

# Assignment\_3

October 11, 2022

Ajeeth Kumar.S  
732119104004

#Unzipping

```
[ ]: 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).

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

#Data Augmentation

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

train_y = ImageDataGenerator(rescale=1./255,
                             zoom_range=0.2,
                             horizontal_flip=True)
test_y = ImageDataGenerator(rescale=1./255)
```

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

Found 4317 images belonging to 5 classes.

#Train

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

```
[ ]: 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'))
```

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

```
[ ]: 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]
```

```
[ ]: 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 [=====] - 35s 766ms/step - loss: 1.6836 -  
accuracy: 0.3702 - lr: 0.0010

Epoch 2/100

44/44 [=====] - 34s 773ms/step - loss: 1.1370 -  
accuracy: 0.5307 - lr: 0.0010

Epoch 3/100

44/44 [=====] - 35s 780ms/step - loss: 1.0635 -  
accuracy: 0.5833 - lr: 0.0010

Epoch 4/100

44/44 [=====] - 33s 743ms/step - loss: 0.9763 -  
accuracy: 0.6236 - lr: 0.0010

Epoch 5/100

44/44 [=====] - 31s 707ms/step - loss: 0.9400 -  
accuracy: 0.6298 - lr: 0.0010

Epoch 6/100

44/44 [=====] - 32s 705ms/step - loss: 0.8905 -  
accuracy: 0.6576 - lr: 0.0010

Epoch 7/100

44/44 [=====] - 32s 730ms/step - loss: 0.8324 -  
accuracy: 0.6759 - lr: 0.0010

Epoch 8/100

44/44 [=====] - 33s 752ms/step - loss: 0.8007 -

```

accuracy: 0.6864 - lr: 0.0010
Epoch 9/100
44/44 [=====] - 31s 691ms/step - loss: 0.7690 -
accuracy: 0.7026 - lr: 0.0010
Epoch 10/100
44/44 [=====] - 32s 721ms/step - loss: 0.7285 -
accuracy: 0.7285 - lr: 0.0010
Epoch 11/100
44/44 [=====] - 32s 722ms/step - loss: 0.6966 -
accuracy: 0.7338 - lr: 0.0010
Epoch 12/100
44/44 [=====] - 31s 706ms/step - loss: 0.6822 -
accuracy: 0.7392 - lr: 0.0010
Epoch 13/100
44/44 [=====] - 33s 740ms/step - loss: 0.6427 -
accuracy: 0.7547 - lr: 0.0010
Epoch 14/100
44/44 [=====] - 31s 689ms/step - loss: 0.6162 -
accuracy: 0.7707 - lr: 0.0010
Epoch 15/100
44/44 [=====] - 31s 688ms/step - loss: 0.5845 -
accuracy: 0.7776 - lr: 0.0010
Epoch 16/100
44/44 [=====] - 31s 708ms/step - loss: 0.5786 -
accuracy: 0.7830 - lr: 0.0010
Epoch 17/100
44/44 [=====] - 31s 694ms/step - loss: 0.5435 -
accuracy: 0.7962 - lr: 0.0010
Epoch 18/100
44/44 [=====] - 31s 696ms/step - loss: 0.4968 -
accuracy: 0.8140 - lr: 0.0010
Epoch 19/100
44/44 [=====] - 33s 739ms/step - loss: 0.4906 -
accuracy: 0.8209 - lr: 0.0010
Epoch 20/100
44/44 [=====] - 33s 738ms/step - loss: 0.4508 -
accuracy: 0.8341 - lr: 0.0010
Epoch 21/100
44/44 [=====] - 31s 699ms/step - loss: 0.4288 -
accuracy: 0.8388 - lr: 0.0010
Epoch 22/100
44/44 [=====] - 30s 679ms/step - loss: 0.4152 -
accuracy: 0.8453 - lr: 0.0010
Epoch 23/100
44/44 [=====] - 33s 749ms/step - loss: 0.4048 -
accuracy: 0.8499 - lr: 0.0010
Epoch 24/100
44/44 [=====] - 31s 708ms/step - loss: 0.3726 -

```

```

accuracy: 0.8682 - lr: 0.0010
Epoch 25/100
44/44 [=====] - 31s 693ms/step - loss: 0.3501 -
accuracy: 0.8768 - lr: 0.0010
Epoch 26/100
44/44 [=====] - 31s 709ms/step - loss: 0.3310 -
accuracy: 0.8749 - lr: 0.0010
Epoch 27/100
44/44 [=====] - 31s 697ms/step - loss: 0.3356 -
accuracy: 0.8749 - lr: 0.0010
Epoch 28/100
44/44 [=====] - 35s 793ms/step - loss: 0.2772 -
accuracy: 0.9048 - lr: 0.0010
Epoch 29/100
44/44 [=====] - 33s 734ms/step - loss: 0.2802 -
accuracy: 0.9036 - lr: 0.0010
Epoch 30/100
44/44 [=====] - 32s 717ms/step - loss: 0.2416 -
accuracy: 0.9159 - lr: 0.0010
Epoch 31/100
44/44 [=====] - 31s 704ms/step - loss: 0.2433 -
accuracy: 0.9152 - lr: 0.0010
Epoch 32/100
44/44 [=====] - 32s 711ms/step - loss: 0.2527 -
accuracy: 0.9071 - lr: 0.0010
Epoch 33/100
44/44 [=====] - 34s 758ms/step - loss: 0.2367 -
accuracy: 0.9143 - lr: 0.0010

```

```
[ ]: <keras.callbacks.History at 0x7f1361a4b410>
```

```
[ ]: model.save('flower_cnn.h5')
```

```
#Test
```

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

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

```
[ ]:
```



```
[ ]: X = image.img_to_array(daisy)
X
```

```
[ ]: array([[[144., 141., 152.],
            [197., 199., 186.],
            [196., 196., 194.],
            ...,
            [115., 115., 89.],
            [151., 151., 143.],
            [162., 162., 154.]],

           [[190., 187., 206.],
            [212., 214., 209.],
            [203., 200., 195.],
            ...,
            [130., 131., 99.],
            [170., 167., 158.],
            [181., 178., 169.]],

           [[203., 201., 215.],
            [214., 218., 217.],
            [198., 191., 181.],
            ...,
            [ 84., 86., 46.],
            [126., 122., 97.],
            [146., 142., 117.]],

           ...,

           [[102., 102., 32.],
            [ 63., 61., 10.],
            [ 53., 53., 15.],
            ...,
            [ 65., 64., 36.],
            [ 48., 47., 16.],
            [ 55., 54., 23.]],

           [[ 78., 77., 21.],
            [ 48., 46., 7.],
            [ 40., 38., 13.],
            ...,
            [ 49., 49., 13.],
            [ 39., 39., 5.],
            [ 52., 52., 18.]],

           [[ 44., 44., 8.],
            [ 25., 24., 3.]])
```

```

[ 21.,  22.,   8.],
...,
[ 50.,  52.,   3.],
[ 54.,  54.,  18.],
[ 57.,  57.,  21.]], dtype=float32)

```

```

[ ]: X= np.expand_dims(X,axis= 0)
X

```

```

[ ]: array([[[[144., 141., 152.],
              [197., 199., 186.],
              [196., 196., 194.],
              ...,
              [115., 115.,  89.],
              [151., 151., 143.],
              [162., 162., 154.]],

            [[190., 187., 206.],
              [212., 214., 209.],
              [203., 200., 195.],
              ...,
              [130., 131.,  99.],
              [170., 167., 158.],
              [181., 178., 169.]],

            [[203., 201., 215.],
              [214., 218., 217.],
              [198., 191., 181.],
              ...,
              [ 84.,  86.,  46.],
              [126., 122.,  97.],
              [146., 142., 117.]],

            ...,

            [[102., 102.,  32.],
              [ 63.,  61.,  10.],
              [ 53.,  53.,  15.],
              ...,
              [ 65.,  64.,  36.],
              [ 48.,  47.,  16.],
              [ 55.,  54.,  23.]],

            [[ 78.,  77.,  21.],
              [ 48.,  46.,   7.],
              [ 40.,  38.,  13.],
              ...,

```

```

[ 49.,  49.,  13.],
[ 39.,  39.,   5.],
[ 52.,  52.,  18.]],

[[ 44.,  44.,   8.],
 [ 25.,  24.,   3.],
 [ 21.,  22.,   8.],
 ...,
 [ 50.,  52.,   3.],
 [ 54.,  54.,  18.],
 [ 57.,  57.,  21.]]], dtype=float32)

```

```

[ ]: value_xyz = list(xtrain.class_indices.keys())
      value_xyz

```

```

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

```

```

[ ]: value_xyz[np.argmax(model.predict(X))]

```

```

[ ]: 'daisy'

```