Assignment_3

October 6, 2022

Priyanga.S
732119104057

#Unzipping

```
[21]: #!unzip '/content/drive/MyDrive/Flowers-Dataset.zip'
```

```
[5]: from google.colab import drive
     drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).

#Data Augmentation

```
[6]: from tensorflow.keras.preprocessing.image import ImageDataGenerator

     train = ImageDataGenerator(rescale=1./255,
                                zoom_range=0.2,
                                horizontal_flip=True)
     test = ImageDataGenerator(rescale=1./255)
```

```
[7]: xtrain = train.flow_from_directory('/content/flowers',
                                        target_size=(64,64),
                                        class_mode='categorical',
                                        batch_size=100)
```

Found 4317 images belonging to 5 classes.

#Train

```
[8]: from tensorflow.keras.models import Sequential
     from tensorflow.keras.layers import Convolution2D, MaxPooling2D, Dense, Flatten
     from keras.callbacks import EarlyStopping, ReduceLROnPlateau
```

```
[9]: model = Sequential()
     model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))
     model.add(MaxPooling2D((2,2)))
     model.add(Flatten())

     model.add(Dense(400,activation='relu'))
```

```
model.add(Dense(200,activation='relu'))
model.add(Dense(100,activation='relu'))
model.add(Dense(5,activation='softmax'))
```

[10]:
```
model.
 ↪compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
```

[11]:
```
early_stopping = EarlyStopping(monitor='accuracy',
                              patience=3)
reduce_lr = ReduceLROnPlateau(monitor='accuracy',
                             patience=5,
                             factor=0.5,min_lr=0.00001)

callback = [reduce_lr,early_stopping]
```

[12]:
```
model.fit_generator(xtrain,
                    steps_per_epoch = len(xtrain),
                    callbacks=callback,
                    epochs=100)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:4: UserWarning:
`Model.fit_generator` is deprecated and will be removed in a future version.
Please use `Model.fit`, which supports generators.
  after removing the cwd from sys.path.

Epoch 1/100
44/44 [==============================] - 33s 731ms/step - loss: 1.6695 -
accuracy: 0.3665 - lr: 0.0010
Epoch 2/100
44/44 [==============================] - 33s 735ms/step - loss: 1.1799 -
accuracy: 0.5152 - lr: 0.0010
Epoch 3/100
44/44 [==============================] - 34s 774ms/step - loss: 1.0697 -
accuracy: 0.5749 - lr: 0.0010
Epoch 4/100
44/44 [==============================] - 32s 729ms/step - loss: 1.0116 -
accuracy: 0.6002 - lr: 0.0010
Epoch 5/100
44/44 [==============================] - 32s 717ms/step - loss: 0.9683 -
accuracy: 0.6220 - lr: 0.0010
Epoch 6/100
44/44 [==============================] - 33s 747ms/step - loss: 0.9115 -
accuracy: 0.6507 - lr: 0.0010
Epoch 7/100
44/44 [==============================] - 33s 740ms/step - loss: 0.8844 -
accuracy: 0.6690 - lr: 0.0010
Epoch 8/100
44/44 [==============================] - 32s 726ms/step - loss: 0.8282 -
```

```
accuracy: 0.6868 - lr: 0.0010
Epoch 9/100
44/44 [==============================] - 34s 774ms/step - loss: 0.8076 -
accuracy: 0.6963 - lr: 0.0010
Epoch 10/100
44/44 [==============================] - 32s 732ms/step - loss: 0.7729 -
accuracy: 0.7065 - lr: 0.0010
Epoch 11/100
44/44 [==============================] - 34s 775ms/step - loss: 0.7458 -
accuracy: 0.7111 - lr: 0.0010
Epoch 12/100
44/44 [==============================] - 33s 748ms/step - loss: 0.7029 -
accuracy: 0.7329 - lr: 0.0010
Epoch 13/100
44/44 [==============================] - 34s 770ms/step - loss: 0.6885 -
accuracy: 0.7403 - lr: 0.0010
Epoch 14/100
44/44 [==============================] - 34s 780ms/step - loss: 0.6668 -
accuracy: 0.7489 - lr: 0.0010
Epoch 15/100
44/44 [==============================] - 32s 729ms/step - loss: 0.6326 -
accuracy: 0.7619 - lr: 0.0010
Epoch 16/100
44/44 [==============================] - 32s 720ms/step - loss: 0.5727 -
accuracy: 0.7885 - lr: 0.0010
Epoch 17/100
44/44 [==============================] - 32s 736ms/step - loss: 0.5651 -
accuracy: 0.7892 - lr: 0.0010
Epoch 18/100
44/44 [==============================] - 32s 724ms/step - loss: 0.5127 -
accuracy: 0.8089 - lr: 0.0010
Epoch 19/100
44/44 [==============================] - 33s 741ms/step - loss: 0.4903 -
accuracy: 0.8205 - lr: 0.0010
Epoch 20/100
44/44 [==============================] - 34s 768ms/step - loss: 0.5089 -
accuracy: 0.8121 - lr: 0.0010
Epoch 21/100
44/44 [==============================] - 32s 724ms/step - loss: 0.4860 -
accuracy: 0.8237 - lr: 0.0010
Epoch 22/100
44/44 [==============================] - 32s 725ms/step - loss: 0.4565 -
accuracy: 0.8277 - lr: 0.0010
Epoch 23/100
44/44 [==============================] - 32s 723ms/step - loss: 0.4007 -
accuracy: 0.8578 - lr: 0.0010
Epoch 24/100
44/44 [==============================] - 32s 718ms/step - loss: 0.4063 -
```

```
accuracy: 0.8531 - lr: 0.0010
Epoch 25/100
44/44 [==============================] - 32s 722ms/step - loss: 0.3702 -
accuracy: 0.8656 - lr: 0.0010
Epoch 26/100
44/44 [==============================] - 34s 767ms/step - loss: 0.3344 -
accuracy: 0.8814 - lr: 0.0010
Epoch 27/100
44/44 [==============================] - 32s 730ms/step - loss: 0.4270 -
accuracy: 0.8450 - lr: 0.0010
Epoch 28/100
44/44 [==============================] - 32s 720ms/step - loss: 0.3342 -
accuracy: 0.8819 - lr: 0.0010
Epoch 29/100
44/44 [==============================] - 32s 720ms/step - loss: 0.3028 -
accuracy: 0.8934 - lr: 0.0010
Epoch 30/100
44/44 [==============================] - 32s 720ms/step - loss: 0.2766 -
accuracy: 0.9048 - lr: 0.0010
Epoch 31/100
44/44 [==============================] - 32s 720ms/step - loss: 0.2871 -
accuracy: 0.8965 - lr: 0.0010
Epoch 32/100
44/44 [==============================] - 32s 715ms/step - loss: 0.2531 -
accuracy: 0.9129 - lr: 0.0010
Epoch 33/100
44/44 [==============================] - 32s 719ms/step - loss: 0.2481 -
accuracy: 0.9134 - lr: 0.0010
Epoch 34/100
44/44 [==============================] - 32s 723ms/step - loss: 0.2356 -
accuracy: 0.9129 - lr: 0.0010
Epoch 35/100
44/44 [==============================] - 32s 731ms/step - loss: 0.1961 -
accuracy: 0.9321 - lr: 0.0010
Epoch 36/100
44/44 [==============================] - 32s 725ms/step - loss: 0.1961 -
accuracy: 0.9331 - lr: 0.0010
Epoch 37/100
44/44 [==============================] - 34s 767ms/step - loss: 0.2013 -
accuracy: 0.9293 - lr: 0.0010
Epoch 38/100
44/44 [==============================] - 32s 720ms/step - loss: 0.1717 -
accuracy: 0.9412 - lr: 0.0010
Epoch 39/100
44/44 [==============================] - 33s 752ms/step - loss: 0.1784 -
accuracy: 0.9402 - lr: 0.0010
Epoch 40/100
44/44 [==============================] - 32s 723ms/step - loss: 0.1988 -
```

```
accuracy: 0.9331 - lr: 0.0010
Epoch 41/100
44/44 [==============================] - 32s 722ms/step - loss: 0.1743 -
accuracy: 0.9409 - lr: 0.0010
```

[12]: `<keras.callbacks.History at 0x7ff1d2e186d0>`

[13]:
```python
model.save('flower_model.h5')
```

#Test

[14]:
```python
import numpy as np
from tensorflow.keras.preprocessing import image
```

[15]:
```python
Timage = image.load_img('/content/download.jpg',target_size=(64,64))
Timage
```

[15]:

[17]:
```python
T = image.img_to_array(Timage)
T
```

[17]:
```
array([[[ 93., 128.,  48.],
        [ 99., 130.,  63.],
        [105., 133.,  56.],
        ...,
        [ 68., 111.,  29.],
        [ 93., 132.,  51.],
        [107., 138.,  60.]],

       [[131., 157.,  84.],
        [138., 159.,  94.],
        [143., 165.,  80.],
        ...,
        [ 86., 119.,  28.],
        [108., 138.,  48.],
        [115., 143.,  56.]],

       [[146., 170.,  84.],
        [156., 175.,  96.],
        [161., 181.,  86.],
        ...,
```

```
       [124., 141.,  37.],
       [135., 153.,  51.],
       [133., 153.,  54.]],

      ...,

      [[113., 134.,  57.],
       [111., 133.,  50.],
       [100., 125.,  34.],
       ...,
       [ 77., 109.,   0.],
       [ 64.,  99.,   5.],
       [ 46.,  85.,   6.]],

      [[133., 158., 103.],
       [117., 143.,  78.],
       [ 93., 126.,  47.],
       ...,
       [ 79., 108.,   2.],
       [ 65.,  98.,   7.],
       [ 47.,  84.,   7.]],

      [[132., 157., 117.],
       [114., 142.,  91.],
       [ 79., 116.,  49.],
       ...,
       [ 76., 103.,   0.],
       [ 64.,  94.,   6.],
       [ 53.,  87.,  13.]]], dtype=float32)
```

[18]:
```python
T= np.expand_dims(T,axis= 0)
```

[19]:
```python
flowers_list = list(xtrain.class_indices.keys())
flowers_list
```

[19]: ['daisy', 'dandelion', 'rose', 'sunflower', 'tulip']

[20]:
```python
flowers_list[np.argmax(model.predict(T))]
```

[20]: 'sunflower'