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

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

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.
```

```
ls
```

```
drive/  sample_data/
```

```
cd /content/drive/MyDrive/IBM_nalaiyathiran/images
```

```
/content/drive/MyDrive/IBM_nalaiyathiran/images
```

```
pwd
```

```
'/content/drive/MyDrive/IBM_nalaiyathiran/images'
```

```
!unzip flowers.zip
```

```
Archive:  flowers.zip
  inflating: flowers/Achillea.jpg
  inflating: flowers/African-Daisy.jpg
  inflating: flowers/American-Lotus.jpg
replace flowers/filigran.jpg? [y]es, [n]o, [A]ll, [N]one, [r]ename: n
                              s, [n]o, [A]ll, [N]one, [r]ename: r
```

Saved successfully!

Image Augmentation

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

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

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

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

```
Found 7 images belonging to 1 classes.
```

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

```
Found 7 images belonging to 1 classes.
```

```
x_train.class_indices
```

```
{'flowers': 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_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_1 (Conv2D)           (None, 62, 62, 32)        896

 max_pooling2d_2 (MaxPooling  (None, 61, 61, 32)       0
 2D)

 flatten_1 (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,validati
```

```
Epoch 1/10
1/1 [==============================] - 2s 2s/step - loss: 8.0560 - accuracy: 0.0000e
```

```
Epoch 2/10
1/1 [==============================] - 1s 828ms/step - loss: 15.1767 - accuracy: 0.0
Epoch 3/10
1/1 [==============================] - 1s 848ms/step - loss: 17.1117 - accuracy: 1.0
Epoch 4/10
1/1 [==============================] - 1s 812ms/step - loss: 24.6044 - accuracy: 0.0
Epoch 5/10
1/1 [==============================] - 1s 828ms/step - loss: 22.8853 - accuracy: 0.0
Epoch 6/10
1/1 [==============================] - 1s 855ms/step - loss: 24.2076 - accuracy: 0.0
Epoch 7/10
1/1 [==============================] - 1s 884ms/step - loss: 25.9766 - accuracy: 0.0
Epoch 8/10
1/1 [==============================] - 1s 845ms/step - loss: 22.2866 - accuracy: 0.0
Epoch 9/10
1/1 [==============================] - 1s 1s/step - loss: 27.6694 - accuracy: 1.0000
Epoch 10/10
1/1 [==============================] - 1s 919ms/step - loss: 59.5761 - accuracy: 0.0
<keras.callbacks.History at 0x7ff496f9a610>
```

```
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_nalaiyathiran/images/flowers/Achillea.jp
```

Saved successfully!

```
img = image.load_img(r"/content/drive/MyDrive/IBM_nalaiyathiran/images/flowers/Achillea.jp
```

```
img
```

Saved successfully!

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

```
x
```

```
array([[[ 67.,  85.,  37.],
        [ 53.,  69.,  22.],
        [ 58.,  74.,  25.],
        ...,
        [174., 168.,  84.],
        [206., 193., 149.],
        [ 64.,  87.,  31.]],

       [[ 74.,  92.,  42.],
        [ 61.,  77.,  30.],
        [ 50.,  66.,  17.],
        ...,
        [145., 140.,  84.],
        [115., 116.,  48.],
        [ 72.,  96.,  34.]],
```

```
            [[ 93., 117.,  59.],
             [ 66.,  82.,  35.],
             [ 47.,  62.,   5.],
             ...,
             [ 96., 120.,  58.],
             [ 82., 106.,  48.],
             [ 91., 117.,  52.]],

            ...,

            [[ 35.,  39.,  12.],
             [ 68.,  74.,  36.],
             [ 55.,  61.,  27.],
             ...,
             [232., 217., 174.],
             [ 68.,  73.,  51.],
             [ 84.,  69.,  64.]],

            [[ 54.,  60.,  22.],
             [ 65.,  70.,  38.],
             [ 51.,  51.,  25.],
             ...,
             [193., 182., 137.],
             [149., 139., 114.],
             [ 82.,  66.,  67.]],

            [[ 57.,  64.,  22.],
             [ 48.,  50.,  28.],
             [ 34.,  32.,  19.],
             ...,
             [154., 130., 102.],
             [134., 100.,  90.],
             [ 89.,  71.,  67.]]], dtype=float32)
```

Saved successfully!

x

```
array([[[[ 67.,  85.,  37.],
         [ 53.,  69.,  22.],
         [ 58.,  74.,  25.],
         ...,
         [174., 168.,  84.],
         [206., 193., 149.],
         [ 64.,  87.,  31.]],

        [[ 74.,  92.,  42.],
         [ 61.,  77.,  30.],
         [ 50.,  66.,  17.],
         ...,
         [145., 140.,  84.],
         [115., 116.,  48.],
         [ 72.,  96.,  34.]],

        [[ 93., 117.,  59.],
         [ 66.,  82.,  35.],
         [ 47.,  62.,   5.],
```

```
                ...,
                [ 96., 120.,  58.],
                [ 82., 106.,  48.],
                [ 91., 117.,  52.]],

               ...,

               [[ 35.,  39.,  12.],
                [ 68.,  74.,  36.],
                [ 55.,  61.,  27.],
                ...,
                [232., 217., 174.],
                [ 68.,  73.,  51.],
                [ 84.,  69.,  64.]],

               [[ 54.,  60.,  22.],
                [ 65.,  70.,  38.],
                [ 51.,  51.,  25.],
                ...,
                [193., 182., 137.],
                [149., 139., 114.],
                [ 82.,  66.,  67.]],

               [[ 57.,  64.,  22.],
                [ 48.,  50.,  28.],
                [ 34.,  32.,  19.],
                ...,
                [154., 130., 102.],
                [134., 100.,  90.],
                [ 89.,  71.,  67.]]]], dtype=float32)
```

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

```
                                        =====] - 0s 100ms/step
```

Saved successfully!

```
pred
```

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

```
x_test.class_indices
```

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

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

```
img = image.load_img(r"/content/drive/MyDrive/IBM_nalaiyathiran/images/flowers/rose.jpg",t
```

```
img
```

```
img = image.load_img(r"/content/drive/MyDrive/IBM_nalaiyathiran/images/flowers/rose.jpg")
```

```
img
```

Saved successfully!

Colab paid products - Cancel contracts here

0s completed at 1:03 PM