

ASSIGNMENT 3, NAME: SANJAY R, ROLL NUMBER: 110819104030

```
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)



```
ls
```

```
drive/ sample_data/
```

```
cd //content/drive/MyDrive/IBM
```

```
/content/drive/MyDrive/IBM
```

```
pwd
```

```
'/content/drive/MyDrive/IBM'
```

```
!unzip images.zip
```

```
unzip: cannot find or open images.zip, images.zip.zip or images.zip.ZIP.
```

Image Augmentation

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
train_datagen = ImageDataGenerator(rescale = 1./255, zoom_range= 0.3, horizontal_flip=True, vertical_flip=True)
```

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

```
x_train = train_datagen.flow_from_directory(r"/content/drive/MyDrive/IBM",target_size= (64,64),class_mode= "categorical",batch_size=
```

```
Found 5 images belonging to 1 classes.
```

```
x_test = test_datagen.flow_from_directory(r"/content/drive/MyDrive/IBM",target_size= (64,64),class_mode= "categorical",batch_size= 24
```

```
Found 5 images belonging to 1 classes.
```

```
x_train.class_indices
```

```
{'images': 0}
```

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

```
from tensorflow.keras.layers import Dense,Convolution2D,MaxPooling2D,Flatten
```

```
model = Sequential()
```

```
model.add(Convolution2D(32,(3,3),activation="relu",strides=(1, 1),input_shape =(64,64,3)))
```

```
model.add(MaxPooling2D(strides=(1, 1)))
```

```
model.add(Flatten())
```

```
model.summary()
```

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 62, 62, 32)	896

```
max_pooling2d (MaxPooling2D (None, 61, 61, 32)      0
)
```

```
flatten (Flatten)          (None, 119072)          0
```

```
=====
Total params: 896
Trainable params: 896
Non-trainable params: 0
```

```
model.add(Dense(300,activation="relu"))
model.add(Dense(300,activation="relu"))
```

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

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

```
len(x_train)
```

```
1
```

```
model.fit(x_train,epochs = 10,steps_per_epoch=len(x_train),validation_data=x_test,validation_steps=len(x_test))
```

```
Epoch 1/10
```

```
1/1 [=====] - 1s 1s/step - loss: 8.0867 - accuracy: 0.0000e+00 - val_loss: 39.2395 - val_accuracy: 0.0
```

```
Epoch 2/10
```

```
1/1 [=====] - 0s 327ms/step - loss: 35.0594 - accuracy: 0.0000e+00 - val_loss: 50.7268 - val_accuracy:
```

```
Epoch 3/10
```

```
1/1 [=====] - 0s 289ms/step - loss: 49.9106 - accuracy: 0.0000e+00 - val_loss: 46.2973 - val_accuracy:
```

```
Epoch 4/10
```

```
1/1 [=====] - 0s 272ms/step - loss: 44.1595 - accuracy: 0.0000e+00 - val_loss: 52.9596 - val_accuracy:
```

```
Epoch 5/10
```

```
1/1 [=====] - 0s 310ms/step - loss: 48.1273 - accuracy: 0.0000e+00 - val_loss: 66.4492 - val_accuracy:
```

```
Epoch 6/10
```

```
1/1 [=====] - 0s 280ms/step - loss: 61.1015 - accuracy: 0.0000e+00 - val_loss: 78.9051 - val_accuracy:
Epoch 7/10
1/1 [=====] - 0s 283ms/step - loss: 74.1666 - accuracy: 0.0000e+00 - val_loss: 71.6539 - val_accuracy:
Epoch 8/10
1/1 [=====] - 0s 282ms/step - loss: 69.7144 - accuracy: 0.0000e+00 - val_loss: 35.9527 - val_accuracy:
Epoch 9/10
1/1 [=====] - 0s 284ms/step - loss: 35.4881 - accuracy: 0.6000 - val_loss: 47.1248 - val_accuracy: 1.0
Epoch 10/10
1/1 [=====] - 0s 289ms/step - loss: 44.4445 - accuracy: 1.0000 - val_loss: 54.2549 - val_accuracy: 0.0
<keras.callbacks.History at 0x7f0333a99690>
```

```
model.save("flower.h5")
```

```
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
```

```
model = load_model("flower.h5")
```

```
img = image.load_img(r"/content/drive/MyDrive/IBM/images/flower4.jpeg")
```

```
img
```



```
img = image.load_img(r"/content/drive/MyDrive/IBM/images/flower1.jpeg",target_size=(700,800))
```

```
img
```



```
x = image.img_to_array(img)
```



x

```
array([[ 79.,  21.,  69.],
       [ 79.,  21.,  69.],
       [ 79.,  21.,  69.],
       ...,
       [ 33.,   8.,  29.],
       [ 33.,   8.,  29.],
       [ 33.,   8.,  29.]],

      [[ 79.,  21.,  69.],
       [ 79.,  21.,  69.],
       [ 79.,  21.,  69.],
       ...,
       [ 33.,   8.,  29.],
       [ 33.,   8.,  29.],
       [ 33.,   8.,  29.]],

      [[ 79.,  21.,  69.],
       [ 79.,  21.,  69.],
       [ 79.,  21.,  69.],
       ...,
       [ 33.,   8.,  29.],
       [ 33.,   8.,  29.],
       [ 33.,   8.,  29.]],

      ...,

      [[ 22.,   9.,  26.],
       [ 22.,   9.,  26.],
       [ 22.,   9.,  26.]])
```

```

...
[129., 121., 132.],
[129., 121., 132.],
[129., 121., 132.]],

[[ 22.,  9., 26.],
 [ 22.,  9., 26.],
 [ 22.,  9., 26.],
...
[129., 121., 132.],
[129., 121., 132.],
[129., 121., 132.]],

[[ 22.,  9., 26.],
 [ 22.,  9., 26.],
 [ 22.,  9., 26.],
...
[129., 121., 132.],
[129., 121., 132.],
[129., 121., 132.]]], dtype=float32)

```

```
x = np.expand_dims(x,axis = 0)
```

```
x
```

```

array([[[[ 79., 21., 69.],
          [ 79., 21., 69.],
          [ 79., 21., 69.],
          ...
          [ 33.,  8., 29.],
          [ 33.,  8., 29.],
          [ 33.,  8., 29.]],

        [[ 79., 21., 69.],
          [ 79., 21., 69.],
          [ 79., 21., 69.],
          ...
          [ 33.,  8., 29.],
          [ 33.,  8., 29.],
          [ 33.,  8., 29.]]],

```

```

[ 33.,   8.,  29.]],

[[ 79.,  21.,  69.],
 [ 79.,  21.,  69.],
 [ 79.,  21.,  69.],
 ...,
 [ 33.,   8.,  29.],
 [ 33.,   8.,  29.],
 [ 33.,   8.,  29.]],

...,

[[ 22.,   9.,  26.],
 [ 22.,   9.,  26.],
 [ 22.,   9.,  26.],
 ...,
 [129., 121., 132.],
 [129., 121., 132.],
 [129., 121., 132.]],

[[ 22.,   9.,  26.],
 [ 22.,   9.,  26.],
 [ 22.,   9.,  26.],
 ...,
 [129., 121., 132.],
 [129., 121., 132.],
 [129., 121., 132.]],

[[ 22.,   9.,  26.],
 [ 22.,   9.,  26.],
 [ 22.,   9.,  26.],
 ...,
 [129., 121., 132.],
 [129., 121., 132.],
 [129., 121., 132.] ]], dtype=float32)

```

```
pred = model.predict(x)
```

```
pred
```



```
x_test.class_indices
```

```
{'images': 0}
```

```
index = ["", "images"]
```

```
img = image.load_img(r"/content/drive/MyDrive/IBM/images/flower3.jpeg", target_size=(64, 64))
```

```
img
```



```
img = image.load_img(r"/content/drive/MyDrive/IBM/images/flower5.jpeg")
```

```
img
```



[Colab paid products](#) - [Cancel contracts here](#)

✓ 0s completed at 3:53 PM

