

Assignment Date: 07 October 2022

Student Name: UKESH G

Student Roll Number: 611219106081

Maximum Marks: 2 Marks

▼ Build CNN Model for Classification Of Flowers

▼ Download the dataset [here](#).

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

```
Archive: /content/Flowers-Dataset.zip
  inflating: flowers/daisy/100080576_f52e8ee070_n.jpg
  inflating: flowers/daisy/10140303196_b88d3d6cec.jpg
  inflating: flowers/daisy/10172379554_b296050f82_n.jpg
  inflating: flowers/daisy/102567486_2748826a8b.jpg
  inflating: flowers/daisy/102636503_21bededa75_n.jpg
  inflating: flowers/daisy/1041525_bd6628ae3c.jpg
  inflating: flowers/daisy/10300722094_28fa978807_n.jpg
  inflating: flowers/daisy/1031799732_e7f4008c03.jpg
  inflating: flowers/daisy/10391248763_1d16681106_n.jpg
  inflating: flowers/daisy/10437754174_22ec990b77_m.jpg
  inflating: flowers/daisy/10437770546_8bb6f7bdd3_m.jpg
  inflating: flowers/daisy/10437929963_bc13eebe0c.jpg
  inflating: flowers/daisy/10466290366_cc72e33532.jpg
  inflating: flowers/daisy/10466558316_a7198b87e2.jpg
  inflating: flowers/daisy/10555749515_13a12a026e.jpg
  inflating: flowers/daisy/10555815624_dc211569b0.jpg
  inflating: flowers/daisy/10555826524_423eb8bf71_n.jpg
  inflating: flowers/daisy/10559679065_50d2b16f6d.jpg
  inflating: flowers/daisy/105806915_a9c13e2106_n.jpg
  inflating: flowers/daisy/10712722853_5632165b04.jpg
  inflating: flowers/daisy/107592979_aaa9cdf78_m.jpg
  inflating: flowers/daisy/10770585085_4742b9dac3_n.jpg
  inflating: flowers/daisy/10841136265_af473efc60.jpg
  inflating: flowers/daisy/10993710036_2033222c91.jpg
  inflating: flowers/daisy/10993818044_4c19b86c82.jpg
  inflating: flowers/daisy/10994032453_ac7f8d9e2e.jpg
  inflating: flowers/daisy/11023214096_b5b39fab08.jpg
  inflating: flowers/daisy/11023272144_fce94401f2_m.jpg
  inflating: flowers/daisy/11023277956_8980d53169_m.jpg
  inflating: flowers/daisy/11124324295_503f3a0804.jpg
  inflating: flowers/daisy/1140299375_3aa7024466.jpg
  inflating: flowers/daisy/11439894966_dca877f0cd.jpg
  inflating: flowers/daisy/1150395827_6f94a5c6e4_n.jpg
  inflating: flowers/daisy/11642632_1e7627a2cc.jpg
```

Saved successfully!

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 1
model.add(MaxPooling2D(pool_size=(2, 2))) # Max pooling layer
model.add(Flatten()) # Flatten layer
model.add(Dense(200,activation='relu')) # Hidden layer 1
model.add(Dense(200,activation='relu')) # Hidden layer 2
model.add(Dense(5,activation='softmax')) # Output layer
```

Saved successfully!

▼ 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_generator` does not support validation data when using the `Model.compile` method.
  """
```

```
Epoch 1/10
```

```
44/44 [=====] - 48s 1s/step - loss: 1.4680 - accuracy: 0.438
```

```
Epoch 2/10
```

```

44/44 [=====] - 46s 1s/step - loss: 1.0784 - accuracy: 0.568
Epoch 3/10
44/44 [=====] - 46s 1s/step - loss: 1.0102 - accuracy: 0.602
Epoch 4/10
44/44 [=====] - 46s 1s/step - loss: 0.9403 - accuracy: 0.635
Epoch 5/10
44/44 [=====] - 47s 1s/step - loss: 0.8933 - accuracy: 0.651
Epoch 6/10
44/44 [=====] - 46s 1s/step - loss: 0.8411 - accuracy: 0.682
Epoch 7/10
44/44 [=====] - 46s 1s/step - loss: 0.8039 - accuracy: 0.696
Epoch 8/10
44/44 [=====] - 46s 1s/step - loss: 0.7868 - accuracy: 0.708
Epoch 9/10
44/44 [=====] - 46s 1s/step - loss: 0.7666 - accuracy: 0.706
Epoch 10/10
44/44 [=====] - 46s 1s/step - loss: 0.7089 - accuracy: 0.731
<keras.callbacks.History at 0x7f47a9895cd0>

```

6. Save The Model

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

Saved successfully!

```

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)

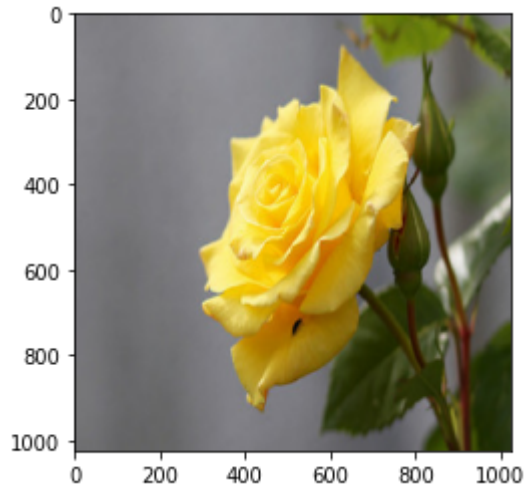
```

```
<matplotlib.image.AxesImage at 0x7f47a5ff8110>
```



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

```
<matplotlib.image.AxesImage at 0x7f47a5ade550>
```



Saved successfully!

