

In [6]:

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

In [2]:

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

In []:

```
!unzip '/content/drive/MyDrive/Flowers-Dataset.zip'
```

In [4]:

```
from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

In [7]:

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

In [8]:

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

In [11]:

```
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 [13]:

```
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.

Create Model

In [14]:

```
model = Sequential()
```

Convolution Layer

In [15]:

```
model.add(Convolution2D(32, (3, 3), activation='relu', input_shape=(64, 64, 3)))
```

MaxPooling

In [16]:

```
model.add(MaxPooling2D(pool_size=(2, 2)))
```

Flatten

In [17]:

```
model.add(Flatten())
```

Dense Layer

In [18]:

```
model.add(Dense(300,activation='relu')) #hiddenlayer 1
model.add(Dense(150,activation='relu')) #hiddenlayer 2
```

Output

In [19]:

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

Compile the model

In [20]:

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

Fit The Model

In [21]:

```
model.fit_generator(xtrain,steps_per_epoch=108,epochs=30,validation_data=xtest,validation_steps=27)
```

Epoch 1/30

44/108 [=====>.....] - ETA: 24s - loss: 1.5416 - accuracy: 0.3903

WARNING:tensorflow:Your input ran out of data; interrupting training. Make sure that your dataset or generator can generate at least `steps_per_epoch * epochs` batches (in this case, 3240 batches). You may need to use the repeat() function when building your dataset.

108/108 [=====] - 24s 215ms/step - loss: 1.5416 - accuracy: 0.3903 - val_loss: 1.1618 - val_accuracy: 0.5141

Out[21]:

<keras.callbacks.History at 0x7fc50ee41650>

Save The Model

In [22]:

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

Test The Model

In [23]:

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

In [25]:

```
img = image.load_img('/content/flowers/tulip/8690789564_394eb04982_n.jpg',target_size=(64,64))
```

In [26]:

```
img
```

```
Out[26]:
```



```
In [27]:
```

```
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
```

```
In [28]:
```

```
pred_prob=model.predict(x)
```

```
1/1 [=====] - 0s 72ms/step
```

```
In [29]:
```

```
class_name=["Daisy","Dandelion","Rose","Sunflower","Tulip"]
```

```
In [30]:
```

```
pred_id=pred_prob.argmax(axis=1)[0]
pred_id
```

```
Out[30]:
```

```
4
```

```
In [31]:
```

```
print("Predicted flower is",str(class_name[pred_id]))
```

```
Predicted flower is Tulip
```