lwdfnsgve

August 12, 2023

[1]:
```
!mkdir -p ~/.kaggle
!cp kaggle.json ~/.kaggle/
```

[2]:
```
!kaggle datasets download -d kausthubkannan/
↪5-flower-types-classification-dataset
```

```
Warning: Your Kaggle API key is readable by other users on this system! To fix
this, you can run 'chmod 600 /root/.kaggle/kaggle.json'
Downloading 5-flower-types-classification-dataset.zip to /content
100% 241M/242M [00:10<00:00, 29.7MB/s]
100% 242M/242M [00:10<00:00, 25.1MB/s]
```

[3]:
```
import zipfile
zip_ref = zipfile.ZipFile('/content/5-flower-types-classification-dataset.zip',␣
↪'r')
zip_ref.extractall('/content')
zip_ref.close()
```

[4]:
```
!pip install split-folders
```

```
Collecting split-folders
  Downloading split_folders-0.5.1-py3-none-any.whl (8.4 kB)
Installing collected packages: split-folders
Successfully installed split-folders-0.5.1
```

[5]:
```
import splitfolders
input_folder ="/content/flower_images"
splitfolders.ratio(input_folder,output="/content",seed=42,ratio=(.7,.2,.1))
```

```
Copying files: 5000 files [00:02, 2181.83 files/s]
```

[6]:
```
import os
 import numpy as np
 import matplotlib.pyplot as plt
 import tensorflow as tf
 import keras
```

[7]:
```
# Without Data Augmentation
# train_data = keras.utils.image_dataset_from_directory(
#     directory = '/content/train',
#     labels='inferred',
#     label_mode = 'categorical',
#     batch_size=32,
#     image_size=(256,256)
# )

# validation_data = keras.utils.image_dataset_from_directory(
#     directory = '/content/val',
#     labels='inferred',
#     label_mode = 'categorical',
#     batch_size=32,
#     image_size=(256,256)
# )

# test_data = keras.utils.image_dataset_from_directory(
#     directory = '/content/test',
#     labels='inferred',
#     label_mode = 'categorical',
#     batch_size=32,
#     image_size=(256,256)
# )

# def process(image,label):
#     image = tf.cast(image/255. ,tf.float32)
#     return image,label

# train_data = train_data.map(process)
# validation_data = validation_data.map(process)
# test_data = test_data.map(process)
```

[23]:
```
# With Data Augmentation
from keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(rescale = 1./255,
                                   rotation_range=40,
                                   width_shift_range =0.3,
                                   height_shift_range=0.3,
                                   shear_range=0.3,
                                   zoom_range=0.3,
                                   horizontal_flip=True,
                                   fill_mode="reflect")

test_datagen = ImageDataGenerator(rescale=1./255)
```

```
train_generator = train_datagen.flow_from_directory(
    "/content/train",␣
 ↪target_size=(256,256),batch_size=50,class_mode="categorical"
)

validation_generator = test_datagen.flow_from_directory(
    "/content/test",␣
 ↪target_size=(256,256),batch_size=50,class_mode="categorical"
)
```

```
Found 3500 images belonging to 5 classes.
Found 500 images belonging to 5 classes.
```

[24]:
```
model = keras.models.Sequential()

model.add(keras.layers.
 ↪Conv2D(32,(3,3),activation="relu",input_shape=(256,256,3)))
model.add(keras.layers.MaxPooling2D((2,2)))

model.add(keras.layers.Conv2D(64,(3,3),activation="relu"))
model.add(keras.layers.MaxPooling2D((2,2)))

model.add(keras.layers.Conv2D(96,(3,3),activation="relu"))
model.add(keras.layers.MaxPooling2D((2,2)))

model.add(keras.layers.Conv2D(128,(3,3),activation="relu"))
model.add(keras.layers.MaxPooling2D((2,2)))

model.add(keras.layers.Flatten())

model.add(keras.layers.Dense(512,activation="relu"))

model.add(keras.layers.Dense(5, activation="softmax"))

model.summary()
```

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_12 (Conv2D)          (None, 254, 254, 32)      896

 max_pooling2d_12 (MaxPoolin  (None, 127, 127, 32)     0
 g2D)

 conv2d_13 (Conv2D)          (None, 125, 125, 64)      18496
```

```
max_pooling2d_13 (MaxPoolin  (None, 62, 62, 64)       0
g2D)

conv2d_14 (Conv2D)           (None, 60, 60, 96)        55392

max_pooling2d_14 (MaxPoolin  (None, 30, 30, 96)       0
g2D)

conv2d_15 (Conv2D)           (None, 28, 28, 128)       110720

max_pooling2d_15 (MaxPoolin  (None, 14, 14, 128)      0
g2D)

flatten_3 (Flatten)          (None, 25088)             0

dense_6 (Dense)              (None, 512)               12845568

dense_7 (Dense)              (None, 5)                 2565

=================================================================
Total params: 13,033,637
Trainable params: 13,033,637
Non-trainable params: 0
_________________________________________________________________
```

[25]:
```
model.compile(optimizer=keras.optimizers.Adam(),␣
 ↪loss="categorical_crossentropy", metrics=["acc"])

history = model.fit_generator(train_generator,
                              steps_per_epoch=70,
                              epochs=33,
                              validation_data = validation_generator,
                              validation_steps=10,
                              verbose=1)
```

```
<ipython-input-25-dee00ba5f506>:3: UserWarning: `Model.fit_generator` is
deprecated and will be removed in a future version. Please use `Model.fit`,
which supports generators.
  history = model.fit_generator(train_generator,

Epoch 1/33
70/70 [==============================] - 69s 968ms/step - loss: 1.3557 - acc:
0.4271 - val_loss: 1.1354 - val_acc: 0.5580
Epoch 2/33
70/70 [==============================] - 68s 967ms/step - loss: 1.1710 - acc:
0.5203 - val_loss: 1.0885 - val_acc: 0.5540
Epoch 3/33
```

```
70/70 [==============================] - 67s 957ms/step - loss: 1.0886 - acc:
0.5646 - val_loss: 1.0916 - val_acc: 0.5940
Epoch 4/33
70/70 [==============================] - 69s 982ms/step - loss: 1.0710 - acc:
0.5711 - val_loss: 1.0614 - val_acc: 0.5840
Epoch 5/33
70/70 [==============================] - 66s 944ms/step - loss: 1.0084 - acc:
0.5934 - val_loss: 0.9638 - val_acc: 0.6240
Epoch 6/33
70/70 [==============================] - 67s 955ms/step - loss: 0.9802 - acc:
0.6143 - val_loss: 0.9428 - val_acc: 0.6180
Epoch 7/33
70/70 [==============================] - 67s 956ms/step - loss: 0.9340 - acc:
0.6343 - val_loss: 0.9779 - val_acc: 0.6340
Epoch 8/33
70/70 [==============================] - 67s 953ms/step - loss: 0.9322 - acc:
0.6317 - val_loss: 0.9101 - val_acc: 0.6420
Epoch 9/33
70/70 [==============================] - 66s 948ms/step - loss: 0.8859 - acc:
0.6571 - val_loss: 0.8244 - val_acc: 0.6700
Epoch 10/33
70/70 [==============================] - 66s 939ms/step - loss: 0.8715 - acc:
0.6623 - val_loss: 0.8746 - val_acc: 0.6840
Epoch 11/33
70/70 [==============================] - 65s 927ms/step - loss: 0.8444 - acc:
0.6743 - val_loss: 0.7869 - val_acc: 0.6880
Epoch 12/33
70/70 [==============================] - 66s 940ms/step - loss: 0.8504 - acc:
0.6709 - val_loss: 0.7885 - val_acc: 0.6900
Epoch 13/33
70/70 [==============================] - 65s 931ms/step - loss: 0.8215 - acc:
0.6826 - val_loss: 0.8325 - val_acc: 0.6780
Epoch 14/33
70/70 [==============================] - 67s 958ms/step - loss: 0.7975 - acc:
0.7029 - val_loss: 0.7597 - val_acc: 0.7240
Epoch 15/33
70/70 [==============================] - 65s 922ms/step - loss: 0.7724 - acc:
0.7026 - val_loss: 0.7266 - val_acc: 0.7140
Epoch 16/33
70/70 [==============================] - 67s 960ms/step - loss: 0.7589 - acc:
0.7071 - val_loss: 0.7574 - val_acc: 0.6980
Epoch 17/33
70/70 [==============================] - 66s 942ms/step - loss: 0.7279 - acc:
0.7234 - val_loss: 0.7108 - val_acc: 0.7260
Epoch 18/33
70/70 [==============================] - 66s 944ms/step - loss: 0.7299 - acc:
0.7300 - val_loss: 0.6892 - val_acc: 0.7300
Epoch 19/33
```

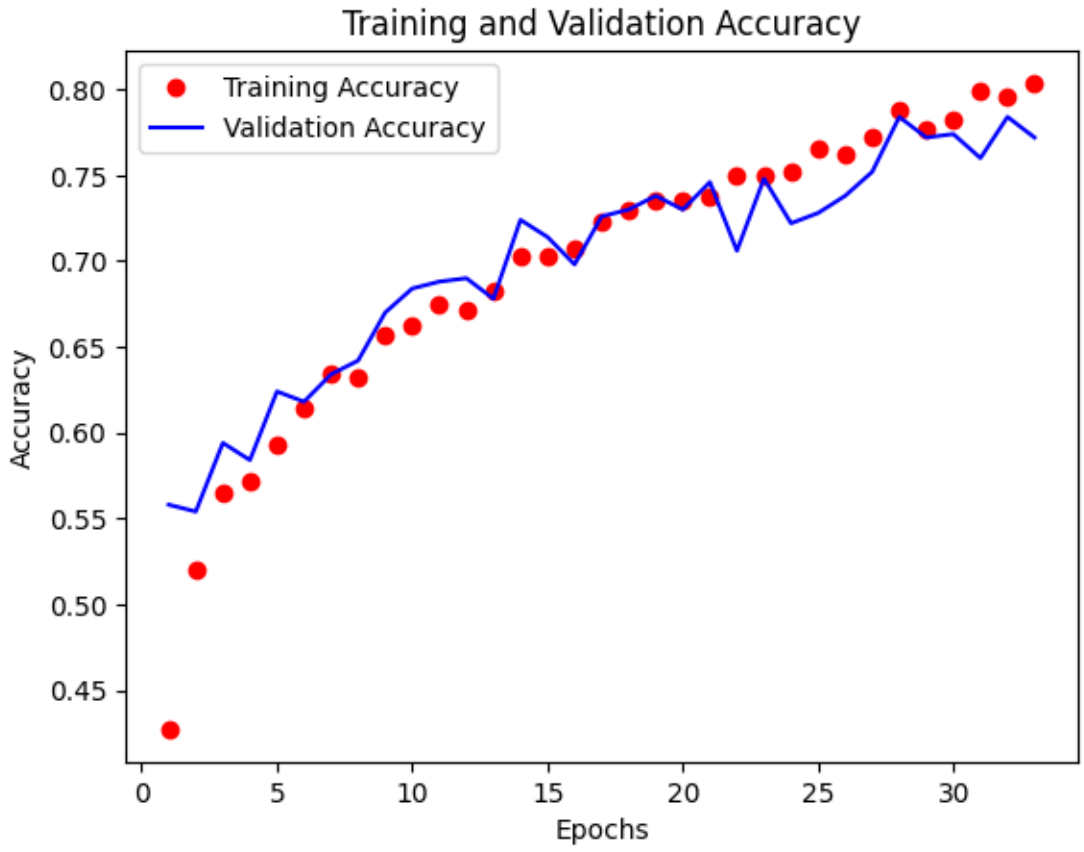
```
70/70 [==============================] - 65s 928ms/step - loss: 0.6888 - acc:
0.7351 - val_loss: 0.6864 - val_acc: 0.7380
Epoch 20/33
70/70 [==============================] - 66s 942ms/step - loss: 0.6954 - acc:
0.7354 - val_loss: 0.7172 - val_acc: 0.7300
Epoch 21/33
70/70 [==============================] - 66s 941ms/step - loss: 0.6805 - acc:
0.7371 - val_loss: 0.6576 - val_acc: 0.7460
Epoch 22/33
70/70 [==============================] - 67s 959ms/step - loss: 0.6600 - acc:
0.7494 - val_loss: 0.6888 - val_acc: 0.7060
Epoch 23/33
70/70 [==============================] - 65s 925ms/step - loss: 0.6594 - acc:
0.7497 - val_loss: 0.6166 - val_acc: 0.7480
Epoch 24/33
70/70 [==============================] - 66s 937ms/step - loss: 0.6411 - acc:
0.7526 - val_loss: 0.7348 - val_acc: 0.7220
Epoch 25/33
70/70 [==============================] - 66s 938ms/step - loss: 0.6333 - acc:
0.7654 - val_loss: 0.6782 - val_acc: 0.7280
Epoch 26/33
70/70 [==============================] - 66s 937ms/step - loss: 0.6189 - acc:
0.7626 - val_loss: 0.6501 - val_acc: 0.7380
Epoch 27/33
70/70 [==============================] - 68s 968ms/step - loss: 0.5864 - acc:
0.7717 - val_loss: 0.6133 - val_acc: 0.7520
Epoch 28/33
70/70 [==============================] - 67s 955ms/step - loss: 0.5546 - acc:
0.7880 - val_loss: 0.5881 - val_acc: 0.7840
Epoch 29/33
70/70 [==============================] - 65s 933ms/step - loss: 0.5856 - acc:
0.7769 - val_loss: 0.6131 - val_acc: 0.7720
Epoch 30/33
70/70 [==============================] - 67s 951ms/step - loss: 0.5577 - acc:
0.7823 - val_loss: 0.5905 - val_acc: 0.7740
Epoch 31/33
70/70 [==============================] - 66s 943ms/step - loss: 0.5473 - acc:
0.7991 - val_loss: 0.5845 - val_acc: 0.7600
Epoch 32/33
70/70 [==============================] - 64s 921ms/step - loss: 0.5233 - acc:
0.7954 - val_loss: 0.5803 - val_acc: 0.7840
Epoch 33/33
70/70 [==============================] - 68s 962ms/step - loss: 0.5354 - acc:
0.8037 - val_loss: 0.5527 - val_acc: 0.7720
```

[26]:
```
model.save("/content/Model")
```

```
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op,
```

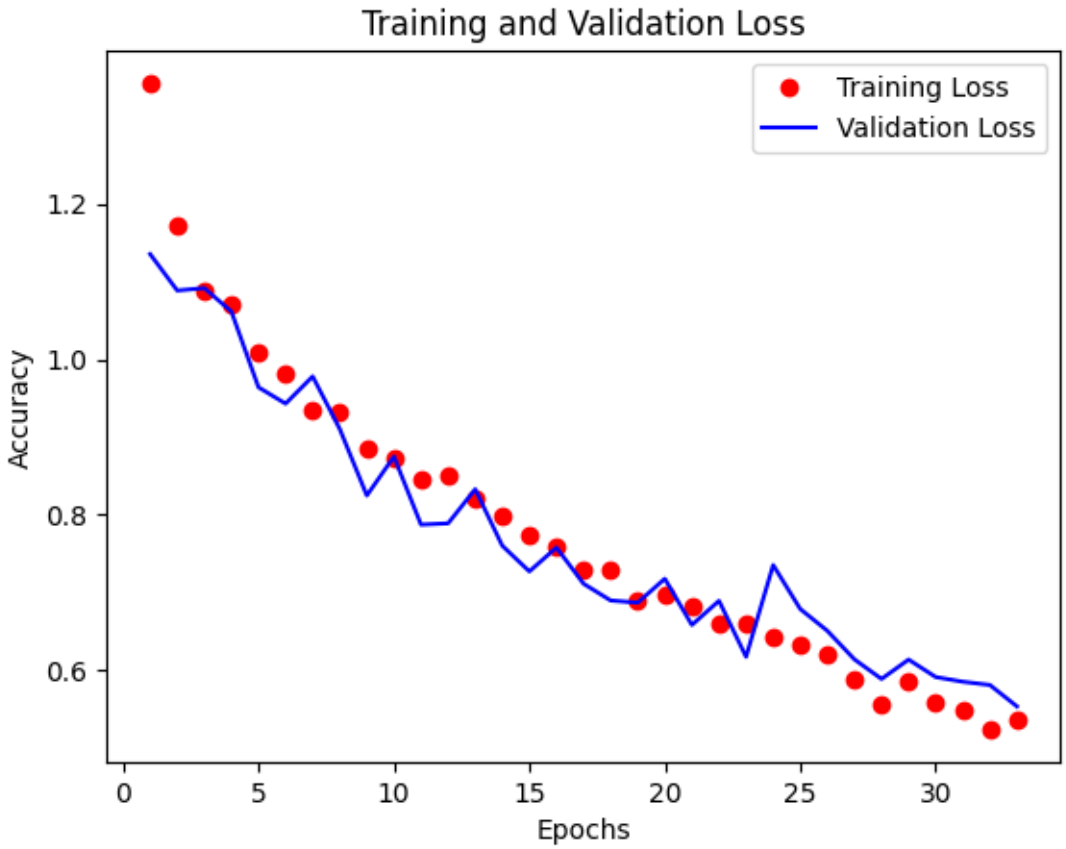
```
_jit_compiled_convolution_op, _jit_compiled_convolution_op,
_jit_compiled_convolution_op while saving (showing 4 of 4). These functions will
not be directly callable after loading.
```

[27]:
```python
# Plotting Metrics
import matplotlib.pyplot as plt
epochs = range(1,34)
acc = history.history["acc"]
val_acc = history.history["val_acc"]
plt.title("Training and Validation Accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.plot(epochs,acc,'ro',label="Training Accuracy")
plt.plot(epochs,val_acc,'b',label="Validation Accuracy")
plt.legend()
plt.show()
```



[28]:
```python
loss = history.history["loss"]
val_loss = history.history["val_loss"]
plt.title("Training and Validation Loss")
```

```
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.plot(epochs,loss,'ro',label="Training Loss")
plt.plot(epochs,val_loss,'b',label="Validation Loss")
plt.legend()
plt.show()
```



[29]:
```
test_data = keras.utils.image_dataset_from_directory(
    directory = '/content/test',
    labels='inferred',
    label_mode = 'categorical',
    batch_size=50,
    image_size=(256,256)
)
```

```
Found 500 files belonging to 5 classes.
```

[30]:
```
score = model.evaluate(test_data, verbose = 1)

print('Test loss:', score[0])
```

```
print('Test accuracy:', score[1])
```

```
16/16 [==============================] - 3s 73ms/step - loss: 144.4908 - acc:
0.5440
Test loss: 144.49082946777344
Test accuracy: 0.5440000295639038
```

[31]:
```
from PIL import Image
```

[32]:
```
img1 = Image.open("/content/lilly.jpg")
img2 = Image.open("/content/orchid.jpg")
img3 = Image.open("/content/tulip.jpg")
```

[33]:
```
img1
```

[33]:

[34]:
```
img2
```

[34]:

[35]:
```
img3
```

[35]:

[41]:
```
for i,img in enumerate([img1,img2,img3]):
  img = img.resize((256,256))
  img = np.array(img)/255.0
  img = np.expand_dims(img, axis=0)
  pred= model.predict(img,verbose=0)
  labels={0:"Lilly",1:"Lotus",2:"Orchid",3:"Sunflower",4:"Tulip"}
  print("Img{} belongs to class {}".format(i,labels[np.argmax(pred)]))
```

```
Img0 belongs to class Lilly
Img1 belongs to class Orchid
Img2 belongs to class Tulip
```

[ ]: