```
import os

def total_files(folder_path):
    num_files = len([f for f in os.listdir(folder_path) if
os.path.isfile(os.path.join(folder_path, f))])
    return num_files

train_files_healthy = "Dataset/Train/Train/Healthy"
train_files_powdery = "Dataset/Train/Train/Powdery"
train_files_rust = "Dataset/Train/Train/Rust"

test_files_healthy = "Dataset/Test/Test/Healthy"
test_files_powdery = "Dataset/Test/Test/Powdery"
test_files_rust = "Dataset/Test/Test/Rust"

valid_files_healthy = "Dataset/Validation/Validation/Healthy"
valid_files_powdery = "Dataset/Validation/Validation/Powdery"
valid_files_rust = "Dataset/Validation/Validation/Rust"

print("Number of healthy leaf images in training set",
total_files(train_files_healthy))
print("Number of powder leaf images in training set",
total_files(train_files_powdery))
print("Number of rusty leaf images in training set",
total_files(train_files_rust))

print("==========================================================")

print("Number of healthy leaf images in test set",
total_files(test_files_healthy))
print("Number of powder leaf images in test set",
total_files(test_files_powdery))
print("Number of rusty leaf images in test set",
total_files(test_files_rust))

print("==========================================================")

print("Number of healthy leaf images in validation set",
total_files(valid_files_healthy))
print("Number of powder leaf images in validation set",
total_files(valid_files_powdery))
print("Number of rusty leaf images in validation set",
total_files(valid_files_rust))
```

```
Number of healthy leaf images in training set 458
Number of powder leaf images in training set 430
Number of rusty leaf images in training set 434
==========================================================
Number of healthy leaf images in test set 50
Number of powder leaf images in test set 50
Number of rusty leaf images in test set 50
==========================================================
Number of healthy leaf images in validation set 20
Number of powder leaf images in validation set 20
Number of rusty leaf images in validation set 20
```

```
from PIL import Image
import IPython.display as display

image_path = 'Dataset/Train/Train/Healthy/8ce77048e12f3dd4.jpg'

with open(image_path, 'rb') as f:
    display.display(display.Image(data=f.read(), width=500))
```

```
image_path = 'Dataset/Train/Train/Rust/80f09587dfc7988e.jpg'

with open(image_path, 'rb') as f:
    display.display(display.Image(data=f.read(), width=500))
```

```
from keras.preprocessing.image import ImageDataGenerator
```

```
train_datagen = ImageDataGenerator(rescale=1./255, shear_range=0.2,
zoom_range=0.2, horizontal_flip=True)
test_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory('Dataset/Train/Train',
                                                    target_size=(225, 225),
                                                    batch_size=32,

class_mode='categorical')

validation_generator =
test_datagen.flow_from_directory('Dataset/Validation/Validation',
                                                         target_size=(225,
225),
                                                         batch_size=32,

class_mode='categorical')
```

```
Found 1322 images belonging to 3 classes.
Found 60 images belonging to 3 classes.
```

```
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense


model = Sequential()
model.add(Conv2D(32, (3, 3), input_shape=(225, 225, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(64, activation='relu'))
model.add(Dense(3, activation='softmax'))

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

history = model.fit(train_generator,
                    batch_size=16,
                    epochs=5,
                    validation_data=validation_generator,
                    validation_batch_size=16
                    )
```

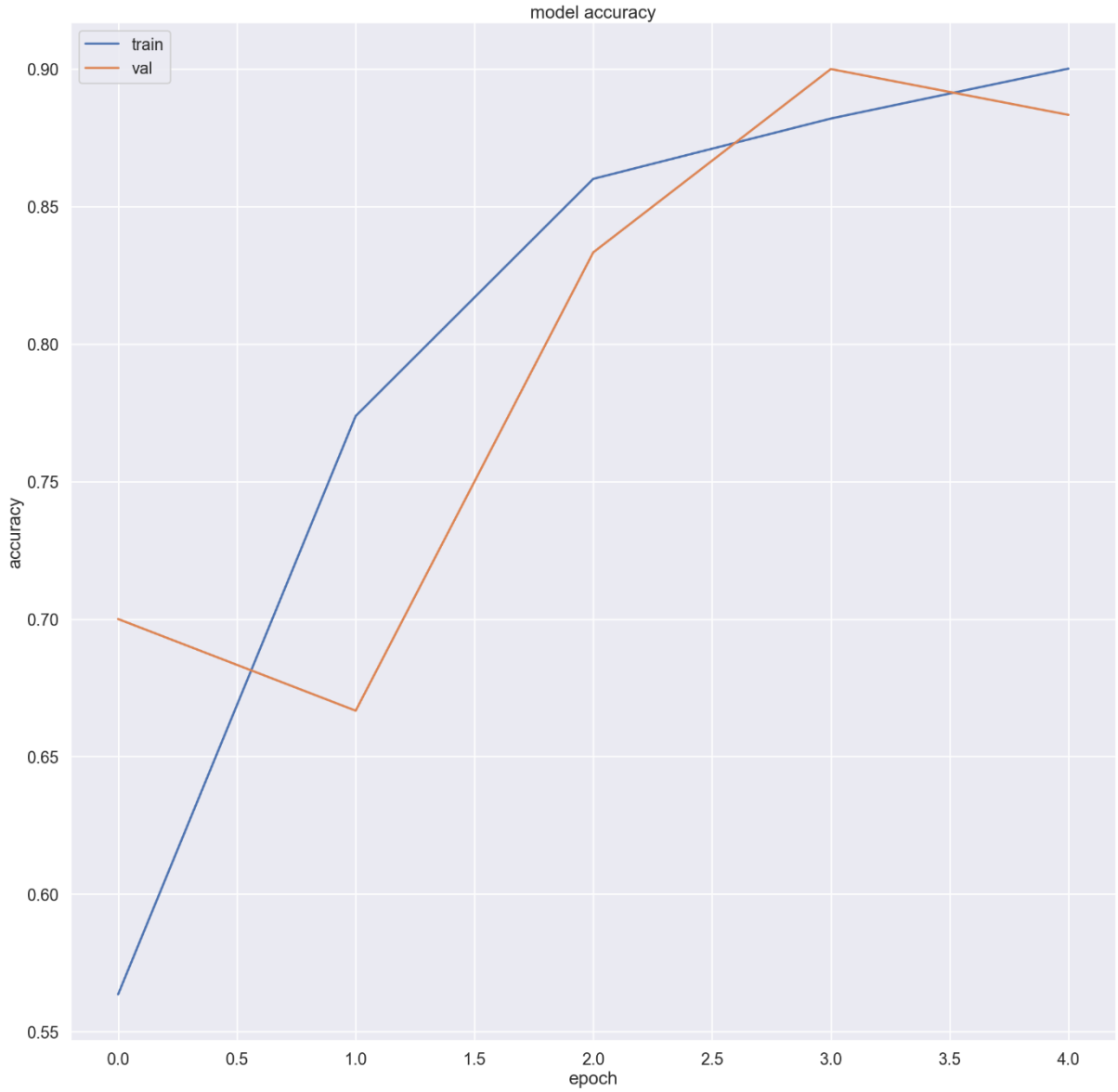
```
Epoch 1/5
42/42 [==============================] - 223s 5s/step - loss: 1.2323 - accu
racy: 0.5635 - val_loss: 0.6878 - val_accuracy: 0.7000
Epoch 2/5
42/42 [==============================] - 174s 4s/step - loss: 0.5399 - accu
racy: 0.7738 - val_loss: 0.7110 - val_accuracy: 0.6667
Epoch 3/5
42/42 [==============================] - 175s 4s/step - loss: 0.3664 - accu
racy: 0.8601 - val_loss: 0.4629 - val_accuracy: 0.8333
Epoch 4/5
42/42 [==============================] - 175s 4s/step - loss: 0.3033 - accu
racy: 0.8820 - val_loss: 0.3759 - val_accuracy: 0.9000
Epoch 5/5
42/42 [==============================] - 101s 2s/step - loss: 0.3112 - accu
racy: 0.9002 - val_loss: 0.3953 - val_accuracy: 0.8833
```

```
from matplotlib import pyplot as plt
from matplotlib.pyplot import figure

import seaborn as sns
sns.set_theme()
sns.set_context("poster")

figure(figsize=(25, 25), dpi=100)

plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```




```
model.save("model.h5")
```

```
from tensorflow.keras.preprocessing.image import load_img, img_to_array
import numpy as np
```

```
def preprocess_image(image_path, target_size=(225, 225)):
    img = load_img(image_path, target_size=target_size)
    x = img_to_array(img)
    x = x.astype('float32') / 255.
    x = np.expand_dims(x, axis=0)
    return x

x = preprocess_image('Dataset/Test/Test/Rust/82f49a4a7b9585f1.jpg')
predictions = model.predict(x)
predictions[0]
```

```
array([2.5705326e-01, 2.5312374e-05, 7.4292141e-01], dtype=float32)
```

```
labels = train_generator.class_indices
labels = {v: k for k, v in labels.items()}
labels
```

Out:

```
{0: 'Healthy', 1: 'Powdery', 2: 'Rust'}
```

```
predicted_label = labels[np.argmax(predictions)]
print(predicted_label)
```

Out:

```
Rust
```