Assignment Date	09 October 2022
Student Name	R.SUVINKUMAR
Student Roll Number	923119106008
Maximum Marks	2 Marks

1.Download the dataset

Already downloaded in question paper.

In [34]:

```
from google.colab import drive
drive.mount('/content/drive')
Drive already mounted at /content/drive; to attempt to forcibly remount, ca
ll drive.mount("/content/drive", force_remount=True).
```

Unzip the data

In [4]:

```
# Extract data

!unzip '/content/drive/MyDrive/Flowers-Dataset (1).zip'
Archive:  /content/drive/MyDrive/Flowers-Dataset (1).zip
  inflating: flowers/daisy/100080576_f52e8ee070_n.jpg
  inflating: flowers/daisy/10140303196_b88d3d6cec.jpg
  inflating: flowers/daisy/10172379554_b296050f82_n.jpg
  inflating: flowers/daisy/10172567486_2748826a8b.jpg
  inflating: flowers/daisy/10172636503_21bededa75_n.jpg
  inflating: flowers/daisy/102841525_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
```

2.Image augmentation

In [6]:

```
# Import necessary lib.

from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```python
# Data Augmentation on training variable

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

```python
# Data Augmentation on flowers data

xtrain = train_datagen.flow_from_directory('/content/flowers',
                                           target_size=(64,64),
                                           class_mode='categorical',
                                           batch_size=100)
Found 4317 images belonging to 5 classes.
```

3.Create model

```python
# Importing reg lib

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

4.Build the model

```python
# Build a CNN block

model = Sequential() # Initializing sequential model
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))
# convolution layer
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(4,activation='softmax')) # Output layer
```

5.Compile the model

```python
# Compiling the model

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

6.Fit the model

Here we have one folder...but fit model wants two or more type of folder to fit the function......

so we couldn't fit the model....in this assignment.....!!!

7.Save model

```python
# Save model

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

8.Testing model

```python
from tensorflow.keras.preprocessing import image
import numpy as np
```

```python
# Testing the model

img = image.load_img('/content/flowers/sunflower/10386503264_e05387e1f7_m.jpg',target_size=(64,64)) # Reading image
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x))
op = ['daisy','dandelion','rose','sunflower','tulip']
op[pred]
```

```
'sunflower'
```

```python
img = image.load_img('/content/flowers/rose/11233672494_d8bf0a3dbf_n.jpg',target_size=(64,64)) # Reading image
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x))
op = ['daisy','dandelion','rose','sunflower','tulip']
op[pred]
```

```
'sunflower'
```

```python
img = image.load_img('/content/flowers/dandelion/8083321316_f62ea76f72_n.jpg',target_size=(64,64)) # Reading image
x = image.img_to_array(img)
```

```
x = np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x))
op = ['daisy','dandelion','rose','sunflower','tulip']
op[pred]
```

```
'dandelion'
```

```
xtrain.class_indices
```

```
{'daisy': 0, 'dandelion': 1, 'rose': 2, 'sunflower': 3, 'tulip': 4}
```

```
x = np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x))
op = ['daisy','dandelion','rose','sunflower','tulip']
op[pred]
```