

assignment06convnet_6.2b_6.3

July 15, 2023

[17]: #6.2b

```
[3]: from contextlib import redirect_stdout
import pathlib
from keras.utils import to_categorical
from pathlib import Path
from sklearn.model_selection import train_test_split
from keras.preprocessing.image import ImageDataGenerator
from keras.preprocessing import image
from keras.layers import Conv2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
import tensorflow as tf
import imageio
from keras.layers.core import Dense, Dropout, Activation
from keras import layers
from keras import models
from keras import optimizers
import os, shutil
from keras.applications import VGG16
from keras.models import Sequential, load_model
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

import time
start_time = time.time()
```

```
2023-07-15 18:56:48.689805: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not
find cuda drivers on your machine, GPU will not be used.
2023-07-15 18:56:48.818075: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not
find cuda drivers on your machine, GPU will not be used.
2023-07-15 18:56:48.819345: I tensorflow/core/platform/cpu_feature_guard.cc:182]
This TensorFlow binary is optimized to use available CPU instructions in
performance-critical operations.
To enable the following instructions: AVX2 AVX512F FMA, in other operations,
rebuild TensorFlow with the appropriate compiler flags.
2023-07-15 18:56:50.191570: W
```

```
tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
```

```
[20]: homebase=Path('/home/jovyan/DSC650/dsc650/')

myresults=Path('/home/jovyan/DSC650/dsc650/assignments/assignment06/').
    ↪joinpath('results')

myresults.mkdir(parents=True, exist_ok=True)
```

```
[21]: CIFAR= tf.keras.datasets.cifar10.load_data()
(x_train, y_train), (x_test, y_test) = CIFAR
x_train.shape,y_train.shape,x_test.shape,y_test.shape
```

```
[21]: ((50000, 32, 32, 3), (50000, 1), (10000, 32, 32, 3), (10000, 1))
```

```
[22]: def load_dataset():
    (x_train, y_train), (x_test, y_test) = CIFAR
    trainY=to_categorical(y_train)
    testY=to_categorical(y_test)

    return x_train, trainY, x_test, testY
```

```
[23]: #homemade scale
def prep_pixels(train, test):
    train_norm = train.astype('float32')
    test_norm = test.astype('float32')
    train_norm = train_norm / 255.0
    test_norm = test_norm / 255.0

    return train_norm, test_norm
```

```
[32]: def sum_diagnostics(history):
    plt.subplot(211)
    plt.title('Cross Entropy Loss')
    plt.plot(history.history['loss'], color='blue',
              label='train')
    plt.plot(history.history['val_loss'], color='orange',
              label='test')
    plt.subplot(212)
    plt.title('Classification Accuracy')
    plt.plot(history.history['accuracy'], color='blue',
              label='train')
    plt.plot(history.history['val_accuracy'], color='orange',
              label='test')
```

```

filedAway=myresults.joinpath('6.2B_SummaryPlt.png')
plt.savefig(filedAway)
plt.close()

```

```

[33]: def plotdeconfusion(cm, classes,
                        normalize=False,
                        cmap= plt.cm.Blues):
    import itertools

    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title("De Confusion Matrix")
    plt.colorbar()
    tickSmarked = np.arange(len(classes))
    plt.xticks(tickSmarked, classes, rotation=45)
    plt.yticks(tickSmarked, classes)

    if normalize:
        cm=cm.astype('float') / cm.sum(axis=1)[:,np.newaxis]

    thresholds=cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]),
                                  range(cm.shape[1])):
        plt.text(j, i, cm[i, j],
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresholds else "black")

    plt.tight_layout()
    plt.ylabel('True')
    plt.xlabel('Predicted')

    imged_file=myresults.joinpath('6.2B_Confusions.png')
    plt.savefig(imged_file)
    plt.show()

```

```

[34]: def defineDEmodel():
    model=Sequential()
    model.add(Conv2D(32, (3, 3), activation='relu',
                     kernel_initializer='he_uniform',
                     padding='same',
                     input_shape=(32, 32, 3)))
    model.add(Conv2D(32, (3, 3), activation='relu',
                     kernel_initializer='he_uniform',
                     padding='same'))
    model.add(MaxPooling2D((2, 2)))
    model.add(Conv2D(64, (3, 3), activation='relu',
                     kernel_initializer='he_uniform',

```

```

        padding='same'))
model.add(Conv2D(64, (3, 3), activation='relu',
                kernel_initializer='he_uniform',
                padding='same'))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(128, (3, 3), activation='relu',
                kernel_initializer='he_uniform',
                padding='same'))
model.add(Conv2D(128, (3, 3), activation='relu',
                kernel_initializer='he_uniform',
                padding='same'))
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu',
                kernel_initializer='he_uniform'))
model.add(Dense(10, activation='softmax'))
from keras.optimizers import SGD
optimum=SGD(lr=0.001, momentum=0.9)
model.compile(optimizer=optimum,
              metrics=['accuracy'],
              loss='categorical_crossentropy')

return model

```

```

[35]: def loader(filename):
    img=tf.keras.utils.load_img(filename, target_size=(32, 32))
    img=img_to_array(img)
    img=img.reshape(1, 32, 32, 3)

    img=img.astype('float32')
    img=img/255.0
    return img

```

```

[36]: def run_example_prediction():
    classes=('airplane', 'automobile', 'bird', 'cat', 'deer',
            'dog', 'frog', 'horse', 'ship', 'truck')
    print("Attempting to predict image: 6.2B_Sampler_CIFAR.png")

    result_model_file =myresults.joinpath('6.2A_model.h5')
    model=loader(result_model_file)
    summary_file=myresults.joinpath('6.2A_ModelSummaryLoaded.txt')
    with open(summary_file, 'w') as f:
        with redirect_stdout(f):
            model.summary()

    filenameed = myresults.joinpath('6.2Bsample_image.png')
    img = loader(filenameed)

```

```

result = model.predict_classes(img)
print("The picture prediction is:.....")
print(classes[result[0]])
print("Time: %s seconds " % (time.time() - start_time))

```

```

[37]: def run_test_harness():
    import warnings
    warnings.filterwarnings('ignore')
    print("loading...")
    x_train, trainY, x_test, testY = load_dataset()
    print("Time: %s seconds" % (time.time() - start_time))

    print("preparing...")
    x_train, x_test=prep_pixels(x_train, x_test)

    for i in range(9):
        plt.subplot(330 + 1 + i)
        z = x_train[i]
        z = np.reshape(z, (32, 32, 3))
        plt.imshow(z)

    img_files = myresults.joinpath('6.2B_Sampler_CIFAR.png')
    plt.savefig(img_files)
    plt.show()
    print("Time: %s seconds" % (time.time() - start_time))

    print("defining...")
    model = definedEModel()
    summary_file = myresults.joinpath('6.2B_ModelSummary.txt')
    with open(summary_file, 'w') as f:
        with redirect_stdout(f):
            model.summary()
    print("Time: %s seconds" % (time.time() - start_time))

    print("adding augmentation with datagen")

    datagen = ImageDataGenerator(rotation_range=40,horizontal_flip=True)
    train_datagen = ImageDataGenerator(rescale=1. / 255, shear_range=0.2,
                                       zoom_range=0.2)
    test_datagen = ImageDataGenerator(rescale=1. / 255)
    print("Time: %s seconds" % (time.time() - start_time))

    print("preparing iterations...")
    iter_training = datagen.flow(x_train, trainY, batch_size=64)
    train_generator = train_datagen.flow(x_train, trainY, batch_size=64)
    validation_generator = test_datagen.flow(x_train, trainY, batch_size=64)

```

```

print("Time: %s seconds" % (time.time() - start_time))

print("fitting...")
#history = model.fit(x_train, trainY, epochs=20,
                    #batch_size=64,
                    #validation_data=(x_test, testY),
                    #verbose=0)

nb_train_samples = 18000
nb_validation_samples = 7000
epochs = 20
batch_size = 144
history = model.fit_generator(train_generator,
                             steps_per_epoch=nb_train_samples // batch_size,
                             epochs=epochs,
                             validation_data=validation_generator,
                             validation_steps=nb_validation_samples // batch_size)

print("Time: %s seconds" % (time.time() - start_time))

print("evaluating...")
_, accuracy = model.evaluate(x_test, testY, verbose=0)
print('> %.2f' % (accuracy*100.0))
print("Time: %s seconds" % (time.time() - start_time))
img_files = myresults.joinpath('6.2B_Sampler_CIFAR.png')
plt.savefig(img_files)
plt.show()
print("Time: %s seconds" % (time.time() - start_time))

results0model_file = myresults.joinpath('6.2B_model.h5')
model.save(results0model_file)
print("I've Saved the trained model at %s " % results0model_file)

print("preparing diagnosis summary...")
sum_diagnostics(history)
print("Time: %s seconds" % (time.time() - start_time))

print("predicting... ")
Y_pred = model.predict(x_test)
Y_pred_classes = np.argmax(Y_pred, axis=1)
Y_true = np.argmax(testY, axis=1)
from sklearn.metrics import confusion_matrix
confusion=confusion_matrix(Y_true, Y_pred_classes)
plotdeconfusion(confusion, classes=range(10))
print("Time: %s seconds" % (time.time() - start_time))

print("preparing classifications")

```

```

classes=('airplane', 'automobile', 'bird', 'cat', 'deer',
        'dog', 'frog', 'horse', 'ship', 'truck')
correct_in = np.nonzero(Y_pred_classes == Y_true)[0]
incorrect_in = np.nonzero(Y_pred_classes != Y_true)[0]
print(len(Y_pred_classes))
print(len(Y_true))
print(len(correct_in), " correctly classed")
print(len(incorrect_in), " incorrectly classed")

plt.rcParams['figure.figsize'] = (7, 14)
figure_evaluation = plt.figure()

for i, correct in enumerate(correct_in[:14]):
    plt.subplot(6, 3, i + 1)
    plt.imshow(x_test[correct], cmap='gray',
               interpolation='none')
    plt.title("Predicted: {}, True: {}".format(
        classes[Y_pred[correct].argmax()],
        classes[testY[correct].argmax()]))
    plt.xticks([])
    plt.yticks([])

images_file = myresults.joinpath('6.2B_CorrectPrediction.png')
plt.savefig(images_file)
plt.show()

for i, incorrect in enumerate(incorrect_in[:9]):
    plt.subplot(6, 3, i + 10)
    plt.imshow(x_test[incorrect], cmap='gray',
               interpolation='none')
    plt.title(
        "Predicted {}, True: {}".format(
            classes[Y_pred[incorrect].argmax()],
            classes[testY[incorrect].argmax()]))
    plt.xticks([])
    plt.yticks([])

images_file = myresults.joinpath('6.2B_IncorrectPrediction.png')
plt.savefig(images_file)
plt.show()
print("Time: %s seconds" % (time.time() - start_time))

```

```
[42]: run_test_harness()
```

loading...

Time: 14.298466920852661 seconds

preparing...



Time: 16.32192039489746 seconds

defining...

Time: 16.463767528533936 seconds

adding augmentation with datagen

Time: 16.46396493911743 seconds

preparing iterations...

Time: 16.464104652404785 seconds

fitting...

Epoch 1/20

2023-07-13 21:34:28.427784: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'Placeholder/_0' with dtype int32

[[{{node Placeholder/_0}}]]

125/125 [=====] - ETA: 0s - loss: 2.3023 - accuracy: 0.1042

2023-07-13 21:34:39.333096: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value


```

for placeholder tensor 'Placeholder/_0' with dtype int32
    [[{{node Placeholder/_0}}]]

125/125 [=====] - 12s 88ms/step - loss: 2.3023 -
accuracy: 0.1042 - val_loss: 2.3020 - val_accuracy: 0.1432
Epoch 2/20
125/125 [=====] - 10s 83ms/step - loss: 2.3021 -
accuracy: 0.1082 - val_loss: 2.3018 - val_accuracy: 0.1022
Epoch 3/20
125/125 [=====] - 11s 84ms/step - loss: 2.3018 -
accuracy: 0.1070 - val_loss: 2.3017 - val_accuracy: 0.0980
Epoch 4/20
125/125 [=====] - 11s 83ms/step - loss: 2.3016 -
accuracy: 0.1082 - val_loss: 2.3011 - val_accuracy: 0.0970
Epoch 5/20
125/125 [=====] - 11s 84ms/step - loss: 2.3012 -
accuracy: 0.1105 - val_loss: 2.3022 - val_accuracy: 0.0921
Epoch 6/20
125/125 [=====] - 10s 83ms/step - loss: 2.3016 -
accuracy: 0.1017 - val_loss: 2.3030 - val_accuracy: 0.0918
Epoch 7/20
125/125 [=====] - 10s 82ms/step - loss: 2.3015 -
accuracy: 0.1054 - val_loss: 2.3005 - val_accuracy: 0.1426
Epoch 8/20
125/125 [=====] - 10s 82ms/step - loss: 2.2996 -
accuracy: 0.1150 - val_loss: 2.3007 - val_accuracy: 0.1022
Epoch 9/20
125/125 [=====] - 10s 83ms/step - loss: 2.3003 -
accuracy: 0.1196 - val_loss: 2.3014 - val_accuracy: 0.1029
Epoch 10/20
125/125 [=====] - 11s 83ms/step - loss: 2.2994 -
accuracy: 0.1269 - val_loss: 2.2982 - val_accuracy: 0.1455
Epoch 11/20
125/125 [=====] - 10s 83ms/step - loss: 2.2983 -
accuracy: 0.1204 - val_loss: 2.3003 - val_accuracy: 0.0990
Epoch 12/20
125/125 [=====] - 11s 84ms/step - loss: 2.2986 -
accuracy: 0.1177 - val_loss: 2.2980 - val_accuracy: 0.1022
Epoch 13/20
125/125 [=====] - 11s 83ms/step - loss: 2.2986 -
accuracy: 0.1224 - val_loss: 2.2973 - val_accuracy: 0.1146
Epoch 14/20
125/125 [=====] - 11s 84ms/step - loss: 2.2987 -
accuracy: 0.1116 - val_loss: 2.2974 - val_accuracy: 0.1191
Epoch 15/20
125/125 [=====] - 10s 83ms/step - loss: 2.2960 -
accuracy: 0.1248 - val_loss: 2.3001 - val_accuracy: 0.1064
Epoch 16/20

```

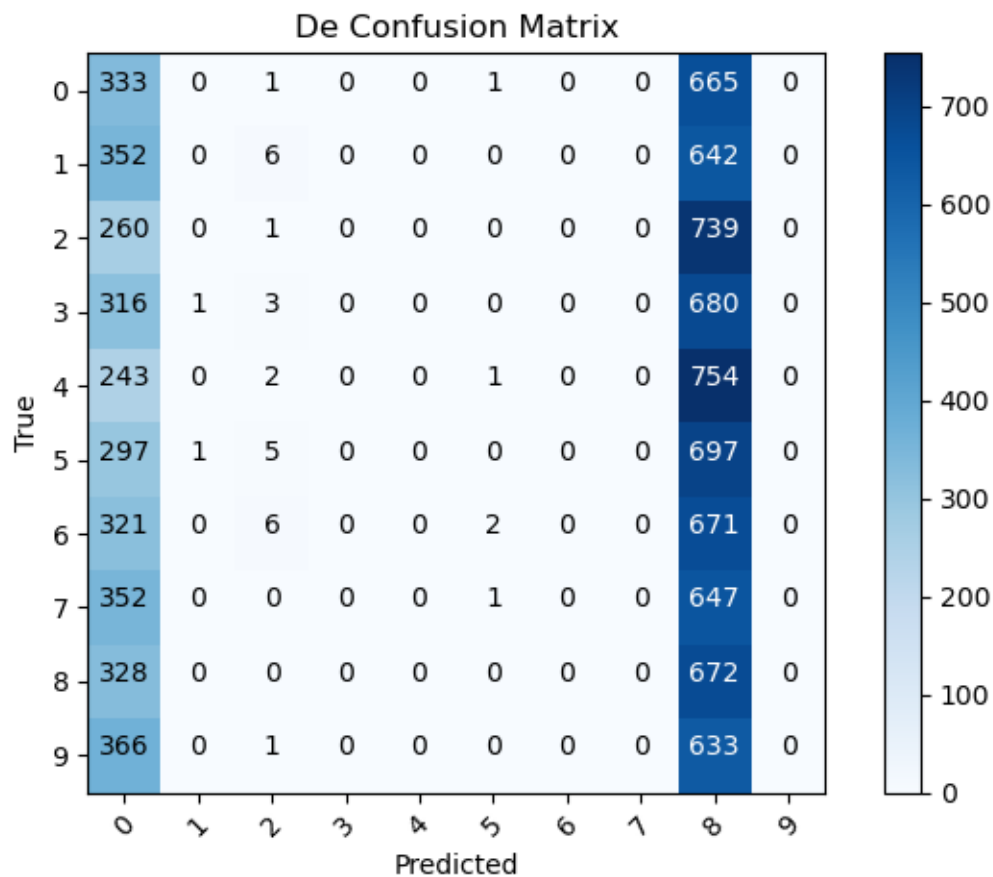
```

125/125 [=====] - 10s 83ms/step - loss: 2.2952 -
accuracy: 0.1258 - val_loss: 2.2917 - val_accuracy: 0.1139
Epoch 17/20
125/125 [=====] - 10s 83ms/step - loss: 2.2935 -
accuracy: 0.1262 - val_loss: 2.2913 - val_accuracy: 0.1090
Epoch 18/20
125/125 [=====] - 10s 83ms/step - loss: 2.2939 -
accuracy: 0.1147 - val_loss: 2.2883 - val_accuracy: 0.1813
Epoch 19/20
125/125 [=====] - 10s 83ms/step - loss: 2.2924 -
accuracy: 0.1285 - val_loss: 2.2918 - val_accuracy: 0.0996
Epoch 20/20
125/125 [=====] - 10s 82ms/step - loss: 2.2928 -
accuracy: 0.1146 - val_loss: 2.2862 - val_accuracy: 0.1914
Time: 228.23956108093262 seconds
evaluating...
> 10.06
Time: 232.31609296798706 seconds

<Figure size 640x480 with 0 Axes>

Time: 232.34130954742432 seconds
I've Saved the trained model at
/home/jovyan/DSC650/dsc650/assignments/assignment06/results/6.2B_model.h5
preparing diagnosis summary...
Time: 233.0051281452179 seconds
predicting...
313/313 [=====] - 4s 12ms/step

```



```

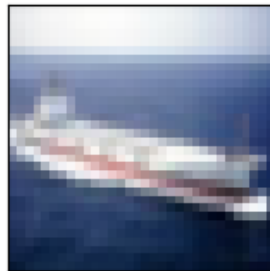
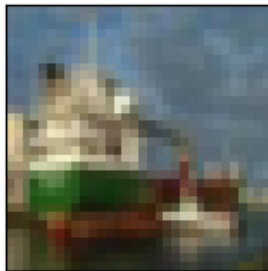
Time: 244.1456606388092 seconds
preparing classifications
10000
10000
1006  correctly classed
8994  incorrectly classed

```

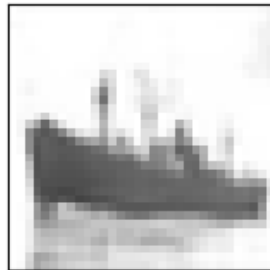
Predicted: ship, True: ship Predicted: ship, True: ship Predicted: airplane, True: airplane



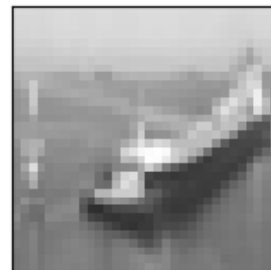
Predicted: ship, True: ship Predicted: ship, True: ship Predicted: ship, True: ship



Predicted: ship, True: ship Predicted: ship, True: ship Predicted: ship, True: ship



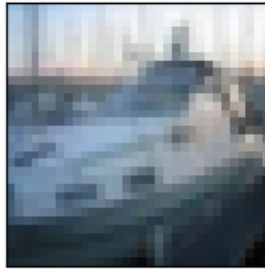
Predicted: ship, True: ship Predicted: airplane, True: airplane Predicted: ship, True: ship



Predicted: ship, True: ship Predicted: airplane, True: airplane



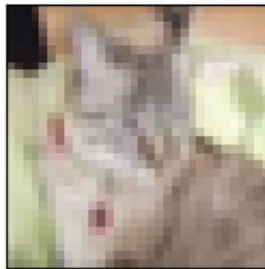
Predicted ship, True: frog Predicted airplane, True: frog Predicted ship, True: airplane



Predicted ship, True: frog Predicted ship, True: frog Predicted bird, True: automobile



Predicted ship, True: frog Predicted ship, True: frog Predicted airplane, True: automobile



Time: 246.01011085510254 seconds

```
[ ]: def run_example_prediction():
    classes=('airplane', 'automobile', 'bird', 'cat', 'deer',
            'dog', 'frog', 'horse', 'ship', 'truck')
    print("Attempting to predict image: 6.2B_Sampler_CIFAR.png")

    result_model_file =myresults.joinpath('6.2B_model.h5')
    model=loader(result_model_file)
    summary_file=myresults.joinpath('6.2B_ModelSummaryLoaded.txt')
    with open(summary_file, 'w') as f:
        with redirect_stdout(f):
            model.summary()

    filenameed = myresults.joinpath('6.2Bsample_image.png')
    img = loader(filenameed)
```

```

result = model.predict_classes(img)
print("The picture prediction is:.....")
print(classes[result[0]])
print("Time: %s seconds " % (time.time() - start_time))

```

```

[43]: homebase=Path('/home/jovyan/DSC650/dsc650/')

myimages=Path('/home/jovyan/DSC650/dsc650/assignments/assignment06/').
    ↪joinpath('images')

myimages.mkdir(parents=True, exist_ok=True)

```

```

[4]: #6.3
import pathlib
from tensorflow.keras.applications.resnet50 import ResNet50
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.resnet50 import preprocess_input,
    ↪decode_predictions
import numpy as np

model = ResNet50(weights='imagenet')

model.summary()

```

Model: "resnet50"

| Layer (type) | Output Shape | Param # | Connected to |
|-------------------------------|----------------------|---------|----------------------|
| ===== | | | |
| input_1 (InputLayer) | [(None, 224, 224, 3 | 0 | [] |
| |)] | | |
| conv1_pad (ZeroPadding2D) | (None, 230, 230, 3) | 0 | ['input_1[0][0]'] |
| | ['input_1[0][0]'] | | |
| conv1_conv (Conv2D) | (None, 112, 112, 64 | 9472 | ['conv1_pad[0][0]'] |
| | ['conv1_pad[0][0]'] | | |
| |) | | |
| conv1_bn (BatchNormalization) | (None, 112, 112, 64 | 256 | ['conv1_conv[0][0]'] |
| | ['conv1_conv[0][0]'] | | |
| |) | | |
| conv1_relu (Activation) | (None, 112, 112, 64 | 0 | ['conv1_bn[0][0]'] |
| | ['conv1_bn[0][0]'] | | |

```

)

pool1_pad (ZeroPadding2D)      (None, 114, 114, 64) 0
['conv1_relu[0][0]']

)

pool1_pool (MaxPooling2D)      (None, 56, 56, 64) 0
['pool1_pad[0][0]']

conv2_block1_1_conv (Conv2D)   (None, 56, 56, 64) 4160
['pool1_pool[0][0]']

conv2_block1_1_bn (BatchNormal (None, 56, 56, 64) 256
['conv2_block1_1_conv[0][0]']
ization)

conv2_block1_1_relu (Activatio (None, 56, 56, 64) 0
['conv2_block1_1_bn[0][0]']
n)

conv2_block1_2_conv (Conv2D)   (None, 56, 56, 64) 36928
['conv2_block1_1_relu[0][0]']

conv2_block1_2_bn (BatchNormal (None, 56, 56, 64) 256
['conv2_block1_2_conv[0][0]']
ization)

conv2_block1_2_relu (Activatio (None, 56, 56, 64) 0
['conv2_block1_2_bn[0][0]']
n)

conv2_block1_0_conv (Conv2D)   (None, 56, 56, 256) 16640
['pool1_pool[0][0]']

conv2_block1_3_conv (Conv2D)   (None, 56, 56, 256) 16640
['conv2_block1_2_relu[0][0]']

conv2_block1_0_bn (BatchNormal (None, 56, 56, 256) 1024
['conv2_block1_0_conv[0][0]']
ization)

conv2_block1_3_bn (BatchNormal (None, 56, 56, 256) 1024
['conv2_block1_3_conv[0][0]']
ization)

conv2_block1_add (Add)          (None, 56, 56, 256) 0
['conv2_block1_0_bn[0][0]',
'conv2_block1_3_bn[0][0]']

```

```

conv2_block1_out (Activation) (None, 56, 56, 256) 0
['conv2_block1_add[0][0]']

conv2_block2_1_conv (Conv2D) (None, 56, 56, 64) 16448
['conv2_block1_out[0][0]']

conv2_block2_1_bn (BatchNormal (None, 56, 56, 64) 256
['conv2_block2_1_conv[0][0]']
ization)

conv2_block2_1_relu (Activatio (None, 56, 56, 64) 0
['conv2_block2_1_bn[0][0]']
n)

conv2_block2_2_conv (Conv2D) (None, 56, 56, 64) 36928
['conv2_block2_1_relu[0][0]']

conv2_block2_2_bn (BatchNormal (None, 56, 56, 64) 256
['conv2_block2_2_conv[0][0]']
ization)

conv2_block2_2_relu (Activatio (None, 56, 56, 64) 0
['conv2_block2_2_bn[0][0]']
n)

conv2_block2_3_conv (Conv2D) (None, 56, 56, 256) 16640
['conv2_block2_2_relu[0][0]']

conv2_block2_3_bn (BatchNormal (None, 56, 56, 256) 1024
['conv2_block2_3_conv[0][0]']
ization)

conv2_block2_add (Add) (None, 56, 56, 256) 0
['conv2_block1_out[0][0]',
'conv2_block2_3_bn[0][0]']

conv2_block2_out (Activation) (None, 56, 56, 256) 0
['conv2_block2_add[0][0]']

conv2_block3_1_conv (Conv2D) (None, 56, 56, 64) 16448
['conv2_block2_out[0][0]']

conv2_block3_1_bn (BatchNormal (None, 56, 56, 64) 256
['conv2_block3_1_conv[0][0]']
ization)

conv2_block3_1_relu (Activatio (None, 56, 56, 64) 0

```



```

['conv2_block3_1_bn[0][0]']
n)

conv2_block3_2_conv (Conv2D) (None, 56, 56, 64) 36928
['conv2_block3_1_relu[0][0]']

conv2_block3_2_bn (BatchNormal (None, 56, 56, 64) 256
['conv2_block3_2_conv[0][0]']
ization)

conv2_block3_2_relu (Activatio (None, 56, 56, 64) 0
['conv2_block3_2_bn[0][0]']
n)

conv2_block3_3_conv (Conv2D) (None, 56, 56, 256) 16640
['conv2_block3_2_relu[0][0]']

conv2_block3_3_bn (BatchNormal (None, 56, 56, 256) 1024
['conv2_block3_3_conv[0][0]']
ization)

conv2_block3_add (Add) (None, 56, 56, 256) 0
['conv2_block2_out[0][0]',
'conv2_block3_3_bn[0][0]']

conv2_block3_out (Activation) (None, 56, 56, 256) 0
['conv2_block3_add[0][0]']

conv3_block1_1_conv (Conv2D) (None, 28, 28, 128) 32896
['conv2_block3_out[0][0]']

conv3_block1_1_bn (BatchNormal (None, 28, 28, 128) 512
['conv3_block1_1_conv[0][0]']
ization)

conv3_block1_1_relu (Activatio (None, 28, 28, 128) 0
['conv3_block1_1_bn[0][0]']
n)

conv3_block1_2_conv (Conv2D) (None, 28, 28, 128) 147584
['conv3_block1_1_relu[0][0]']

conv3_block1_2_bn (BatchNormal (None, 28, 28, 128) 512
['conv3_block1_2_conv[0][0]']
ization)

conv3_block1_2_relu (Activatio (None, 28, 28, 128) 0
['conv3_block1_2_bn[0][0]']

```

n)

conv3_block1_0_conv (Conv2D) (None, 28, 28, 512) 131584
['conv2_block3_out[0][0]']

conv3_block1_3_conv (Conv2D) (None, 28, 28, 512) 66048
['conv3_block1_2_relu[0][0]']

conv3_block1_0_bn (BatchNormal (None, 28, 28, 512) 2048
['conv3_block1_0_conv[0][0]']
ization)

conv3_block1_3_bn (BatchNormal (None, 28, 28, 512) 2048
['conv3_block1_3_conv[0][0]']
ization)

conv3_block1_add (Add) (None, 28, 28, 512) 0
['conv3_block1_0_bn[0][0]',
'conv3_block1_3_bn[0][0]']

conv3_block1_out (Activation) (None, 28, 28, 512) 0
['conv3_block1_add[0][0]']

conv3_block2_1_conv (Conv2D) (None, 28, 28, 128) 65664
['conv3_block1_out[0][0]']

conv3_block2_1_bn (BatchNormal (None, 28, 28, 128) 512
['conv3_block2_1_conv[0][0]']
ization)

conv3_block2_1_relu (Activatio (None, 28, 28, 128) 0
['conv3_block2_1_bn[0][0]']
n)

conv3_block2_2_conv (Conv2D) (None, 28, 28, 128) 147584
['conv3_block2_1_relu[0][0]']

conv3_block2_2_bn (BatchNormal (None, 28, 28, 128) 512
['conv3_block2_2_conv[0][0]']
ization)

conv3_block2_2_relu (Activatio (None, 28, 28, 128) 0
['conv3_block2_2_bn[0][0]']
n)

conv3_block2_3_conv (Conv2D) (None, 28, 28, 512) 66048
['conv3_block2_2_relu[0][0]']

```

conv3_block2_3_bn (BatchNormal (None, 28, 28, 512) 2048
['conv3_block2_3_conv[0][0]'
ization)

conv3_block2_add (Add) (None, 28, 28, 512) 0
['conv3_block1_out[0][0]',
'conv3_block2_3_bn[0][0]']

conv3_block2_out (Activation) (None, 28, 28, 512) 0
['conv3_block2_add[0][0]']

conv3_block3_1_conv (Conv2D) (None, 28, 28, 128) 65664
['conv3_block2_out[0][0]']

conv3_block3_1_bn (BatchNormal (None, 28, 28, 128) 512
['conv3_block3_1_conv[0][0]'
ization)

conv3_block3_1_relu (Activatio (None, 28, 28, 128) 0
['conv3_block3_1_bn[0][0]'
n)

conv3_block3_2_conv (Conv2D) (None, 28, 28, 128) 147584
['conv3_block3_1_relu[0][0]']

conv3_block3_2_bn (BatchNormal (None, 28, 28, 128) 512
['conv3_block3_2_conv[0][0]'
ization)

conv3_block3_2_relu (Activatio (None, 28, 28, 128) 0
['conv3_block3_2_bn[0][0]'
n)

conv3_block3_3_conv (Conv2D) (None, 28, 28, 512) 66048
['conv3_block3_2_relu[0][0]']

conv3_block3_3_bn (BatchNormal (None, 28, 28, 512) 2048
['conv3_block3_3_conv[0][0]'
ization)

conv3_block3_add (Add) (None, 28, 28, 512) 0
['conv3_block2_out[0][0]',
'conv3_block3_3_bn[0][0]']

conv3_block3_out (Activation) (None, 28, 28, 512) 0
['conv3_block3_add[0][0]']

conv3_block4_1_conv (Conv2D) (None, 28, 28, 128) 65664

```

```

['conv3_block3_out[0][0]']

conv3_block4_1_bn (BatchNormal (None, 28, 28, 128) 512
['conv3_block4_1_conv[0][0]']
ization)

conv3_block4_1_relu (Activatio (None, 28, 28, 128) 0
['conv3_block4_1_bn[0][0]']
n)

conv3_block4_2_conv (Conv2D) (None, 28, 28, 128) 147584
['conv3_block4_1_relu[0][0]']

conv3_block4_2_bn (BatchNormal (None, 28, 28, 128) 512
['conv3_block4_2_conv[0][0]']
ization)

conv3_block4_2_relu (Activatio (None, 28, 28, 128) 0
['conv3_block4_2_bn[0][0]']
n)

conv3_block4_3_conv (Conv2D) (None, 28, 28, 512) 66048
['conv3_block4_2_relu[0][0]']

conv3_block4_3_bn (BatchNormal (None, 28, 28, 512) 2048
['conv3_block4_3_conv[0][0]']
ization)

conv3_block4_add (Add) (None, 28, 28, 512) 0
['conv3_block3_out[0][0]',
'conv3_block4_3_bn[0][0]']

conv3_block4_out (Activation) (None, 28, 28, 512) 0
['conv3_block4_add[0][0]']

conv4_block1_1_conv (Conv2D) (None, 14, 14, 256) 131328
['conv3_block4_out[0][0]']

conv4_block1_1_bn (BatchNormal (None, 14, 14, 256) 1024
['conv4_block1_1_conv[0][0]']
ization)

conv4_block1_1_relu (Activatio (None, 14, 14, 256) 0
['conv4_block1_1_bn[0][0]']
n)

conv4_block1_2_conv (Conv2D) (None, 14, 14, 256) 590080
['conv4_block1_1_relu[0][0]']

```

```

conv4_block1_2_bn (BatchNormal (None, 14, 14, 256) 1024
['conv4_block1_2_conv[0][0]']
ization)

conv4_block1_2_relu (Activatio (None, 14, 14, 256) 0
['conv4_block1_2_bn[0][0]']
n)

conv4_block1_0_conv (Conv2D) (None, 14, 14, 1024 525312
['conv3_block4_out[0][0]']
)

conv4_block1_3_conv (Conv2D) (None, 14, 14, 1024 263168
['conv4_block1_2_relu[0][0]']
)

conv4_block1_0_bn (BatchNormal (None, 14, 14, 1024 4096
['conv4_block1_0_conv[0][0]']
ization)
)

conv4_block1_3_bn (BatchNormal (None, 14, 14, 1024 4096
['conv4_block1_3_conv[0][0]']
ization)
)

conv4_block1_add (Add) (None, 14, 14, 1024 0
['conv4_block1_0_bn[0][0]',
]
)
['conv4_block1_3_bn[0][0]']

conv4_block1_out (Activation) (None, 14, 14, 1024 0
['conv4_block1_add[0][0]']
)

conv4_block2_1_conv (Conv2D) (None, 14, 14, 256) 262400
['conv4_block1_out[0][0]']

conv4_block2_1_bn (BatchNormal (None, 14, 14, 256) 1024
['conv4_block2_1_conv[0][0]']
ization)

conv4_block2_1_relu (Activatio (None, 14, 14, 256) 0
['conv4_block2_1_bn[0][0]']
n)

conv4_block2_2_conv (Conv2D) (None, 14, 14, 256) 590080
['conv4_block2_1_relu[0][0]']

```

```

conv4_block2_2_bn (BatchNormal (None, 14, 14, 256) 1024
['conv4_block2_2_conv[0][0]']
ization)

conv4_block2_2_relu (Activatio (None, 14, 14, 256) 0
['conv4_block2_2_bn[0][0]']
n)

conv4_block2_3_conv (Conv2D) (None, 14, 14, 1024 263168
['conv4_block2_2_relu[0][0]']
)

conv4_block2_3_bn (BatchNormal (None, 14, 14, 1024 4096
['conv4_block2_3_conv[0][0]']
ization)

conv4_block2_add (Add) (None, 14, 14, 1024 0
['conv4_block1_out[0][0]',
)
'conv4_block2_3_bn[0][0]']

conv4_block2_out (Activation) (None, 14, 14, 1024 0
['conv4_block2_add[0][0]']
)

conv4_block3_1_conv (Conv2D) (None, 14, 14, 256) 262400
['conv4_block2_out[0][0]']

conv4_block3_1_bn (BatchNormal (None, 14, 14, 256) 1024
['conv4_block3_1_conv[0][0]']
ization)

conv4_block3_1_relu (Activatio (None, 14, 14, 256) 0
['conv4_block3_1_bn[0][0]']
n)

conv4_block3_2_conv (Conv2D) (None, 14, 14, 256) 590080
['conv4_block3_1_relu[0][0]']

conv4_block3_2_bn (BatchNormal (None, 14, 14, 256) 1024
['conv4_block3_2_conv[0][0]']
ization)

conv4_block3_2_relu (Activatio (None, 14, 14, 256) 0
['conv4_block3_2_bn[0][0]']
n)

conv4_block3_3_conv (Conv2D) (None, 14, 14, 1024 263168

```

```

['conv4_block3_2_relu[0][0]']
    )

    conv4_block3_3_bn (BatchNormal (None, 14, 14, 1024 4096
['conv4_block3_3_conv[0][0]']
    ization)
    )

    conv4_block3_add (Add) (None, 14, 14, 1024 0
['conv4_block2_out[0][0]',
    )
    'conv4_block3_3_bn[0][0]']

    conv4_block3_out (Activation) (None, 14, 14, 1024 0
['conv4_block3_add[0][0]']
    )

    conv4_block4_1_conv (Conv2D) (None, 14, 14, 256) 262400
['conv4_block3_out[0][0]']

    conv4_block4_1_bn (BatchNormal (None, 14, 14, 256) 1024
['conv4_block4_1_conv[0][0]']
    ization)

    conv4_block4_1_relu (Activatio (None, 14, 14, 256) 0
['conv4_block4_1_bn[0][0]']
    n)

    conv4_block4_2_conv (Conv2D) (None, 14, 14, 256) 590080
['conv4_block4_1_relu[0][0]']

    conv4_block4_2_bn (BatchNormal (None, 14, 14, 256) 1024
['conv4_block4_2_conv[0][0]']
    ization)

    conv4_block4_2_relu (Activatio (None, 14, 14, 256) 0
['conv4_block4_2_bn[0][0]']
    n)

    conv4_block4_3_conv (Conv2D) (None, 14, 14, 1024 263168
['conv4_block4_2_relu[0][0]']
    )

    conv4_block4_3_bn (BatchNormal (None, 14, 14, 1024 4096
['conv4_block4_3_conv[0][0]']
    ization)
    )

    conv4_block4_add (Add) (None, 14, 14, 1024 0
['conv4_block3_out[0][0]',

```

```

)
'conv4_block4_3_bn[0][0]']

conv4_block4_out (Activation) (None, 14, 14, 1024 0
['conv4_block4_add[0][0]']
)

conv4_block5_1_conv (Conv2D) (None, 14, 14, 256) 262400
['conv4_block4_out[0][0]']

conv4_block5_1_bn (BatchNormal (None, 14, 14, 256) 1024
['conv4_block5_1_conv[0][0]']
ization)

conv4_block5_1_relu (Activatio (None, 14, 14, 256) 0
['conv4_block5_1_bn[0][0]']
n)

conv4_block5_2_conv (Conv2D) (None, 14, 14, 256) 590080
['conv4_block5_1_relu[0][0]']

conv4_block5_2_bn (BatchNormal (None, 14, 14, 256) 1024
['conv4_block5_2_conv[0][0]']
ization)

conv4_block5_2_relu (Activatio (None, 14, 14, 256) 0
['conv4_block5_2_bn[0][0]']
n)

conv4_block5_3_conv (Conv2D) (None, 14, 14, 1024 263168
['conv4_block5_2_relu[0][0]']
)

conv4_block5_3_bn (BatchNormal (None, 14, 14, 1024 4096
['conv4_block5_3_conv[0][0]']
ization)
)

conv4_block5_add (Add) (None, 14, 14, 1024 0
['conv4_block4_out[0][0]',
)
'conv4_block5_3_bn[0][0]']

conv4_block5_out (Activation) (None, 14, 14, 1024 0
['conv4_block5_add[0][0]']
)

conv4_block6_1_conv (Conv2D) (None, 14, 14, 256) 262400
['conv4_block5_out[0][0]']

```



```

conv4_block6_1_bn (BatchNormal (None, 14, 14, 256) 1024
['conv4_block6_1_conv[0][0]']
ization)

conv4_block6_1_relu (Activatio (None, 14, 14, 256) 0
['conv4_block6_1_bn[0][0]']
n)

conv4_block6_2_conv (Conv2D) (None, 14, 14, 256) 590080
['conv4_block6_1_relu[0][0]']

conv4_block6_2_bn (BatchNormal (None, 14, 14, 256) 1024
['conv4_block6_2_conv[0][0]']
ization)

conv4_block6_2_relu (Activatio (None, 14, 14, 256) 0
['conv4_block6_2_bn[0][0]']
n)

conv4_block6_3_conv (Conv2D) (None, 14, 14, 1024 263168
['conv4_block6_2_relu[0][0]']
)

conv4_block6_3_bn (BatchNormal (None, 14, 14, 1024 4096
['conv4_block6_3_conv[0][0]']
ization)
)

conv4_block6_add (Add) (None, 14, 14, 1024 0
['conv4_block5_out[0][0]',
)
'conv4_block6_3_bn[0][0]']

conv4_block6_out (Activation) (None, 14, 14, 1024 0
['conv4_block6_add[0][0]']
)

conv5_block1_1_conv (Conv2D) (None, 7, 7, 512) 524800
['conv4_block6_out[0][0]']

conv5_block1_1_bn (BatchNormal (None, 7, 7, 512) 2048
['conv5_block1_1_conv[0][0]']
ization)

conv5_block1_1_relu (Activatio (None, 7, 7, 512) 0
['conv5_block1_1_bn[0][0]']
n)

```

| | | |
|---|--------------------|---------|
| conv5_block1_2_conv (Conv2D) ['conv5_block1_1_relu[0][0]'] | (None, 7, 7, 512) | 2359808 |
| conv5_block1_2_bn (BatchNormal ization) | (None, 7, 7, 512) | 2048 |
| conv5_block1_2_relu (Activatio n) | (None, 7, 7, 512) | 0 |
| conv5_block1_0_conv (Conv2D) ['conv4_block6_out[0][0]'] | (None, 7, 7, 2048) | 2099200 |
| conv5_block1_3_conv (Conv2D) ['conv5_block1_2_relu[0][0]'] | (None, 7, 7, 2048) | 1050624 |
| conv5_block1_0_bn (BatchNormal ization) | (None, 7, 7, 2048) | 8192 |
| conv5_block1_3_bn (BatchNormal ization) | (None, 7, 7, 2048) | 8192 |
| conv5_block1_add (Add) ['conv5_block1_0_bn[0][0]', 'conv5_block1_3_bn[0][0]'] | (None, 7, 7, 2048) | 0 |
| conv5_block1_out (Activation) ['conv5_block1_add[0][0]'] | (None, 7, 7, 2048) | 0 |
| conv5_block2_1_conv (Conv2D) ['conv5_block1_out[0][0]'] | (None, 7, 7, 512) | 1049088 |
| conv5_block2_1_bn (BatchNormal ization) | (None, 7, 7, 512) | 2048 |
| conv5_block2_1_relu (Activatio n) | (None, 7, 7, 512) | 0 |
| conv5_block2_2_conv (Conv2D) ['conv5_block2_1_relu[0][0]'] | (None, 7, 7, 512) | 2359808 |
| conv5_block2_2_bn (BatchNormal ization) | (None, 7, 7, 512) | 2048 |

```

ization)

conv5_block2_2_relu (Activation) (None, 7, 7, 512) 0
['conv5_block2_2_bn[0][0]']
n)

conv5_block2_3_conv (Conv2D) (None, 7, 7, 2048) 1050624
['conv5_block2_2_relu[0][0]']

conv5_block2_3_bn (BatchNormal (None, 7, 7, 2048) 8192
['conv5_block2_3_conv[0][0]']
ization)

conv5_block2_add (Add) (None, 7, 7, 2048) 0
['conv5_block1_out[0][0]',
'conv5_block2_3_bn[0][0]']

conv5_block2_out (Activation) (None, 7, 7, 2048) 0
['conv5_block2_add[0][0]']

conv5_block3_1_conv (Conv2D) (None, 7, 7, 512) 1049088
['conv5_block2_out[0][0]']

conv5_block3_1_bn (BatchNormal (None, 7, 7, 512) 2048
['conv5_block3_1_conv[0][0]']
ization)

conv5_block3_1_relu (Activation) (None, 7, 7, 512) 0
['conv5_block3_1_bn[0][0]']
n)

conv5_block3_2_conv (Conv2D) (None, 7, 7, 512) 2359808
['conv5_block3_1_relu[0][0]']

conv5_block3_2_bn (BatchNormal (None, 7, 7, 512) 2048
['conv5_block3_2_conv[0][0]']
ization)

conv5_block3_2_relu (Activation) (None, 7, 7, 512) 0
['conv5_block3_2_bn[0][0]']
n)

conv5_block3_3_conv (Conv2D) (None, 7, 7, 2048) 1050624
['conv5_block3_2_relu[0][0]']

conv5_block3_3_bn (BatchNormal (None, 7, 7, 2048) 8192
['conv5_block3_3_conv[0][0]']
ization)

```

```

conv5_block3_add (Add)          (None, 7, 7, 2048)    0
['conv5_block2_out[0][0]',
'conv5_block3_3_bn[0][0]']

conv5_block3_out (Activation)   (None, 7, 7, 2048)    0
['conv5_block3_add[0][0]']

avg_pool (GlobalAveragePooling (None, 2048)      0
['conv5_block3_out[0][0]']
2D)

predictions (Dense)            (None, 1000)          2049000
['avg_pool[0][0]']

```

```

=====
Total params: 25,636,712
Trainable params: 25,583,592
Non-trainable params: 53,120
-----

```

```

[7]: flowerPower='http://download.tensorflow.org/example_images/flower_photos.tgz'
data_dirs=tf.keras.utils.get_file('flower_photos',
                                   origin=flowerPower,
                                   untar=True)
data_dirs=pathlib.Path(data_dirs)

```

```

[46]: from contextlib import redirect_stdout

summaryFile = myimages.joinpath('Assignment6.3ModelSummary.txt')
with open(summaryFile, 'w') as f:
    with redirect_stdout(f):
        model.summary()

```

```

[8]: import os
import numpy as np
from keras.preprocessing.image import ImageDataGenerator
import matplotlib.pyplot as plt
import PIL
import pathlib
from pathlib import Path
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.python.keras.layers import Dense, Flatten

```

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.optimizers import Adam

import warnings
warnings.filterwarnings('ignore')
```

```
[9]: print(data_dirs)
```

/home/jovyan/.keras/datasets/flower_photos

```
[20]: roses=list(data_dirs.glob('roses/*'))
      print(roses[5])
      PIL.Image.open(str(roses[5]))
```

/home/jovyan/.keras/datasets/flower_photos/roses/13342823005_16d3df58df_n.jpg

[20]:



```
[50]: print(roses[7])
      PIL.Image.open(str(roses[7]))
```

/home/jovyan/.keras/datasets/flower_photos/roses/2735666555_01d53e74fe.jpg

[50]:



```
[51]: sunflower=list(data_dirs.glob('sunflowers/*'))  
      print(sunflower[7])  
      PIL.Image.open(str(sunflower[7]))
```

```
/home/jovyan/.keras/datasets/flower_photos/sunflowers/22203670478_9ec5c2700b_n.j  
pg
```

```
[51]:
```



```
[52]: print(sunflower[9])  
      PIL.Image.open(str(sunflower[9]))
```

/home/jovyan/.keras/datasets/flower_photos/sunflowers/3594967811_697184b026_n.jpg

[52]:



```
[10]: img_height, img_width=180,180  
      batch_size=10
```

```

train_ds=tf.keras.preprocessing.image_dataset_from_directory(data_dirs,
                                                             validation_split=0.
↪2,
                                                             subset='training',
                                                             seed=123,
                                                             ↵
↪label_mode='categorical',
                                                             ↵
↪image_size=(img_height,img_width),
                                                             ↵
↪batch_size=batch_size)

```

Found 3670 files belonging to 5 classes.
Using 2936 files for training.

```

[11]: val_ds=tf.keras.preprocessing.image_dataset_from_directory(data_dirs,
                                                                validation_split=0.2,
                                                                subset='validation',
                                                                seed=123,
                                                                ↵
↪label_mode='categorical',
                                                                ↵
↪image_size=(img_height,img_width),
                                                                batch_size=batch_size)

```

Found 3670 files belonging to 5 classes.
Using 734 files for validation.

```

[12]: class_names=train_ds.class_names
print(class_names)

```

```
['daisy', 'dandelion', 'roses', 'sunflowers', 'tulips']
```

```

[13]: #Training ResNet50
#https://keras.io/api/applications/resnet/#resnet50-function
resnet_model=Sequential()
pretrained_model=tf.keras.applications.ResNet50(
    include_top=False,
    input_shape=(180,180,3),
    weights="imagenet")
for layer in pretrained_model.layers:
    layer.trainable=False

```

```

[14]: resnet_model.add(pretrained_model)
resnet_model.add(Flatten())
resnet_model.add(Dense(512,activation='relu'))
resnet_model.add(Dense(5,activation='softmax'))

```



```
[15]: resnet_model.summary()
```

```
Model: "sequential"
```

| Layer (type) | Output Shape | Param # |
|----------------------------------|--------------------|----------|
| resnet50 (Functional) | (None, 6, 6, 2048) | 23587712 |
| module_wrapper (ModuleWrapper) | (None, 73728) | 0 |
| module_wrapper_1 (ModuleWrapper) | (None, 512) | 37749248 |
| module_wrapper_2 (ModuleWrapper) | (None, 5) | 2565 |

```
=====  
Total params: 61,339,525  
Trainable params: 37,751,813  
Non-trainable params: 23,587,712  
=====
```

```
[22]: resnet_model.compile(optimizer=Adam(learning_rate=0.001),  
                           loss='categorical_crossentropy',  
                           metrics=['accuracy'])
```

```
[23]: epochs=10  
history=resnet_model.fit(train_ds,  
                          validation_data=val_ds,  
                          epochs=epochs)
```

```
Epoch 1/10
```

```
2023-07-15 19:02:54.020120: I tensorflow/core/common_runtime/executor.cc:1197]  
[/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an  
error and you can ignore this message): INVALID_ARGUMENT: You must feed a value  
for placeholder tensor 'Placeholder/_4' with dtype int32 and shape [2936]  
[[{{node Placeholder/_4}}]]
```

```
2023-07-15 19:02:54.020786: I tensorflow/core/common_runtime/executor.cc:1197]  
[/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an  
error and you can ignore this message): INVALID_ARGUMENT: You must feed a value  
for placeholder tensor 'Placeholder/_4' with dtype int32 and shape [2936]  
[[{{node Placeholder/_4}}]]
```

```
294/294 [=====] - ETA: 0s - loss: 4.9349 - accuracy:  
0.7469
```

```
2023-07-15 19:04:50.916362: I tensorflow/core/common_runtime/executor.cc:1197]
```

```
[/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'Placeholder/_0' with dtype string and shape [734]
```

```
[[{{node Placeholder/_0}}]]
```

```
2023-07-15 19:04:50.917084: I tensorflow/core/common_runtime/executor.cc:1197]
```

```
[/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'Placeholder/_0' with dtype string and shape [734]
```

```
[[{{node Placeholder/_0}}]]
```

```
294/294 [=====] - 131s 429ms/step - loss: 4.9349 - accuracy: 0.7469 - val_loss: 0.5286 - val_accuracy: 0.8297
```

```
Epoch 2/10
```

```
294/294 [=====] - 123s 418ms/step - loss: 0.2536 - accuracy: 0.9217 - val_loss: 0.5723 - val_accuracy: 0.8610
```

```
Epoch 3/10
```

```
294/294 [=====] - 122s 416ms/step - loss: 0.1361 - accuracy: 0.9588 - val_loss: 0.6972 - val_accuracy: 0.8515
```

```
Epoch 4/10
```

```
294/294 [=====] - 123s 418ms/step - loss: 0.0574 - accuracy: 0.9806 - val_loss: 0.5915 - val_accuracy: 0.8692
```

```
Epoch 5/10
```

```
294/294 [=====] - 122s 416ms/step - loss: 0.0337 - accuracy: 0.9884 - val_loss: 0.7609 - val_accuracy: 0.8406
```

```
Epoch 6/10
```

```
294/294 [=====] - 123s 418ms/step - loss: 0.0711 - accuracy: 0.9775 - val_loss: 0.9054 - val_accuracy: 0.8420
```

```
Epoch 7/10
```

```
294/294 [=====] - 123s 417ms/step - loss: 0.1699 - accuracy: 0.9581 - val_loss: 1.1073 - val_accuracy: 0.8433
```

```
Epoch 8/10
```

```
294/294 [=====] - 122s 415ms/step - loss: 0.1299 - accuracy: 0.9724 - val_loss: 1.4985 - val_accuracy: 0.8365
```

```
Epoch 9/10
```

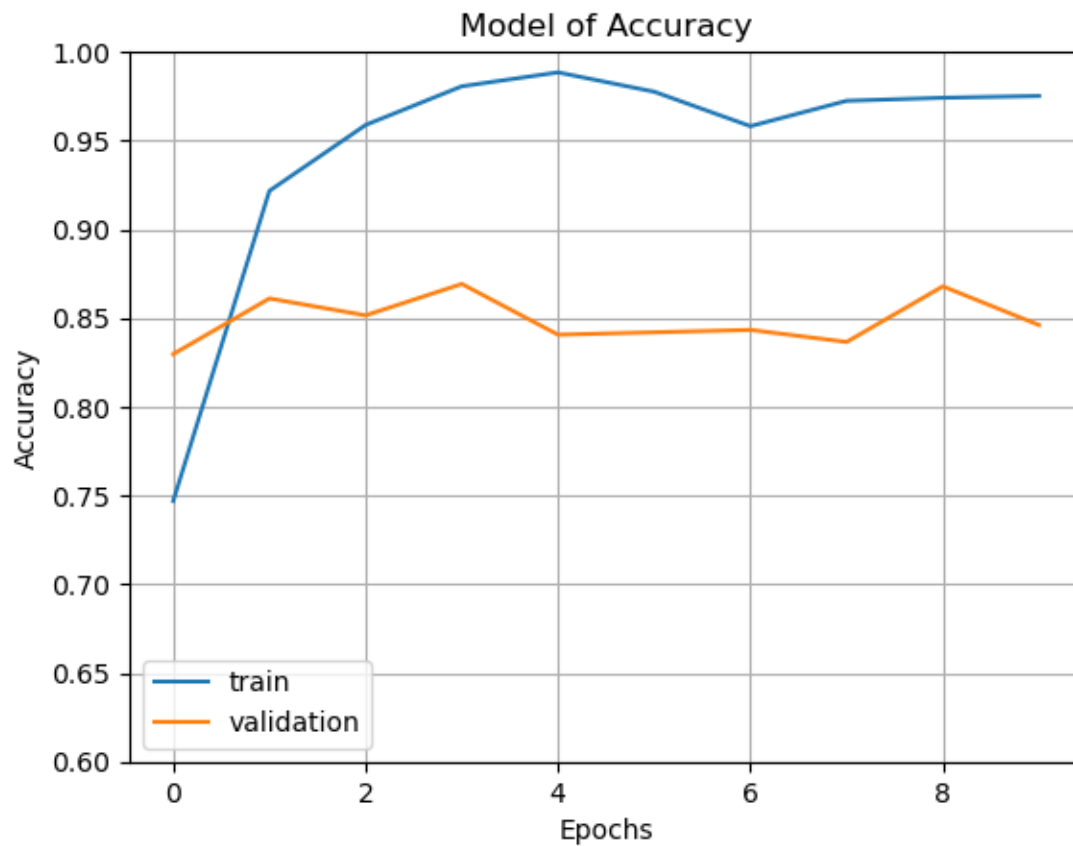
```
294/294 [=====] - 121s 412ms/step - loss: 0.1155 - accuracy: 0.9741 - val_loss: 1.0236 - val_accuracy: 0.8678
```

```
Epoch 10/10
```

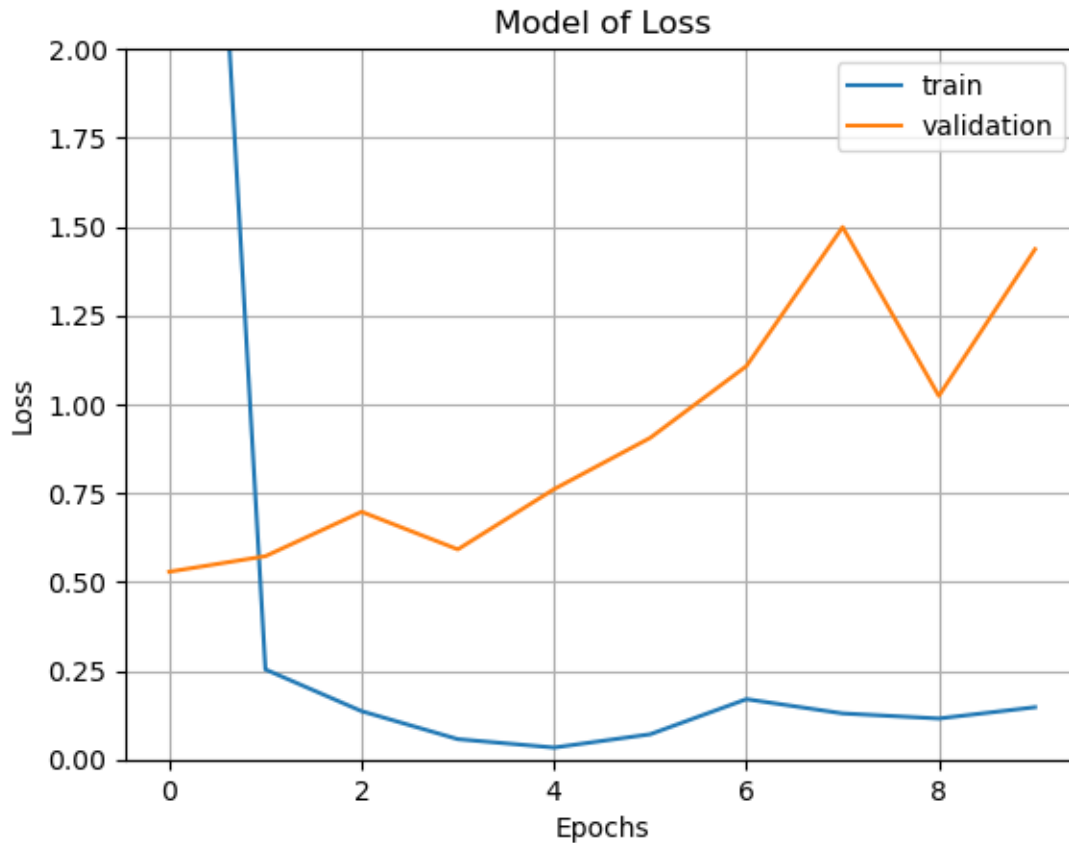
```
294/294 [=====] - 121s 412ms/step - loss: 0.1473 - accuracy: 0.9751 - val_loss: 1.4362 - val_accuracy: 0.8460
```

```
[24]: fig1=plt.gcf()
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.axis(ymin=0.6,ymax=1)
plt.grid()
plt.title('Model of Accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epochs')
```

```
plt.legend(['train', 'validation'])  
plt.show()
```



```
[25]: plt.plot(history.history['loss'])  
plt.plot(history.history['val_loss'])  
plt.axis(ymin=0,ymax=2)  
plt.grid()  
plt.title('Model of Loss')  
plt.ylabel('Loss')  
plt.xlabel('Epochs')  
plt.legend(['train', 'validation'])  
plt.show()
```



```
[17]: !pip install opencv-python --quiet
```

```
[26]: #Making Predictions with the model
import cv2
image=cv2.imread(str(roses[2]))
image_resized=cv2.resize(image,(img_height,img_width))
image=np.expand_dims(image_resized, axis=0)
print(image.shape)
```

```
(1, 180, 180, 3)
```

```
[27]: pred=resnet_model.predict(image)
print(pred)
```

```
1/1 [=====] - 2s 2s/step
[[8.9446425e-16 3.5474751e-21 1.0000000e+00 2.1229429e-16 1.8048296e-10]]
```

```
[28]: output_class=class_names[np.argmax(pred)]
print("The prediction is for the class of ", output_class)
```

```
The prediction is for the class of  roses
```

```
[29]: under_class=class_names[np.argmax(pred)]  
      print("The prediction for the under class is a ", under_class)
```

The prediction for the under class is a dandelion

```
[30]: homebase=Path('/home/jovyan/DSC650/dsc650/')  
  
      myimages=Path('/home/jovyan/DSC650/dsc650/assignments/assignment06/').  
          ↪joinpath('images')  
  
      myimages.mkdir(parents=True, exist_ok=True)
```

```
[32]: from contextlib import redirect_stdout  
  
      summaryFile = myimages.joinpath('Assignment6.3Modelimages.txt')  
      with open(summaryFile, 'w') as f:  
          with redirect_stdout(f):  
              train_ds
```

```
[33]: homebase=Path('/home/jovyan/DSC650/dsc650/')  
  
      mypredictions=Path('/home/jovyan/DSC650/dsc650/assignments/assignment06/results/  
          ↪').joinpath('predictions')  
  
      mypredictions.mkdir(parents=True, exist_ok=True)
```

```
[ ]:
```

```
[35]: from contextlib import redirect_stdout  
  
      summarypreds = mypredictions.joinpath('resnet50')  
      with open(summarypreds, 'w') as f:  
          with redirect_stdout(f):  
              pred
```

```
[ ]:
```