

## LOAD DATASET

```
!unzip '/content/Flowers-Dataset.zip'
```

## IMPORT MODULES

```
from keras.preprocessing.image import ImageDataGenerator, image
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
import tensorflow as tf
import numpy as np
import pathlib
import PIL
```

## IMAGE AUGMENTATION

```
data_dir="/content/flowers"
```

```
flowers_train_datagen =
ImageDataGenerator(rescale=1./255, zoom_range=0.2, horizontal_flip=True)
xtrain =
flowers_train_datagen.flow_from_directory(data_dir, target_size=(64, 64)
, class_mode='categorical', batch_size=100)
```

Found 4317 images belonging to 5 classes.

## CREATE MODEL

```
model = Sequential()
```

## ADD LAYERS

```
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(64, 64, 3)))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(300, activation='relu'))
model.add(Dense(150, activation='relu'))
model.add(Dense(5, activation='softmax'))
```

## COMPILE THE MODEL

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

## FIT THE MODEL

```
model.fit_generator(xtrain,steps_per_epoch=len(xtrain),epochs=30)
```

```
Epoch 1/30
44/44 [=====] - 30s 692ms/step - loss: 0.0630
- acc: 0.9766
Epoch 2/30
44/44 [=====] - 34s 767ms/step - loss: 0.0893
- acc: 0.9705
Epoch 3/30
44/44 [=====] - 30s 687ms/step - loss: 0.0572
- acc: 0.9818
Epoch 4/30
44/44 [=====] - 29s 660ms/step - loss: 0.0681
- acc: 0.9784
Epoch 5/30
44/44 [=====] - 28s 635ms/step - loss: 0.0709
- acc: 0.9782
Epoch 6/30
44/44 [=====] - 28s 636ms/step - loss: 0.0598
- acc: 0.9839
Epoch 7/30
44/44 [=====] - 28s 629ms/step - loss: 0.0873
- acc: 0.9714
Epoch 8/30
44/44 [=====] - 31s 715ms/step - loss: 0.0957
- acc: 0.9660
Epoch 9/30
44/44 [=====] - 30s 690ms/step - loss: 0.0751
- acc: 0.9800
Epoch 10/30
44/44 [=====] - 28s 641ms/step - loss: 0.0534
- acc: 0.9829
Epoch 11/30
44/44 [=====] - 28s 633ms/step - loss: 0.0716
- acc: 0.9764
Epoch 12/30
44/44 [=====] - 28s 642ms/step - loss: 0.0612
- acc: 0.9804
Epoch 13/30
44/44 [=====] - 28s 636ms/step - loss: 0.0654
- acc: 0.9795
Epoch 14/30
44/44 [=====] - 28s 640ms/step - loss: 0.0659
- acc: 0.9794
Epoch 15/30
44/44 [=====] - 28s 632ms/step - loss: 0.0515
- acc: 0.9854
Epoch 16/30
44/44 [=====] - 28s 632ms/step - loss: 0.0624
```

```
- acc: 0.9811
Epoch 17/30
44/44 [=====] - 28s 640ms/step - loss: 0.0613
- acc: 0.9814
Epoch 18/30
44/44 [=====] - 28s 642ms/step - loss: 0.0581
- acc: 0.9827
Epoch 19/30
44/44 [=====] - 28s 626ms/step - loss: 0.0624
- acc: 0.9804
Epoch 20/30
44/44 [=====] - 28s 628ms/step - loss: 0.0397
- acc: 0.9882
Epoch 21/30
44/44 [=====] - 28s 640ms/step - loss: 0.0357
- acc: 0.9889
Epoch 22/30
44/44 [=====] - 28s 629ms/step - loss: 0.0461
- acc: 0.9889
Epoch 23/30
44/44 [=====] - 28s 640ms/step - loss: 0.0410
- acc: 0.9879
Epoch 24/30
44/44 [=====] - 29s 649ms/step - loss: 0.0531
- acc: 0.9843
Epoch 25/30
44/44 [=====] - 28s 631ms/step - loss: 0.0642
- acc: 0.9762
Epoch 26/30
44/44 [=====] - 27s 621ms/step - loss: 0.0726
- acc: 0.9787
Epoch 27/30
44/44 [=====] - 27s 624ms/step - loss: 0.0628
- acc: 0.9789
Epoch 28/30
44/44 [=====] - 27s 622ms/step - loss: 0.0523
- acc: 0.9839
Epoch 29/30
44/44 [=====] - 27s 621ms/step - loss: 0.0454
- acc: 0.9848
Epoch 30/30
44/44 [=====] - 28s 637ms/step - loss: 0.0454
- acc: 0.9835
```

<keras.callbacks.History at 0x274d7539cf8>

## SAVE THE MODEL

```
model.save('assign3.h5')
```

## TEST THE MODEL

```
data_dir=pathlib.Path(data_dir)
daisy=list(data_dir.glob('daisy/*'))
dandelion=list(data_dir.glob('dandelion/*'))
rose=list(data_dir.glob('rose/*'))
sun=list(data_dir.glob('sunflower/*'))
tulip=list(data_dir.glob('tulip/*'))

img = image.load_img(sun[0],target_size=(64,64))
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x))
print(pred, model.predict(x))
op = ['daisy','dandelion','rose','sunflower','tulip']
print(op[pred])
```

```
3 [[0. 0. 0. 1. 0.]]
sunflower
```

```
img = image.load_img(rose[1],target_size=(64,64))
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x))
print(pred, model.predict(x))
op = ['daisy','dandelion','rose','sunflower','tulip']
print(op[pred])
```

```
2 [[0. 0. 1. 0. 0.]]
rose
```

```
img = image.load_img(daisy[1],target_size=(64,64))
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x))
print(pred, model.predict(x))
op = ['daisy','dandelion','rose','sunflower','tulip']
print(op[pred])
```

```
0 [[1. 0. 0. 0. 0.]]
daisy
```

```
img = image.load_img(dandelion[19],target_size=(64,64))
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x))
print(pred, model.predict(x))
op = ['daisy','dandelion','rose','sunflower','tulip']
print(op[pred])
```

```
1 [[0. 1. 0. 0. 0.]]
dandelion
```

```
img = image.load_img(tulip[0],target_size=(64,64))
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x))
print(pred, model.predict(x))
op = ['daisy','dandelion','rose','sunflower','tulip']
print(op[pred])
```

```
4 [[0. 0. 0. 0. 1.]]
tulip
```