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

```
Mounted at /content/drive
```

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,
zoom_range=0.2,horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)
```

Image Augmentation

```
x_train = train_datagen.flow_from_directory('/content/drive/MyDrive/Flowers-Dataset/flowers',
target_size=(64,64),
class_mode='categorical',
batch_size=100)
```

```
Found 2880 images belonging to 5 classes.
```

```
x_test=test_datagen.flow_from_directory('/content/drive/MyDrive/Flowers-Dataset/flowers',
target_size=(64,64),
class_mode='categorical',
batch_size=100)
```

```
Found 2880 images belonging to 5 classes.
```

```
x_train.class_indices
```

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

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

INITIALISING AND CREATING MODEL

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

```
model.add(Dense(150,activation='relu'))
model.add(Dense(5,activation='softmax'))

model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 62, 62, 32)        896

 max_pooling2d (MaxPooling2D  (None, 31, 31, 32)       0
 )

 flatten (Flatten)           (None, 30752)             0

 dense (Dense)               (None, 300)               9225900

 dense_1 (Dense)             (None, 150)               45150

 dense_2 (Dense)             (None, 5)                 755

=================================================================
Total params: 9,272,701
Trainable params: 9,272,701
Non-trainable params: 0
_________________________________________________________________
```

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

```
model.fit_generator(x_train,steps_per_epoch=len(x_train), validation_data=x_test, validation_
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: `Model.f
  """Entry point for launching an IPython kernel.
Epoch 1/30
29/29 [==============================] - 70s 2s/step - loss: 0.9804 - accuracy: 0.557
Epoch 2/30
29/29 [==============================] - 42s 1s/step - loss: 0.8774 - accuracy: 0.622
Epoch 3/30
29/29 [==============================] - 43s 1s/step - loss: 0.8238 - accuracy: 0.651
Epoch 4/30
29/29 [==============================] - 42s 1s/step - loss: 0.7804 - accuracy: 0.670
Epoch 5/30
29/29 [==============================] - 43s 1s/step - loss: 0.7300 - accuracy: 0.700
Epoch 6/30
29/29 [==============================] - 42s 1s/step - loss: 0.7191 - accuracy: 0.701
Epoch 7/30
29/29 [==============================] - 42s 1s/step - loss: 0.6712 - accuracy: 0.734
Epoch 8/30
29/29 [==============================] - 45s 2s/step - loss: 0.6495 - accuracy: 0.732
Epoch 9/30
29/29 [==============================] - 43s 2s/step - loss: 0.6430 - accuracy: 0.744
Epoch 10/30
29/29 [==============================] - 44s 2s/step - loss: 0.6198 - accuracy: 0.745
```

```
Epoch 11/30
29/29 [==============================] - 43s 1s/step - loss: 0.5869 - accuracy: 0.758
Epoch 12/30
29/29 [==============================] - 45s 2s/step - loss: 0.5547 - accuracy: 0.780
Epoch 13/30
29/29 [==============================] - 43s 1s/step - loss: 0.5109 - accuracy: 0.792
Epoch 14/30
29/29 [==============================] - 43s 1s/step - loss: 0.5060 - accuracy: 0.801
Epoch 15/30
29/29 [==============================] - 44s 2s/step - loss: 0.4765 - accuracy: 0.808
Epoch 16/30
29/29 [==============================] - 43s 1s/step - loss: 0.4847 - accuracy: 0.810
Epoch 17/30
29/29 [==============================] - 43s 1s/step - loss: 0.4389 - accuracy: 0.824
Epoch 18/30
29/29 [==============================] - 43s 1s/step - loss: 0.4335 - accuracy: 0.828
Epoch 19/30
29/29 [==============================] - 43s 1s/step - loss: 0.4049 - accuracy: 0.843
Epoch 20/30
29/29 [==============================] - 42s 1s/step - loss: 0.4192 - accuracy: 0.837
Epoch 21/30
29/29 [==============================] - 42s 1s/step - loss: 0.3605 - accuracy: 0.864
Epoch 22/30
29/29 [==============================] - 42s 1s/step - loss: 0.3529 - accuracy: 0.863
Epoch 23/30
29/29 [==============================] - 44s 2s/step - loss: 0.3266 - accuracy: 0.886
Epoch 24/30
29/29 [==============================] - 44s 2s/step - loss: 0.3084 - accuracy: 0.881
Epoch 25/30
29/29 [==============================] - 42s 1s/step - loss: 0.3314 - accuracy: 0.871
Epoch 26/30
29/29 [==============================] - 43s 1s/step - loss: 0.3209 - accuracy: 0.876
Epoch 27/30
29/29 [==============================] - 44s 2s/step - loss: 0.2855 - accuracy: 0.897
Epoch 28/30
```

SAVE THE MODEL

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

Test the model

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

```
model=load_model('flowers.h5')
```

```
val = list(x_train.class_indices.keys())
val
```

```
['daisy', 'dandelion', 'rose', 'sunflower', 'tulip']
```

```
img=image.load_img("/content/drive/MyDrive/Flowers-Dataset/flowers/rose/12240303_80d87f77a3_n
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
y=np.argmax(model.predict(x),axis=1)
```

```
1/1 [==============================] - 0s 104ms/step
```

```
index=['daisy','dandelion','rose','sunflower','tulip']
index[y[0]]
```

```
'tulip'
```