In [4]:

```
!unzip '/content/flowers'
```

```
Archive:  /content/Flowers-Dataset.zip
replace flowers/daisy/100080576_f52e8ee070_n.jpg? [y]es, [n]o, [A]ll, [N]one, [r]ename:
```

In [6]:

```
#DATA AUGUMENTATION

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

In [7]:

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

In [8]:

```
test_datagen = ImageDataGenerator(rescale=1./255)
```

In [14]:

```
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.
```

In [15]:

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

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

CNN MODEL TRAINING

In [10]:

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

In [11]:

```
model = Sequential()

#CONVOLUTION LAYER
model.add(Convolution2D(32,(3,3),activation='relu', input_shape=(64,64,3)))

#MAX POOLING LATER
model.add(MaxPooling2D(pool_size=(2, 2)))

#FLATTEN
model.add(Flatten())

#FULLY CONNECTED LAYER
model.add(Dense(400,activation='relu'))
model.add(Dense(300,activation='relu'))

#OUTPUT
model.add(Dense(5,activation='softmax'))
```

In [12]:

```
# COMPILE

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

In [16]:

```
#FIT THE MODEL

model.fit_generator(xtrain,
                    steps_per_epoch=len(xtrain),
                    epochs=10,
                    validation_data=xtest,
                    validation_steps=len(xtest),
)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:7: UserWarning: `Model.fit_generator` is deprecated and
will be removed in a future version. Please use `Model.fit`, which supports generators.
  import sys
```

```
Epoch 1/10
44/44 [==============================] - 31s 504ms/step - loss: 1.7737 - accuracy: 0.3639 - val_loss: 1.2427 - val_ac
curacy: 0.4605
Epoch 2/10
44/44 [==============================] - 22s 501ms/step - loss: 1.1417 - accuracy: 0.5298 - val_loss: 1.0688 - val_ac
curacy: 0.5826
Epoch 3/10
44/44 [==============================] - 22s 502ms/step - loss: 1.0359 - accuracy: 0.5958 - val_loss: 0.9995 - val_ac
curacy: 0.6224
Epoch 4/10
44/44 [==============================] - 22s 502ms/step - loss: 0.9638 - accuracy: 0.6301 - val_loss: 1.0157 - val_ac
curacy: 0.6090
Epoch 5/10
44/44 [==============================] - 22s 502ms/step - loss: 0.9027 - accuracy: 0.6484 - val_loss: 0.8879 - val_ac
curacy: 0.6715
Epoch 6/10
44/44 [==============================] - 22s 500ms/step - loss: 0.8541 - accuracy: 0.6769 - val_loss: 0.8847 - val_ac
curacy: 0.6725
Epoch 7/10
44/44 [==============================] - 22s 500ms/step - loss: 0.8241 - accuracy: 0.6921 - val_loss: 0.9194 - val_ac
curacy: 0.6379
Epoch 8/10
44/44 [==============================] - 22s 497ms/step - loss: 0.7962 - accuracy: 0.6921 - val_loss: 0.7008 - val_ac
curacy: 0.7362
Epoch 9/10
44/44 [==============================] - 22s 502ms/step - loss: 0.7265 - accuracy: 0.7283 - val_loss: 0.6862 - val_ac
curacy: 0.7445
Epoch 10/10
44/44 [==============================] - 22s 496ms/step - loss: 0.7160 - accuracy: 0.7359 - val_loss: 0.6717 - val_ac
curacy: 0.7399
```

Out[16]:

```
<keras.callbacks.History at 0x7fec702daa90>
```

In [17]:

```
#SAVE MODEL

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

In [18]:

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

In [24]:

```
#TESTING

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

Out[24]:

```
'dandelion'
```