evaluation on Custom CNN model:

```
NameError                                Traceback (most recent call last)
Cell In[1], line 4
      1 # Evaluate Custom CNN
      2 # custom_eval = custom_model.evaluate(val_data_gen)
      3 print("\nEvaluation on Custom CNN model:")
----> 4 print(f"Loss: {custom_history.history['loss']:.4f}")
      5 print(f"Accuracy: {custom_history.history['accuracy']:.4f}")
      7 # evaluate_model(custom_model, val_data_gen)

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

[]: