

In [1]:

```
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 [3]:

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

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

In []:

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

Image Augmentation

In [5]:

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

In [6]:

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

In [7]:

```
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 [8]:

```
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.

Model

Compile The Model

In [9]:

```
model = Sequential()
model.add(Convolution2D(32, (3,3), activation = 'relu', input_shape = (64,64,3)))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Flatten())
model.add(Dense(300, activation = 'relu'))
```

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

In [10]:

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

Fit The Model

In [11]:

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

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.  
    """Entry point for launching an IPython kernel.
```

Epoch 1/30

44/108 [=====>.....] - ETA: 41s - loss: 1.9497 - accuracy: 0.3537

```
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 [=====] - 40s 355ms/step - loss: 1.9497 - accuracy: 0.3537 - val_loss: 1.2284 - val_accuracy: 0.4837

Out[11]:

```
<keras.callbacks.History at 0x7f5e55e9ce50>
```

Save The Model

In [12]:

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

Test The Model

In [13]:

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

In [14]:

```
img = image.load_img('/content/flowers/tulip/9976515506_d496c5e72c.jpg', target_size=(64, 64))
```

In [15]:

```
img
```

Out[15]:



In [24]:

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

In [23]:

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

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

```
1/1 [=====] - 0s 22ms/step
```

```
In [25]:
```

```
op = ['daisy', 'dandelion', 'rose', 'sunflower', 'tulip']  
op[pred]
```

```
Out[25]:
```

```
'tulip'
```