rice-disease-classifier-2

March 1, 2024

[1]:
```
from __future__ import absolute_import, print_function,division,␣
 ↪unicode_literals

import os.path
import glob
import shutil

import tensorflow as tf
assert tf.__version__.startswith('2')

from tensorflow import keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.layers import␣
 ↪Conv2D,Flatten,MaxPooling2D,Dropout,Dense,Activation
from keras import regularizers
import keras

import numpy as np
import matplotlib.pyplot as plt
import pathlib

print ('successful')
```

```
successful
```

[2]:
```
print(tf.__version__)
```

```
2.4.1
```

[3]:
```
BATCH_SIZE = 8
IMG_HEIGHT = 224
IMG_WIDTH = 224
```

[4]:
```
data_dir ="../input/rice-leaf-diseases/rice_leaf_diseases"
data_dir = pathlib.Path(data_dir)

CLASS_NAMES = np.array(['Leaf Blight','Brown Spot','Leaf Smut'])
```

```
print('Class Names: ', CLASS_NAMES)
```

```
Class Names:  ['Leaf Blight' 'Brown Spot' 'Leaf Smut']
```

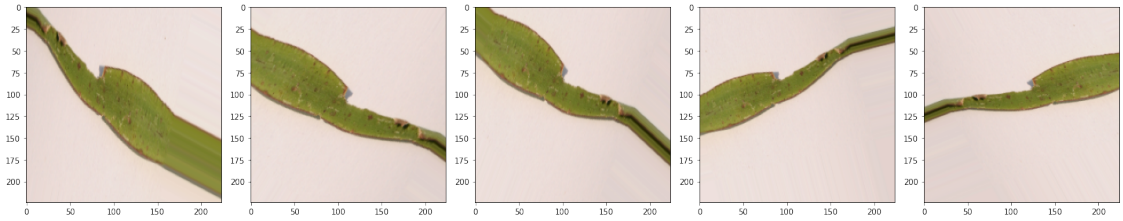
[5]:
```
train_path = '../input/rice-leaf-diseases/rice_leaf_diseases'
test_path = '../input/rice-leaf-diseases/rice_leaf_diseases'
```

[6]:
```
image_train_gen = ImageDataGenerator(rescale=1./255,
                                     zoom_range=0.50,
                                     rotation_range=45,
                                     horizontal_flip=True,
                                     width_shift_range=0.15,
                                     height_shift_range=0.15)

train_data_gen = image_train_gen.flow_from_directory(train_path,
                                                     shuffle=True,
                                                     batch_size=BATCH_SIZE,
                                                    ␣
 ↪target_size=(IMG_HEIGHT,IMG_WIDTH),
                                                     class_mode='sparse')

img_val_gen = ImageDataGenerator(rescale=1./255)
val_data_gen = img_val_gen.flow_from_directory(test_path,
                                               batch_size=BATCH_SIZE,
                                              ␣
 ↪target_size=(IMG_HEIGHT,IMG_WIDTH),
                                               class_mode='sparse')
```

```
Found 120 images belonging to 3 classes.
Found 120 images belonging to 3 classes.
```

[7]:
```
def plotImages(image_arr):
    fig,axes = plt.subplots(1, 5, figsize=(20,20))
    axes = axes.flatten()
    for img,ax in zip(image_arr,axes):
        ax.imshow(img)
    plt.tight_layout()
    plt.show()
```

[8]:
```
# Plot a few training images
img_array = [train_data_gen[0][0][0] for i in range(5)]
plotImages(img_array)
```



[9]:
```
# plot a few val images
img_array  = [val_data_gen[0][0][0] for i in range(5)]
plotImages(img_array)
```

[10]:
```
# Model building
#Instatiating A convnet

model = Sequential()
model.add(Conv2D(16, (3,3), input_shape=(224,224,3), activation="relu"))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Conv2D(32, (3,3), 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(Dropout(0.2))
model.add(Dense(128,activation="relu"))
model.add(Dropout(0.2))
model.add(Dense(3, activation="softmax"))

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

model.summary()
```

```
Model: "sequential"
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 222, 222, 16)      448
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 111, 111, 16)      0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 109, 109, 32)      4640
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 54, 54, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 52, 52, 64)        18496
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 26, 26, 64)        0
_________________________________________________________________
flatten (Flatten)            (None, 43264)             0
_________________________________________________________________
dropout (Dropout)            (None, 43264)             0
_________________________________________________________________
dense (Dense)                (None, 128)               5537920
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 3)                 387
=================================================================
Total params: 5,561,891
Trainable params: 5,561,891
Non-trainable params: 0
_________________________________________________________________
```

[11]:
```
EPOCHS=50
history = model.fit_generator(train_data_gen, epochs=EPOCHS,␣
 ↪validation_data=val_data_gen)
```

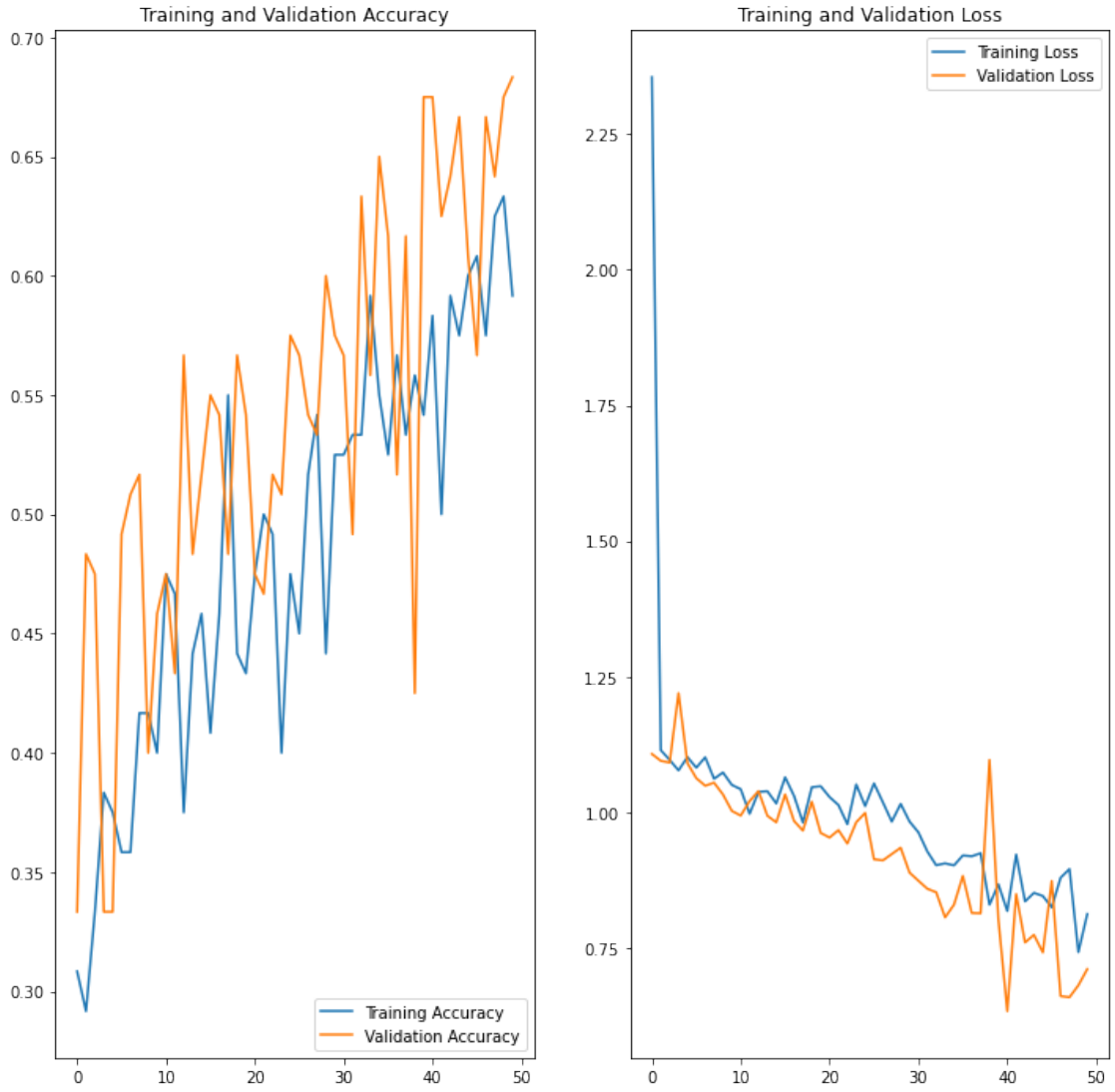
```
/opt/conda/lib/python3.7/site-
packages/tensorflow/python/keras/engine/training.py:1844: UserWarning:
`Model.fit_generator` is deprecated and will be removed in a future version.
Please use `Model.fit`, which supports generators.
  warnings.warn('`Model.fit_generator` is deprecated and '

Epoch 1/50
15/15 [==============================] - 12s 777ms/step - loss: 3.0111 -
accuracy: 0.2745 - val_loss: 1.1084 - val_accuracy: 0.3333
Epoch 2/50
15/15 [==============================] - 10s 654ms/step - loss: 1.1207 -
accuracy: 0.2794 - val_loss: 1.0957 - val_accuracy: 0.4833
Epoch 3/50
15/15 [==============================] - 10s 650ms/step - loss: 1.0968 -
```

```
accuracy: 0.3379 - val_loss: 1.0924 - val_accuracy: 0.4750
Epoch 4/50
15/15 [==============================] - 9s 639ms/step - loss: 1.0867 -
accuracy: 0.3765 - val_loss: 1.2203 - val_accuracy: 0.3333
Epoch 5/50
15/15 [==============================] - 10s 654ms/step - loss: 1.1741 -
accuracy: 0.3260 - val_loss: 1.0914 - val_accuracy: 0.3333
Epoch 6/50
15/15 [==============================] - 9s 638ms/step - loss: 1.0748 -
accuracy: 0.3696 - val_loss: 1.0638 - val_accuracy: 0.4917
Epoch 7/50
15/15 [==============================] - 10s 657ms/step - loss: 1.0784 -
accuracy: 0.4198 - val_loss: 1.0498 - val_accuracy: 0.5083
Epoch 8/50
15/15 [==============================] - 10s 661ms/step - loss: 1.0618 -
accuracy: 0.3689 - val_loss: 1.0554 - val_accuracy: 0.5167
Epoch 9/50
15/15 [==============================] - 10s 645ms/step - loss: 1.0693 -
accuracy: 0.4649 - val_loss: 1.0337 - val_accuracy: 0.4000
Epoch 10/50
15/15 [==============================] - 9s 639ms/step - loss: 1.0641 -
accuracy: 0.3904 - val_loss: 1.0035 - val_accuracy: 0.4583
Epoch 11/50
15/15 [==============================] - 9s 639ms/step - loss: 1.0204 -
accuracy: 0.5002 - val_loss: 0.9949 - val_accuracy: 0.4750
Epoch 12/50
15/15 [==============================] - 10s 669ms/step - loss: 0.9873 -
accuracy: 0.4685 - val_loss: 1.0216 - val_accuracy: 0.4333
Epoch 13/50
15/15 [==============================] - 9s 634ms/step - loss: 1.0197 -
accuracy: 0.3335 - val_loss: 1.0399 - val_accuracy: 0.5667
Epoch 14/50
15/15 [==============================] - 10s 640ms/step - loss: 1.0551 -
accuracy: 0.3957 - val_loss: 0.9945 - val_accuracy: 0.4833
Epoch 15/50
15/15 [==============================] - 10s 694ms/step - loss: 1.0044 -
accuracy: 0.4820 - val_loss: 0.9825 - val_accuracy: 0.5167
Epoch 16/50
15/15 [==============================] - 10s 651ms/step - loss: 1.0584 -
accuracy: 0.4248 - val_loss: 1.0339 - val_accuracy: 0.5500
Epoch 17/50
15/15 [==============================] - 9s 635ms/step - loss: 1.0109 -
accuracy: 0.4612 - val_loss: 0.9849 - val_accuracy: 0.5417
Epoch 18/50
15/15 [==============================] - 10s 679ms/step - loss: 1.0051 -
accuracy: 0.5294 - val_loss: 0.9671 - val_accuracy: 0.4833
Epoch 19/50
15/15 [==============================] - 10s 649ms/step - loss: 0.9986 -
```

```
accuracy: 0.4820 - val_loss: 1.0205 - val_accuracy: 0.5667
Epoch 20/50
15/15 [==============================] - 10s 663ms/step - loss: 1.0622 -
accuracy: 0.4248 - val_loss: 0.9629 - val_accuracy: 0.5417
Epoch 21/50
15/15 [==============================] - 9s 641ms/step - loss: 1.0076 -
accuracy: 0.4397 - val_loss: 0.9544 - val_accuracy: 0.4750
Epoch 22/50
15/15 [==============================] - 10s 666ms/step - loss: 1.0086 -
accuracy: 0.5014 - val_loss: 0.9682 - val_accuracy: 0.4667
Epoch 23/50
15/15 [==============================] - 9s 630ms/step - loss: 0.9784 -
accuracy: 0.5020 - val_loss: 0.9433 - val_accuracy: 0.5167
Epoch 24/50
15/15 [==============================] - 10s 649ms/step - loss: 1.0181 -
accuracy: 0.4546 - val_loss: 0.9826 - val_accuracy: 0.5083
Epoch 25/50
15/15 [==============================] - 9s 643ms/step - loss: 1.0044 -
accuracy: 0.4791 - val_loss: 0.9998 - val_accuracy: 0.5750
Epoch 26/50
15/15 [==============================] - 10s 642ms/step - loss: 1.0451 -
accuracy: 0.4416 - val_loss: 0.9144 - val_accuracy: 0.5667
Epoch 27/50
15/15 [==============================] - 9s 641ms/step - loss: 0.9963 -
accuracy: 0.5264 - val_loss: 0.9123 - val_accuracy: 0.5417
Epoch 28/50
15/15 [==============================] - 10s 659ms/step - loss: 0.9689 -
accuracy: 0.5456 - val_loss: 0.9240 - val_accuracy: 0.5333
Epoch 29/50
15/15 [==============================] - 10s 650ms/step - loss: 1.0336 -
accuracy: 0.4201 - val_loss: 0.9357 - val_accuracy: 0.6000
Epoch 30/50
15/15 [==============================] - 9s 641ms/step - loss: 0.9533 -
accuracy: 0.5617 - val_loss: 0.8898 - val_accuracy: 0.5750
Epoch 31/50
15/15 [==============================] - 10s 643ms/step - loss: 0.9382 -
accuracy: 0.5345 - val_loss: 0.8749 - val_accuracy: 0.5667
Epoch 32/50
15/15 [==============================] - 10s 652ms/step - loss: 0.9554 -
accuracy: 0.5353 - val_loss: 0.8602 - val_accuracy: 0.4917
Epoch 33/50
15/15 [==============================] - 10s 660ms/step - loss: 0.9175 -
accuracy: 0.5476 - val_loss: 0.8537 - val_accuracy: 0.6333
Epoch 34/50
15/15 [==============================] - 9s 633ms/step - loss: 0.9401 -
accuracy: 0.5971 - val_loss: 0.8074 - val_accuracy: 0.5583
Epoch 35/50
15/15 [==============================] - 10s 653ms/step - loss: 0.8931 -
```

```
accuracy: 0.5488 - val_loss: 0.8308 - val_accuracy: 0.6500
Epoch 36/50
15/15 [==============================] - 10s 652ms/step - loss: 0.9257 -
accuracy: 0.4885 - val_loss: 0.8838 - val_accuracy: 0.6167
Epoch 37/50
15/15 [==============================] - 10s 646ms/step - loss: 0.9359 -
accuracy: 0.5518 - val_loss: 0.8156 - val_accuracy: 0.5167
Epoch 38/50
15/15 [==============================] - 10s 642ms/step - loss: 0.9125 -
accuracy: 0.5101 - val_loss: 0.8150 - val_accuracy: 0.6167
Epoch 39/50
15/15 [==============================] - 10s 658ms/step - loss: 0.7747 -
accuracy: 0.6393 - val_loss: 1.0976 - val_accuracy: 0.4250
Epoch 40/50
15/15 [==============================] - 9s 627ms/step - loss: 0.9141 -
accuracy: 0.5028 - val_loss: 0.8036 - val_accuracy: 0.6750
Epoch 41/50
15/15 [==============================] - 10s 655ms/step - loss: 0.8790 -
accuracy: 0.5407 - val_loss: 0.6348 - val_accuracy: 0.6750
Epoch 42/50
15/15 [==============================] - 10s 653ms/step - loss: 1.0064 -
accuracy: 0.4193 - val_loss: 0.8503 - val_accuracy: 0.6250
Epoch 43/50
15/15 [==============================] - 10s 632ms/step - loss: 0.7850 -
accuracy: 0.6380 - val_loss: 0.7612 - val_accuracy: 0.6417
Epoch 44/50
15/15 [==============================] - 10s 658ms/step - loss: 0.8648 -
accuracy: 0.5755 - val_loss: 0.7752 - val_accuracy: 0.6667
Epoch 45/50
15/15 [==============================] - 10s 655ms/step - loss: 0.8568 -
accuracy: 0.5595 - val_loss: 0.7430 - val_accuracy: 0.6083
Epoch 46/50
15/15 [==============================] - 9s 642ms/step - loss: 0.8237 -
accuracy: 0.5996 - val_loss: 0.8745 - val_accuracy: 0.5667
Epoch 47/50
15/15 [==============================] - 10s 663ms/step - loss: 0.9220 -
accuracy: 0.5834 - val_loss: 0.6625 - val_accuracy: 0.6667
Epoch 48/50
15/15 [==============================] - 10s 644ms/step - loss: 0.9779 -
accuracy: 0.5795 - val_loss: 0.6604 - val_accuracy: 0.6417
Epoch 49/50
15/15 [==============================] - 10s 640ms/step - loss: 0.7130 -
accuracy: 0.6859 - val_loss: 0.6824 - val_accuracy: 0.6750
Epoch 50/50
15/15 [==============================] - 9s 638ms/step - loss: 0.7852 -
accuracy: 0.6131 - val_loss: 0.7123 - val_accuracy: 0.6833
```

[12]:
```
# Plot training and validation graphs
acc = history.history['accuracy']
val_accuracy = history.history['val_accuracy']

loss = history.history['loss']
val_loss = history.history['val_loss']

epochs_range = range(EPOCHS)

plt.figure(figsize=(12,12))
plt.subplot(1,2,1)
plt.plot(epochs_range,acc,label='Training Accuracy')
plt.plot(epochs_range,val_accuracy,label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')

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



[13]:
```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
```