Assignment Date: 07 October 2022

Student Name: Niveatha Ruba.G

Student Roll Number: 611219106054

Maximum Marks: 2 Marks

Build CNN Model for Classification Of Flowers

Download the dataset here.

```
# Unzip data
!unzip '/content/Flowers-Dataset.zip'
```

```
  inflating: flowers/daisy/14372713423_61e2daae88.jpg
  inflating: flowers/daisy/14399435971_ea5868c792.jpg
  inflating: flowers/daisy/14402451388_56545a374a_n.jpg
  inflating: flowers/daisy/144076848_57e1d662e3_m.jpg
  inflating: flowers/daisy/144099102_bf63a41e4f_n.jpg
  inflating: flowers/daisy/1441939151_b271408c8d_n.jpg
  inflating: flowers/daisy/14421389519_d5fd353eb4.jpg
  inflating: flowers/daisy/144603918_b9de002f60_m.jpg
  inflating: flowers/daisy/14471433500_cdaa22e3ea_m.jpg
  inflating: flowers/daisy/14485782498_fb342ec301.jpg
  inflating: flowers/daisy/14507818175_05219b051c_m.jpg
  inflating: flowers/daisy/14523675369_97c31d0b5b.jpg
  inflating: flowers/daisy/14551098743_2842e7a004_n.jpg
  inflating: flowers/daisy/14554906452_35f066ffe9_n.jpg
  inflating: flowers/daisy/14564545365_1f1d267bf1_n.jpg
  inflating: flowers/daisy/14569895116_32f0dcb0f9.jpg
  inflating: flowers/daisy/14591326135_930703dbed_m.jpg
  inflating: flowers/daisy/14600779226_7bbc288d40_m.jpg
  inflating: flowers/daisy/14613443462_d4ed356201.jpg
  inflating: flowers/daisy/14621687774_ec52811acd_n.jpg
  inflating: flowers/daisy/14674743211_f68b13f6d9.jpg
  inflating: flowers/daisy/14698531521_0c2f0c6539.jpg
  inflating: flowers/daisy/147068564_32bb4350cc.jpg
  inflating: flowers/daisy/14707111433_cce08ee007.jpg
  inflating: flowers/daisy/14716799982_ed6d626a66.jpg
  inflating: flowers/daisy/14816364517_2423021484_m.jpg
  inflating: flowers/daisy/14866200659_6462c723cb_m.jpg
  inflating: flowers/daisy/14907815010_bff495449f.jpg
  inflating: flowers/daisy/14921511479_7b0a647795.jpg
  inflating: flowers/daisy/15029936576_8d6f96c72c_n.jpg
  inflating: flowers/daisy/15100730728_a450c5f422_n.jpg
  inflating: flowers/daisy/15207766_fc2f1d692c_n.jpg  
  inflating: flowers/daisy/15306268004_4680ba95e1.jpg  
  inflating: flowers/daisy/153210866_03cc9f2f36.jpg  
  inflating: flowers/daisy/15327813273_06cdf42210.jpg  
  inflating: flowers/daisy/154332674_453cea64f4.jpg  
  inflating: flowers/daisy/15760153042_a2a90e9da5_m.jpg  
  inflating: flowers/daisy/15760811380_4d686c892b_n.jpg  
  inflating: flowers/daisy/15784493690_b1858cdb2b_n.jpg  
  inflating: flowers/daisy/15813862117_dedcd1c56f_m.jpg  
  inflating: flowers/daisy/15853110333_229c439e7f.jpg  
  inflating: flowers/daisy/158869618_f1a6704236_n.jpg  
  inflating: flowers/daisy/1602025317 6_60f2a6a5ca_n.jpg  
  inflating: flowers/daisy/16025261368_911703a536_n.jpg  
  inflating: flowers/daisy/16056178001_bebc2153fe_n.jpg  
  inflating: flowers/daisy/16121105382_b96251e506_m.jpg  
  inflating: flowers/daisy/16161045294_70c76ce846_n.jpg  
  inflating: flowers/daisy/162362896_99c7d851c8_n.jpg  
  inflating: flowers/daisy/162362897_1d21b70621_m.jpg  
  inflating: flowers/daisy/16291797949_a1b1b7c2bd_n.jpg  
  inflating: flowers/daisy/16323838000_3818bce5c6_n.jpg  
  inflating: flowers/daisy/16360180712_b72695928c_n.jpg  
  inflating: flowers/daisy/163978992_8128b49d3e_n.jpg  
  inflating: flowers/daisy/16401288243_36112bd52f_m.jpg  
  inflating: flowers/daisy/16482676953_5296227d40_n.jpg  
  inflating: flowers/daisy/16492248512_61a57dfec1_m.jpg  
  inflating: flowers/daisy/16527403771_2391f137c4_n.jpg  
```

1. Image Augmentation

```
#import lib.

from tensorflow.keras.preprocessing.image import ImageDataGenerator


#augmentation on flowers

rose_datagen=ImageDataGenerator(rescale=1./255,
                                zoom_range=0.2,
                                horizontal_flip=True)


tulip_datagen=ImageDataGenerator(rescale=1./255,
                                zoom_range=0.2,
                                horizontal_flip=True)


xrose = rose_datagen.flow_from_directory('/content/flowers',
                                           target_size=(64,64),
                                           class_mode='categorical',
                                           batch_size=100)
```

```
     Found 4317 images belonging to 5 classes.
```

```
xtulip = tulip_datagen.flow_from_directory('/content/flowers',
                                           target_size=(64,64),
                                           class_mode='categorical',
                                           batch_size=100)
```

```
     Found 4317 images belonging to 5 classes.
```

2. Create Model

```
#import lib.

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

3. Add Layers (Convolution,MaxPooling,Flatten,Dense-(Hidden Layers),Output)

```
# Add a layers

model = Sequential() # Initializing sequential model
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3))) # convolution laye
model.add(MaxPooling2D(pool_size=(2, 2))) # Max pooling layer
model.add(Flatten()) # Flatten layer
model.add(Dense(300,activation='relu')) # Hidden layer 1
model.add(Dense(150,activation='relu')) # Hidden layer 2
model.add(Dense(5,activation='softmax')) # Output layer
```

4. Compile The Model

```
# Compiling the model

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

5. Fit The Model

```
model.fit_generator(xrose,
                    steps_per_epoch=len(xrose),
                    epochs=10,
                    validation_data=xtulip,
                    validation_steps=len(xtulip))
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: UserWarning: `Model.fit_
  """
Epoch 1/10
44/44 [==============================] - 44s 977ms/step - loss: 1.4462 - accuracy: 0.42
Epoch 2/10
44/44 [==============================] - 42s 968ms/step - loss: 1.0792 - accuracy: 0.56
Epoch 3/10
44/44 [==============================] - 42s 960ms/step - loss: 1.0447 - accuracy: 0.59
Epoch 4/10
44/44 [==============================] - 42s 949ms/step - loss: 0.9505 - accuracy: 0.63
Epoch 5/10
44/44 [==============================] - 41s 938ms/step - loss: 0.9207 - accuracy: 0.64
Epoch 6/10
44/44 [==============================] - 42s 952ms/step - loss: 0.8762 - accuracy: 0.67
Epoch 7/10
44/44 [==============================] - 42s 949ms/step - loss: 0.8472 - accuracy: 0.68
Epoch 8/10
44/44 [==============================] - 42s 949ms/step - loss: 0.8051 - accuracy: 0.69
Epoch 9/10
44/44 [==============================] - 42s 953ms/step - loss: 0.7541 - accuracy: 0.71
Epoch 10/10
44/44 [==============================] - 41s 938ms/step - loss: 0.7553 - accuracy: 0.71
<keras.callbacks.History at 0x7fe14b52ca50>
```

6. Save The Model

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

7. Test The Model

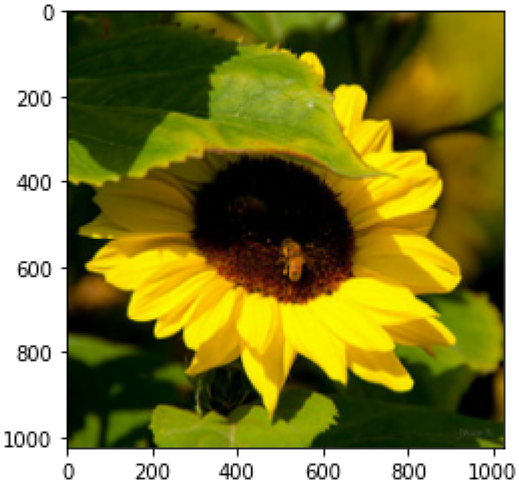
```
from tensorflow.keras.preprocessing import image
import numpy as np
import matplotlib.pyplot as plt
```

```
#testing 1
img = image.load_img('/content/flowers/sunflower/12471443383_b71e7a7480_m.jpg',target_size=(6
x = image.img_to_array(img) # Converting image into array
x = np.expand_dims(x,axis=0) # expanding Dimensions
pred = np.argmax(model.predict(x)) # Predicting the higher probablity index
```

```
op = ['daisy','dandelion','rose','sunflower','tulip'] # Creating list
op[pred] # List indexing with output
```

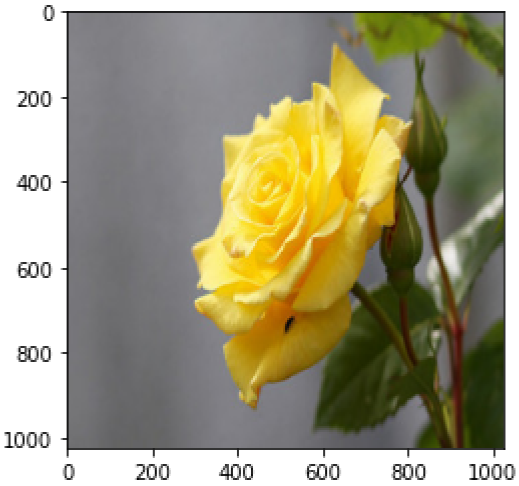
```
img = image.load_img('/content/flowers/sunflower/12471443383_b71e7a7480_m.jpg',target_size=(1
plt.imshow(img)
```

```
<matplotlib.image.AxesImage at 0x7fe1464e7890>
```



```
img = image.load_img('/content/flowers/rose/14145188939_b4de638bd3_n.jpg',target_size=(1024,1
plt.imshow(img)
```

```
<matplotlib.image.AxesImage at 0x7fe147877910>
```



Colab paid products - Cancel contracts here

● ✕