Assignment Date: 07 October 2022

Student Name: Mohanapriya C

Student Roll Number: 611219106045

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/14088053307_1a13a0bf91_n.jpg
inflating: flowers/daisy/14114116486_0bb6649bc1_m.jpg
inflating: flowers/daisy/14147016029_8d3cf2414e.jpg
inflating: flowers/daisy/14163875973_467224aaf5_m.jpg
inflating: flowers/daisy/14167534527_781ceb1b7a_n.jpg
inflating: flowers/daisy/14167543177_cd36b54ac6_n.jpg
inflating: flowers/daisy/14219214466_3ca6104eae_m.jpg
inflating: flowers/daisy/14221836990_90374e6b34.jpg
inflating: flowers/daisy/14221848160_7f0a37c395.jpg
inflating: flowers/daisy/14245834619_153624f836.jpg
inflating: flowers/daisy/14264136211_9531fbc144.jpg
inflating: flowers/daisy/14272874304_47c0a46f5a.jpg
inflating: flowers/daisy/14307766919_fac3c37a6b_m.jpg
inflating: flowers/daisy/14330343061_99478302d4_m.jpg
inflating: flowers/daisy/14332947164_9b13513c71_m.jpg
inflating: flowers/daisy/14333681205_a07c9f1752_m.jpg
inflating: flowers/daisy/14350958832_29bdd3a254.jpg
inflating: flowers/daisy/14354051035_1037b30421_n.jpg
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
```

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 l
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.f
  """
Epoch 1/10
44/44 [==============================] - 46s 1s/step - loss: 0.7158 - accuracy: 0.735
Epoch 2/10
44/44 [==============================] - 46s 1s/step - loss: 0.6948 - accuracy: 0.737
Epoch 3/10
44/44 [==============================] - 47s 1s/step - loss: 0.6612 - accuracy: 0.756
Epoch 4/10
44/44 [==============================] - 46s 1s/step - loss: 0.6112 - accuracy: 0.769
```

```
Epoch 5/10
44/44 [==============================] - 46s 1s/step - loss: 0.5903 - accuracy: 0.776
Epoch 6/10
44/44 [==============================] - 45s 1s/step - loss: 0.5854 - accuracy: 0.785
Epoch 7/10
44/44 [==============================] - 45s 1s/step - loss: 0.5399 - accuracy: 0.797
Epoch 8/10
44/44 [==============================] - 45s 1s/step - loss: 0.5052 - accuracy: 0.817
Epoch 9/10
44/44 [==============================] - 45s 1s/step - loss: 0.4940 - accuracy: 0.818
Epoch 10/10
44/44 [==============================] - 45s 1s/step - loss: 0.4596 - accuracy: 0.837
<keras.callbacks.History at 0x7f13f0946bd0>
```

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
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
```

```
'sunflower'
```

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

```
<matplotlib.image.AxesImage at 0x7f13e6c0cbd0>
```


```
img = image.load_img('/content/flowers/rose/14145188939_b4de638bd3_n.jpg',target_size=(102
plt.imshow(img)
```

```
<matplotlib.image.AxesImage at 0x7f13e6f44890>
```