ASSIGNMENT 3, NAME: MITHOON N S, ROLL NUMBER: 110819104301

```
from google.colab import drive
drive.mount('/content/drive')
```

```
    Mounted at /content/drive
```

```
ls
```

```
    drive/  sample_data/
```

```
cd /content/drive/MyDrive/tanmoy_IBM_nalaiyathiran/flowers
```

```
    /content/drive/MyDrive/tanmoy_IBM_nalaiyathiran/flowers
```

```
pwd
```

```
    '/content/drive/MyDrive/tanmoy_IBM_nalaiyathiran/flowers'
```

```
!unzip flowers.zip
```

```
    Archive:  flowers.zip
      inflating: flowers/Achillea.jpg
      inflating: flowers/African-Daisy.jpg
      inflating: flowers/American-Lotus.jpg
      inflating: flowers/filigran.jpg
      inflating: flowers/rose.jpg
```

Image Augmentation

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

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

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

```
x_train = train_datagen.flow_from_directory(r"/content/drive/MyDrive/tanmoy_IBM_nalaiyathi
```

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

```
x_test = test_datagen.flow_from_directory(r"/content/drive/MyDrive/tanmoy_IBM_nalaiyathira
```

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

```
x_train.class_indices
```

```
{'flowers': 0}
```

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

```python
model = Sequential()
```

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

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

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

```python
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
_____
```

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

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

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

```python
len(x_train)
```

```
1
```

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

```
Epoch 1/10
1/1 [==============================] - 2s 2s/step - loss: 8.0674 - accuracy: 0.0000e
Epoch 2/10
1/1 [==============================] - 1s 820ms/step - loss: 31.5680 - accuracy: 1.0
Epoch 3/10
```

```
1/1 [==============================] - 1s 837ms/step - loss: 38.8966 - accuracy: 0.0
Epoch 4/10
1/1 [==============================] - 1s 865ms/step - loss: 52.5986 - accuracy: 0.0
Epoch 5/10
1/1 [==============================] - 1s 825ms/step - loss: 69.2919 - accuracy: 0.0
Epoch 6/10
1/1 [==============================] - 1s 918ms/step - loss: 91.7591 - accuracy: 0.0
Epoch 7/10
1/1 [==============================] - 1s 827ms/step - loss: 123.6914 - accuracy: 0.
Epoch 8/10
1/1 [==============================] - 1s 834ms/step - loss: 135.3233 - accuracy: 0.
Epoch 9/10
1/1 [==============================] - 1s 813ms/step - loss: 179.7045 - accuracy: 0.
Epoch 10/10
1/1 [==============================] - 1s 843ms/step - loss: 207.5992 - accuracy: 0.
<keras.callbacks.History at 0x7f58029383d0>
```

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

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

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

```python
img = image.load_img(r"/content/drive/MyDrive/tanmoy_IBM_nalaiyathiran/flowers/African-Dai
```

```python
img
```

```python
img = image.load_img(r"/content/drive/MyDrive/tanmoy_IBM_nalaiyathiran/flowers/African-Dai
```



```python
img
```



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



```python
x
```

```
array([[[108., 163.,  46.],
        [ 93., 141.,  23.],
        [ 35.,  69.,   9.],
        ...,
        [130., 188., 138.],
        [ 56.,  93.,  39.],
        [ 52.,  90.,  41.]],

       [[ 23.,  41.,  15.],
        [ 15.,  41.,   0.],
        [ 30.,  63.,  10.],
        ...,
        [ 57., 113.,  52.],
        [101., 149.,  87.],
        [ 60.,  98.,  47.]],

       [[  2.,   0.,   3.],
        [  2.,   0.,   1.],
        [ 68., 103.,  23.],
        ...,
        [ 43.,  76.,  29.],
        [ 53.,  92.,  39.],
        [ 61., 100.,  45.]],

       ...,

       [[ 23.,  31.,  18.],
        [253., 189., 162.],
        [255., 172., 138.],
        ...,
        [ 38.,  65.,  22.],
        [ 58.,  97.,  53.],
        [ 15.,  52.,   0.]],

       [[ 47.,  45.,  24.],
        [124.,  59.,  41.],
        [254., 144., 129.],
```

```
              ...,
              [ 35.,  66.,  22.],
              [ 23.,  59.,  13.],
              [ 32.,  68.,  22.]],

            [[255., 168., 140.],
             [209.,  94.,  65.],
             [248., 166., 152.],
              ...,
             [ 19.,  44.,   5.],
             [ 14.,  39.,   0.],
             [ 20.,  48.,  10.]]], dtype=float32)
```

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

```python
x
```

```
     array([[[[108., 163.,  46.],
              [ 93., 141.,  23.],
              [ 35.,  69.,   9.],
              ...,
              [130., 188., 138.],
              [ 56.,  93.,  39.],
              [ 52.,  90.,  41.]],

             [[ 23.,  41.,  15.],
              [ 15.,  41.,   0.],
              [ 30.,  63.,  10.],
              ...,
              [ 57., 113.,  52.],
              [101., 149.,  87.],
              [ 60.,  98.,  47.]],

             [[  2.,   0.,   3.],
              [  2.,   0.,   1.],
              [ 68., 103.,  23.],
              ...,
              [ 43.,  76.,  29.],
              [ 53.,  92.,  39.],
              [ 61., 100.,  45.]],

              ...,

             [[ 23.,  31.,  18.],
              [253., 189., 162.],
              [255., 172., 138.],
              ...,
              [ 38.,  65.,  22.],
              [ 58.,  97.,  53.],
              [ 15.,  52.,   0.]],

             [[ 47.,  45.,  24.],
              [124.,  59.,  41.],
              [254., 144., 129.],
              ...,
              [ 35.,  66.,  22.],
              [ 23.,  59.,  13.],
              [ 32.,  68.,  22.]],
```

```
          [[255., 168., 140.],
           [209.,  94.,  65.],
           [248., 166., 152.],
           ...,
           [ 19.,  44.,   5.],
           [ 14.,  39.,   0.],
           [ 20.,  48.,  10.]]]], dtype=float32)
```

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

```
    1/1 [==============================] - 0s 126ms/step
```

```
pred
```

```
    array([[0., 0., 0., 0., 1.]], dtype=float32)
```

```
x_test.class_indices
```

```
    {'flowers': 0}
```

```
index = ["","flowers"]
```

```
img = image.load_img(r"/content/drive/MyDrive/tanmoy_IBM_nalaiyathiran/flowers/filigran.j
```

```
img
```



```
img = image.load_img(r"/content/drive/MyDrive/tanmoy_IBM_nalaiyathiran/flowers/Achillea.j
```

```
img
```

✓ 4s   completed at 11:04 AM   ● ✕