Assignment -3

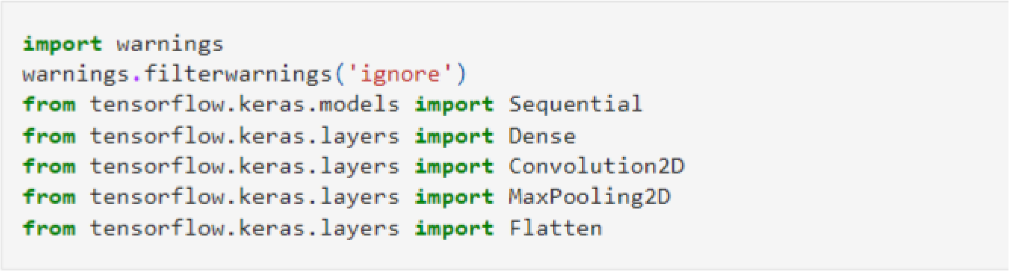
Build CNN Model for Classification of Flowers

Assignment Date	01 October 2022
Student Name	Thahir Ahamed T
Student Roll Number	910019106045

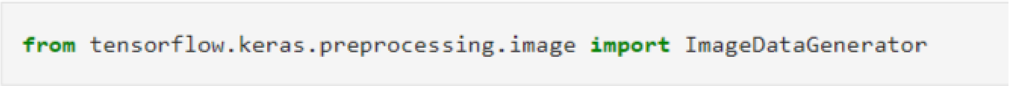
- ## Download the Dataset: - Dataset

# The dataset of images of flowers is downloaded and uploaded into the Colab files and then unzipped

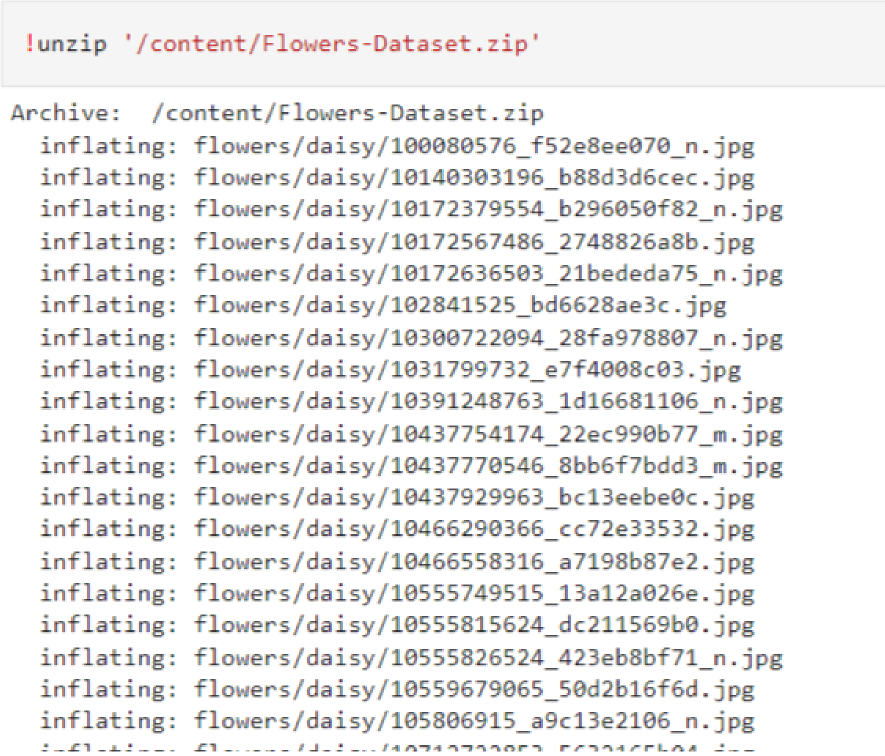
- ## Import the necessary libraries



```
import warnings
warnings.filterwarnings('ignore')
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
```



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



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

```
Archive:  /content/Flowers-Dataset.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
  inflating: flowers/daisy/10559679065_50d2b16f6d.jpg
  inflating: flowers/daisy/105806915_a9c13e2106_n.jpg
  inflating: flowers/daisy/10712722853_5632165b04.jpg
```

- ## Image Agumentation

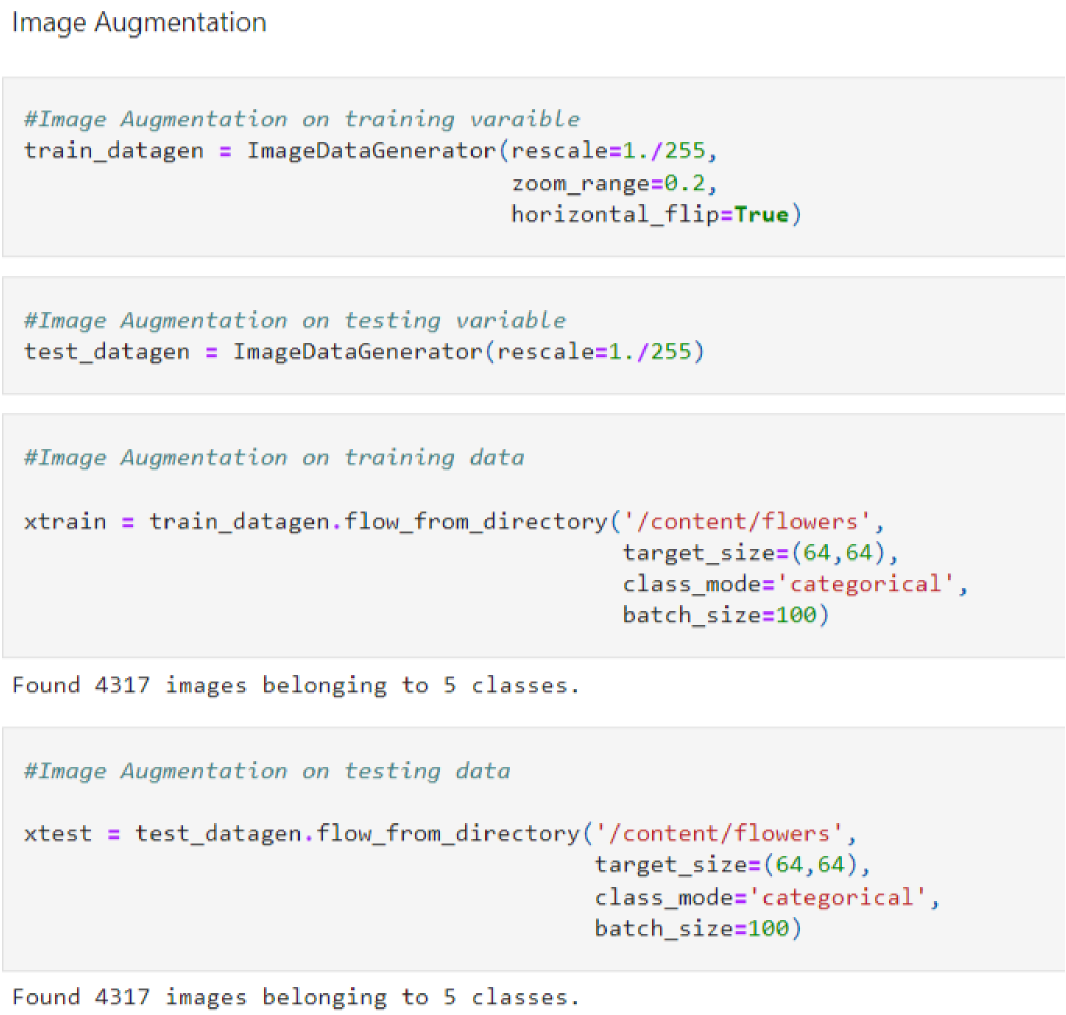


Image Augmentation

```
#Image Augmentation on training varaible
train_datagen = ImageDataGenerator(rescale=1./255,
                                   zoom_range=0.2,
                                   horizontal_flip=True)
```

```
#Image Augmentation on testing variable
test_datagen = ImageDataGenerator(rescale=1./255)
```

```
#Image Augmentation on training data

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

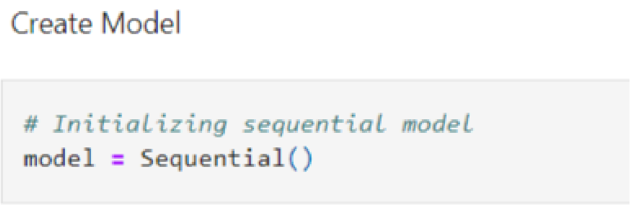
```
Found 4317 images belonging to 5 classes.
```

```
#Image Augmentation on testing data

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.
```

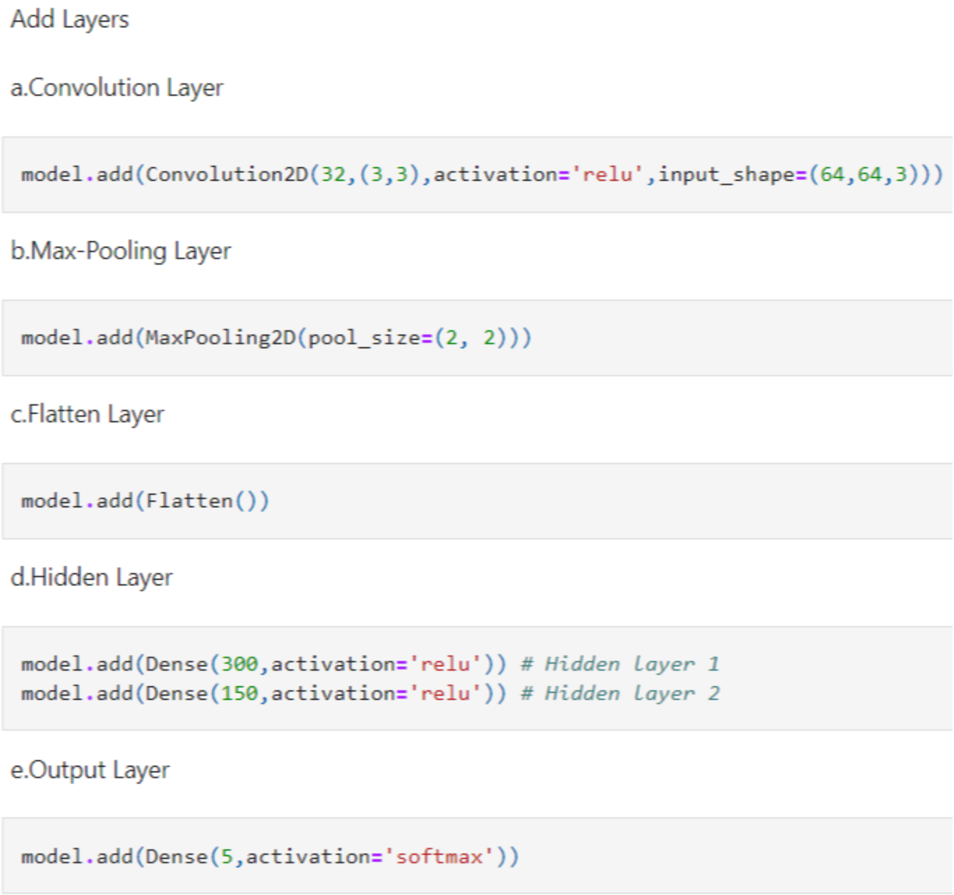
- ## Create Model



Create Model

```
# Initializing sequential model
model = Sequential()
```

- **Add Layers (Convolution,MaxPooling,Flatten,Dense-(Hidden Layers), Output)**



Add Layers

a.Convolution Layer

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

b.Max-Pooling Layer

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

c.Flatten Layer

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

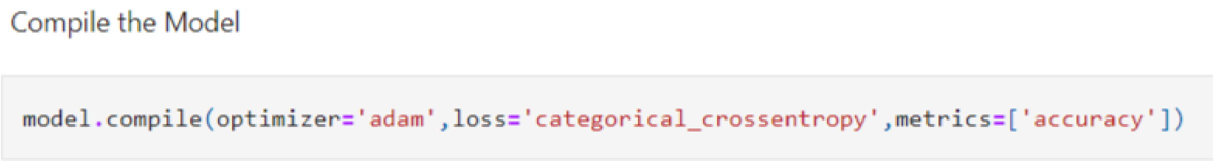
d.Hidden Layer

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

e.Output Layer

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

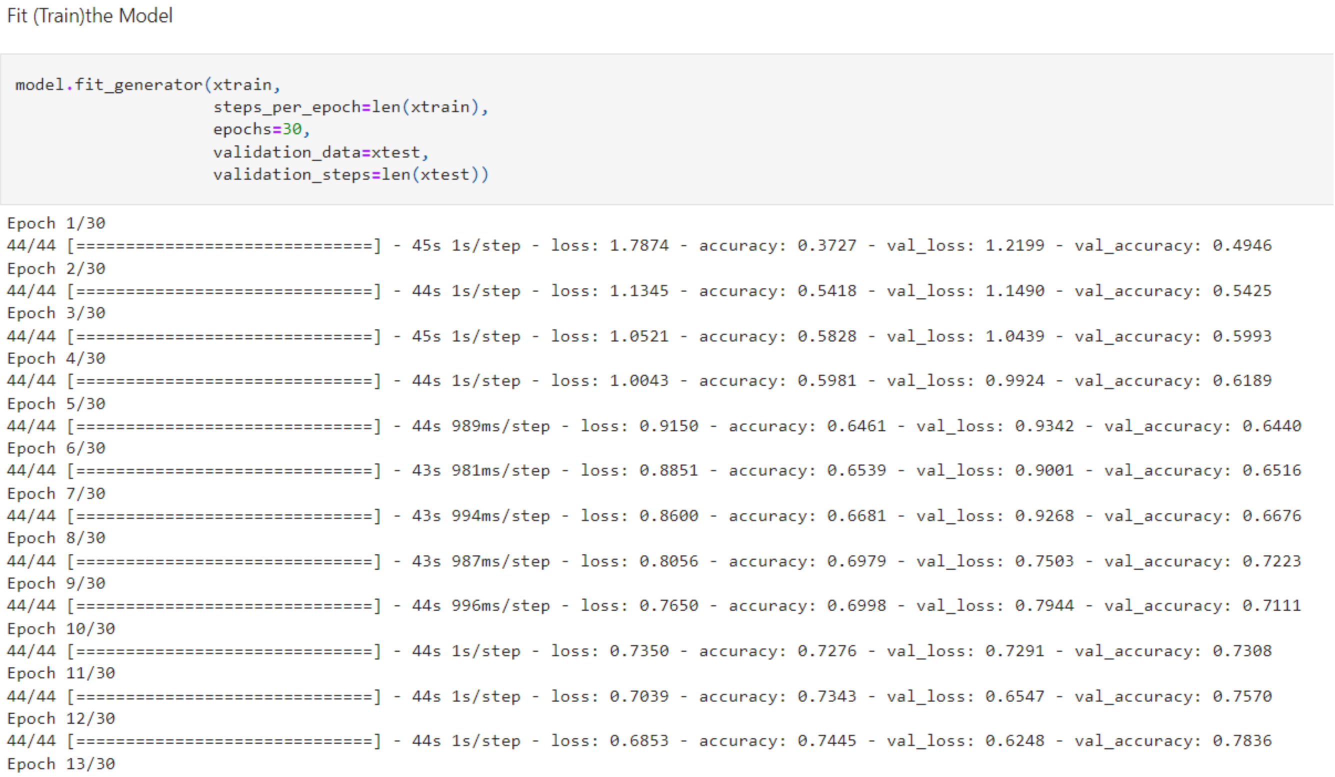
- **Compile the Model**



Compile the Model

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

- ## Fit the Model

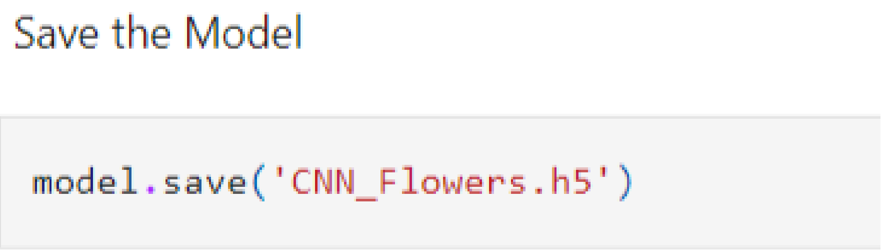


Fit (Train)the Model

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

```
Epoch 1/30
44/44 [==============================] - 45s 1s/step - loss: 1.7874 - accuracy: 0.3727 - val_loss: 1.2199 - val_accuracy: 0.4946
Epoch 2/30
44/44 [==============================] - 44s 1s/step - loss: 1.1345 - accuracy: 0.5418 - val_loss: 1.1490 - val_accuracy: 0.5425
Epoch 3/30
44/44 [==============================] - 45s 1s/step - loss: 1.0521 - accuracy: 0.5828 - val_loss: 1.0439 - val_accuracy: 0.5993
Epoch 4/30
44/44 [==============================] - 44s 1s/step - loss: 1.0043 - accuracy: 0.5981 - val_loss: 0.9924 - val_accuracy: 0.6189
Epoch 5/30
44/44 [==============================] - 44s 989ms/step - loss: 0.9150 - accuracy: 0.6461 - val_loss: 0.9342 - val_accuracy: 0.6440
Epoch 6/30
44/44 [==============================] - 43s 981ms/step - loss: 0.8851 - accuracy: 0.6539 - val_loss: 0.9001 - val_accuracy: 0.6516
Epoch 7/30
44/44 [==============================] - 43s 994ms/step - loss: 0.8600 - accuracy: 0.6681 - val_loss: 0.9268 - val_accuracy: 0.6676
Epoch 8/30
44/44 [==============================] - 43s 987ms/step - loss: 0.8056 - accuracy: 0.6979 - val_loss: 0.7503 - val_accuracy: 0.7223
Epoch 9/30
44/44 [==============================] - 44s 996ms/step - loss: 0.7650 - accuracy: 0.6998 - val_loss: 0.7944 - val_accuracy: 0.7111
Epoch 10/30
44/44 [==============================] - 44s 1s/step - loss: 0.7350 - accuracy: 0.7276 - val_loss: 0.7291 - val_accuracy: 0.7308
Epoch 11/30
44/44 [==============================] - 44s 1s/step - loss: 0.7039 - accuracy: 0.7343 - val_loss: 0.6547 - val_accuracy: 0.7570
Epoch 12/30
44/44 [==============================] - 44s 1s/step - loss: 0.6853 - accuracy: 0.7445 - val_loss: 0.6248 - val_accuracy: 0.7836
Epoch 13/30
```

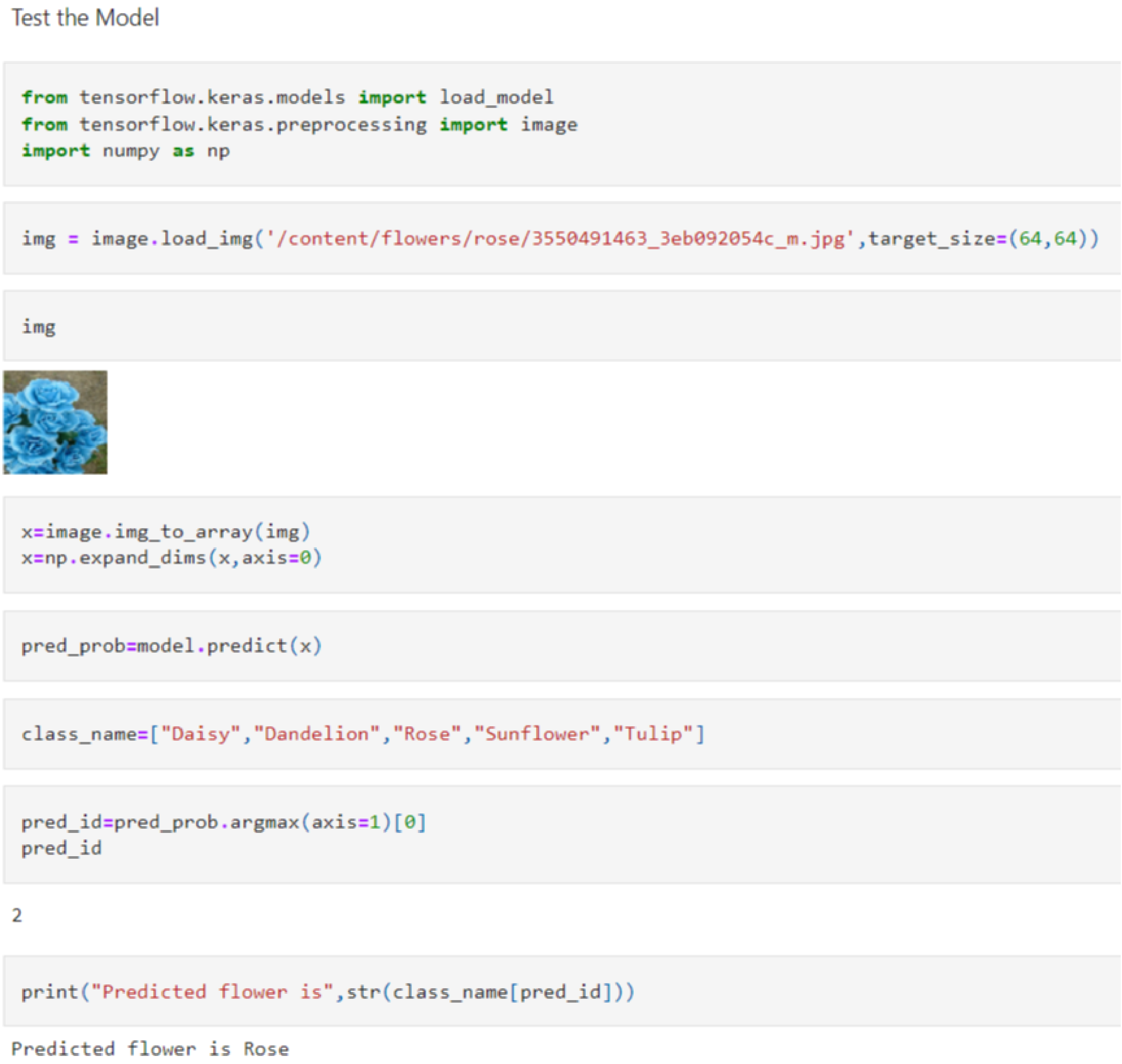
- ## Save The Model



Save the Model

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

- **Test The Model**



Test the Model

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

```
img = image.load_img('/content/flowers/rose/3550491463_3eb092054c_m.jpg',target_size=(64,64))
```

```
img
```

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

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

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

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

```
2
```

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

```
Predicted flower is Rose
```