

ASSIGNMENT 3, NAME: JAYA KUMAR S, ROLL NUMBER: 110819104015

```
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/IMAGE
```

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

```
pwd
```

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

```
!unzip FLOWERS.zip
```

```
Archive:  FLOWERS.zip
  inflating: IMAGE/F1.jpeg
  inflating: IMAGE/F2.jpeg
  inflating: IMAGE/F3.jpeg
  inflating: IMAGE/F4.jpeg
  inflating: IMAGE/F5.jpeg
```

Saving...



```
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/IMAGE", target_size= (64,64), class_mode= "categorical", batch_size= 10)

Found 5 images belonging to 1 classes.

x_test = test_datagen.flow_from_directory(r"/content/drive/MyDrive/IBM/IMAGE", target_size= (64,64), class_mode= "categorical", batch_size= 10)

Found 5 images belonging to 1 classes.

x_train.class_indices

{'IMAGE': 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.summary()
```

Saving...



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

Saving...



Epoch 3/10

1/1 [=====] - 1s 852ms/step - loss: 36.5931 - accuracy: 0.0000e+00 - val_loss: 64.1929 - val_accuracy:

```

Epoch 4/10
1/1 [=====] - 1s 864ms/step - loss: 55.9806 - accuracy: 0.0000e+00 - val_loss: 45.6361 - val_accuracy:
Epoch 5/10
1/1 [=====] - 1s 1s/step - loss: 44.4671 - accuracy: 0.0000e+00 - val_loss: 47.2510 - val_accuracy: 0.
Epoch 6/10
1/1 [=====] - 1s 871ms/step - loss: 45.2314 - accuracy: 0.0000e+00 - val_loss: 61.4744 - val_accuracy:
Epoch 7/10
1/1 [=====] - 1s 989ms/step - loss: 57.3274 - accuracy: 0.0000e+00 - val_loss: 58.2878 - val_accuracy:
Epoch 8/10
1/1 [=====] - 1s 888ms/step - loss: 57.3067 - accuracy: 0.0000e+00 - val_loss: 58.7243 - val_accuracy:
Epoch 9/10
1/1 [=====] - 1s 1s/step - loss: 56.6390 - accuracy: 0.0000e+00 - val_loss: 96.0627 - val_accuracy: 0.
Epoch 10/10
1/1 [=====] - 1s 835ms/step - loss: 87.5147 - accuracy: 0.0000e+00 - val_loss: 85.2706 - val_accuracy:
<keras.callbacks.History at 0x7fac0332bed0>

```

```
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/IMAGE/F1.jpeg")
```

```
img
```

Saving...





```
img = image.load_img(r"/content/drive/MyDrive/IBM/IMAGE/F1.jpeg",target_size=(64,64))
```



```
img
```



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

```
x
```

```
array([[ 95.,  72.,  58.],
       [127.,  93.,  65.],
       [144., 106.,  67.],
       ...,
       [ 50.,  72.,  23.],
       [ 49.,  71.,  25.],
       [ 44.,  70.,  22.]],

      [[110.,  86.,  74.],
       [142., 108.,  81.],
       [151., 113.,  76.],
       ...,
       [ 50.,  72.,  23.],
       [ 49.,  71.,  25.],
       [ 44.,  70.,  23.]],
```

Saving...



```
       [155., 115.,  84.],
       [157., 119.,  83.],
       ...,
```

```

[ 50., 72., 23.],
[ 49., 71., 25.],
[ 44., 70., 25.]],

...,

[[ 33., 32., 38.],
 [ 33., 32., 37.],
 [ 38., 38., 40.],
 ...,
 [ 42., 58., 13.],
 [ 49., 61., 25.],
 [ 28., 40., 20.]],

[[ 34., 35., 40.],
 [ 36., 34., 39.],
 [ 36., 37., 39.],
 ...,
 [ 68., 69., 27.],
 [ 68., 69., 25.],
 [ 38., 47., 30.]],

[[ 35., 36., 41.],
 [ 37., 35., 40.],
 [ 39., 40., 42.],
 ...,
 [ 90., 91., 47.],
 [114., 117., 60.],
 [ 82., 80., 57.] ]], dtype=float32)

```

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

```
x
```

Saving...



```

[ 44., 70., 25.],
...,
[ 50., 72., 23.],

```

```
[ 49.,  71.,  25.],
[ 44.,  70.,  22.]],
```

```
[[110.,  86.,  74.],
 [142., 108.,  81.],
 [151., 113.,  76.],
 ...,
 [ 50.,  72.,  23.],
 [ 49.,  71.,  25.],
 [ 44.,  70.,  23.]],
```

```
[[121.,  97.,  87.],
 [153., 119.,  94.],
 [157., 119.,  83.],
 ...,
 [ 50.,  72.,  23.],
 [ 49.,  71.,  25.],
 [ 44.,  70.,  25.]],
```

```
...,
```

```
[[ 33.,  32.,  38.],
 [ 33.,  32.,  37.],
 [ 38.,  38.,  40.],
 ...,
 [ 42.,  58.,  13.],
 [ 49.,  61.,  25.],
 [ 28.,  40.,  20.]],
```

```
[[ 34.,  35.,  40.],
 [ 36.,  34.,  39.],
 [ 36.,  37.,  39.],
 ...,
 [ 68.,  69.,  27.],
 [ 68.,  69.,  25.],
 [ 38.,  47.,  30.]],
```

```
[ 37.,  35.,  40.],
 [ 39.,  40.,  42.],
 ...,
```

Saving...



```
[ 90.,  91.,  47.],
 [114., 117.,  60.],
 [ 82.,  80.,  57.] ]], dtype=float32)
```

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

```
1/1 [=====] - 0s 165ms/step
```

```
pred
```

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

```
x_test.class_indices
```

```
{'IMAGE': 0}
```

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

```
img = image.load_img(r"/content/drive/MyDrive/IBM/IMAGE/F3.jpeg", target_size=(64,64))
```

```
img
```



Saving...



ve/MyDrive/IBM/IMAGE/F3.jpeg")

```
img
```




Saving...



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

✓ 0s completed at 11:13 AM

