

Assignment_3

October 10, 2022

1 Download the Dataset

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

Mounted at /content/drive

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

2 Image Augmentation

```
[9]: # Importing Library  
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
[10]: # expanding training and testing variable  
train_data=ImageDataGenerator(rescale=1./255,zoom_range=0.  
    ↳2,horizontal_flip=True)  
test_data=ImageDataGenerator(rescale=1./255)
```

```
[11]: #Data augmentation on testing data  
xtrain = train_data.flow_from_directory('/content/  
    ↳flowers',target_size=(76,76),class_mode='categorical',batch_size=200)
```

Found 4317 images belonging to 5 classes.

```
[81]: xtest = test_data.flow_from_directory('/content/  
    ↳flowers',target_size=(76,76),class_mode='categorical',batch_size=200)
```

Found 4317 images belonging to 5 classes.

3 Create Model

```
[82]: #Importing Libraries
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D, MaxPooling2D, Flatten, Dense

model = Sequential()
```

4 Add Layers (Convolution,MaxPooling,Flatten,Dense-(Hidden Layers),Output)

```
[83]: #Building a CNN block

model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(76,76,3))) #↳
    ↳Convolution layer
model.add(MaxPooling2D(pool_size=(2, 2))) # Max pooling layer
model.add(Flatten()) # Flatten layer
model.add(Dense(4317,activation='relu')) # Hidden layer 1
model.add(Dense(2500,activation='relu')) # Hidden layer 2
model.add(Dense(5,activation='softmax')) # Output layer
```

5 Compile The Model

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

6 Fit The Model

```
[88]: # Training model
model.fit_generator(xtrain,steps_per_epoch=len(xtrain),
                    epochs=15,
                    validation_data=xtest,
                    validation_steps=len(xtest))
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: UserWarning:
`Model.fit_generator` is deprecated and will be removed in a future version.
Please use `Model.fit`, which supports generators.
"""
```

```
Epoch 1/15
22/22 [=====] - 24s 1s/step - loss: 0.7827 - accuracy:
```

```

0.7007 - val_loss: 0.8009 - val_accuracy: 0.7007
Epoch 2/15
22/22 [=====] - 23s 1s/step - loss: 0.7489 - accuracy:
0.7116 - val_loss: 0.7660 - val_accuracy: 0.7121
Epoch 3/15
22/22 [=====] - 23s 1s/step - loss: 0.7323 - accuracy:
0.7283 - val_loss: 0.6890 - val_accuracy: 0.7410
Epoch 4/15
22/22 [=====] - 23s 1s/step - loss: 0.6910 - accuracy:
0.7415 - val_loss: 0.5964 - val_accuracy: 0.7825
Epoch 5/15
22/22 [=====] - 23s 1s/step - loss: 0.6489 - accuracy:
0.7568 - val_loss: 0.6406 - val_accuracy: 0.7619
Epoch 6/15
22/22 [=====] - 23s 1s/step - loss: 0.6533 - accuracy:
0.7496 - val_loss: 0.6916 - val_accuracy: 0.7294
Epoch 7/15
22/22 [=====] - 27s 1s/step - loss: 0.6184 - accuracy:
0.7637 - val_loss: 0.6493 - val_accuracy: 0.7545
Epoch 8/15
22/22 [=====] - 24s 1s/step - loss: 0.5652 - accuracy:
0.7871 - val_loss: 0.5722 - val_accuracy: 0.7874
Epoch 9/15
22/22 [=====] - 23s 1s/step - loss: 0.5405 - accuracy:
0.7929 - val_loss: 0.5371 - val_accuracy: 0.8019
Epoch 10/15
22/22 [=====] - 23s 1s/step - loss: 0.5119 - accuracy:
0.8112 - val_loss: 0.4794 - val_accuracy: 0.8251
Epoch 11/15
22/22 [=====] - 23s 1s/step - loss: 0.4825 - accuracy:
0.8256 - val_loss: 0.4344 - val_accuracy: 0.8362
Epoch 12/15
22/22 [=====] - 23s 1s/step - loss: 0.4712 - accuracy:
0.8270 - val_loss: 0.4508 - val_accuracy: 0.8344
Epoch 13/15
22/22 [=====] - 23s 1s/step - loss: 0.4564 - accuracy:
0.8372 - val_loss: 0.5350 - val_accuracy: 0.8089
Epoch 14/15
22/22 [=====] - 23s 1s/step - loss: 0.4587 - accuracy:
0.8328 - val_loss: 0.5626 - val_accuracy: 0.8082
Epoch 15/15
22/22 [=====] - 23s 1s/step - loss: 0.4247 - accuracy:
0.8395 - val_loss: 0.3706 - val_accuracy: 0.8564

```

[88]: <keras.callbacks.History at 0x7f365a048090>

7 Save The Model

```
[89]: model.save('flowers.h5')
```

8 Test The Model

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

```
[93]: # Testing 1(daisy)

img = image.load_img('/content/flowers/daisy/100080576_f52e8ee070_n.
    ↳jpg',target_size=(76,76))
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
prediction = np.argmax(model.predict(x))
op = ['daisy','dandelion','rose','sunflower','tulip']
op[prediction]
```

```
[93]: 'daisy'
```

```
[95]: #Testing 2(dandelion)

img = image.load_img('/content/flowers/dandelion/1195255751_d58b3d3076.
    ↳jpg',target_size=(76,76))
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
prediction = np.argmax(model.predict(x))
op = ['daisy','dandelion','rose','sunflower','tulip']
op[prediction]
```

```
[95]: 'dandelion'
```

```
[96]: #Testing 3()

img = image.load_img('/content/flowers/rose/7456887736_54e4ebac03_n.
    ↳jpg',target_size=(76,76))
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
prediction = np.argmax(model.predict(x))
op = ['daisy','dandelion','rose','sunflower','tulip']
op[prediction]
```

```
[96]: 'rose'
```

[98]: *#Testing 4(Sunflower)*

```
img = image.load_img('/content/flowers/sunflower/7012364067_5ffc7654c9_m.  
→jpg',target_size=(76,76))  
x = image.img_to_array(img)  
x = np.expand_dims(x,axis=0)  
prediction = np.argmax(model.predict(x))  
op = ['daisy','dandelion','rose','sunflower','tulip']  
op[prediction]
```

[98]: 'sunflower'

[99]: *#Testing 5(tulip)*

```
img = image.load_img('/content/flowers/tulip/8892851067_79242a7362_n.  
→jpg',target_size=(76,76))  
x = image.img_to_array(img)  
x = np.expand_dims(x,axis=0)  
prediction = np.argmax(model.predict(x))  
op = ['daisy','dandelion','rose','sunflower','tulip']  
op[prediction]
```

[99]: 'tulip'