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

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.
```

```
ls
```

```
drive/  sample_data/
```

```
cd /content/drive/MyDrive/IBM_PROJECTS/images
```

```
/content/drive/MyDrive/IBM_PROJECTS/images
```

```
pwd
```

```
'/content/drive/MyDrive/IBM_PROJECTS/images'
```

```
!unzip flowers.zip
```

```
Archive:  flowers.zip
  inflating: images/f1.jpg
  inflating: images/f2.jpg
  inflating: images/f3.jpg
  inflating: images/f4.jpg
  inflating: images/f5.jpg
```

Image Augmentation

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

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



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

```
x_train = train_datagen.flow_from_directory(r"/content/drive/MyDrive/IBM_PROJECTS/images",
```

```
Found 5 images belonging to 1 classes.
```

```
x_test = test_datagen.flow_from_directory(r"/content/drive/MyDrive/IBM_PROJECTS/images",ta
```

```
Found 5 images belonging to 1 classes.
```

```
x_train.class_indices
```

```
{'images': 0}
```

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

```
model = Sequential()
```

```
model.add(Convolution2D(32,(3,3),activation="relu",strides=(1, 1),input_shape =(64,64,3)))
```

```
model.add(MaxPooling2D(strides=(1, 1)))
```

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

```
model.summary()
```

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

 max_pooling2d (MaxPooling2D  (None, 61, 61, 32)       0
 )

 flatten (Flatten)           (None, 119072)            0

=================================================================
Total params: 896
Trainable params: 896
Non-trainable params: 0
_________________________________________________________________
```

```
model.add(Dense(300,activation="relu"))
model.add(Dense(300,activation="relu"))
```



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

```
len(x_train)
```

```
1
```

```
model.fit(x_train,epochs = 10,steps_per_epoch=len(x_train),validation_data=x_test,validati
```

```
Epoch 1/10
1/1 [==============================] - 2s 2s/step - loss: 8.0543 - accuracy: 0.6000
Epoch 2/10
1/1 [==============================] - 1s 693ms/step - loss: 14.1376 - accuracy: 0.0
Epoch 3/10
```

```
1/1 [==============================] - 1s 680ms/step - loss: 57.1539 - accuracy: 1.0
Epoch 4/10
1/1 [==============================] - 1s 696ms/step - loss: 47.3888 - accuracy: 0.8
Epoch 5/10
1/1 [==============================] - 1s 666ms/step - loss: 72.8625 - accuracy: 0.0
Epoch 6/10
1/1 [==============================] - 1s 647ms/step - loss: 63.3128 - accuracy: 0.0
Epoch 7/10
1/1 [==============================] - 1s 649ms/step - loss: 51.0375 - accuracy: 0.0
Epoch 8/10
1/1 [==============================] - 1s 648ms/step - loss: 39.3324 - accuracy: 0.0
Epoch 9/10
1/1 [==============================] - 1s 813ms/step - loss: 37.9547 - accuracy: 1.0
Epoch 10/10
1/1 [==============================] - 1s 736ms/step - loss: 63.6433 - accuracy: 0.0
<keras.callbacks.History at 0x7fa54c0ccbd0>
```

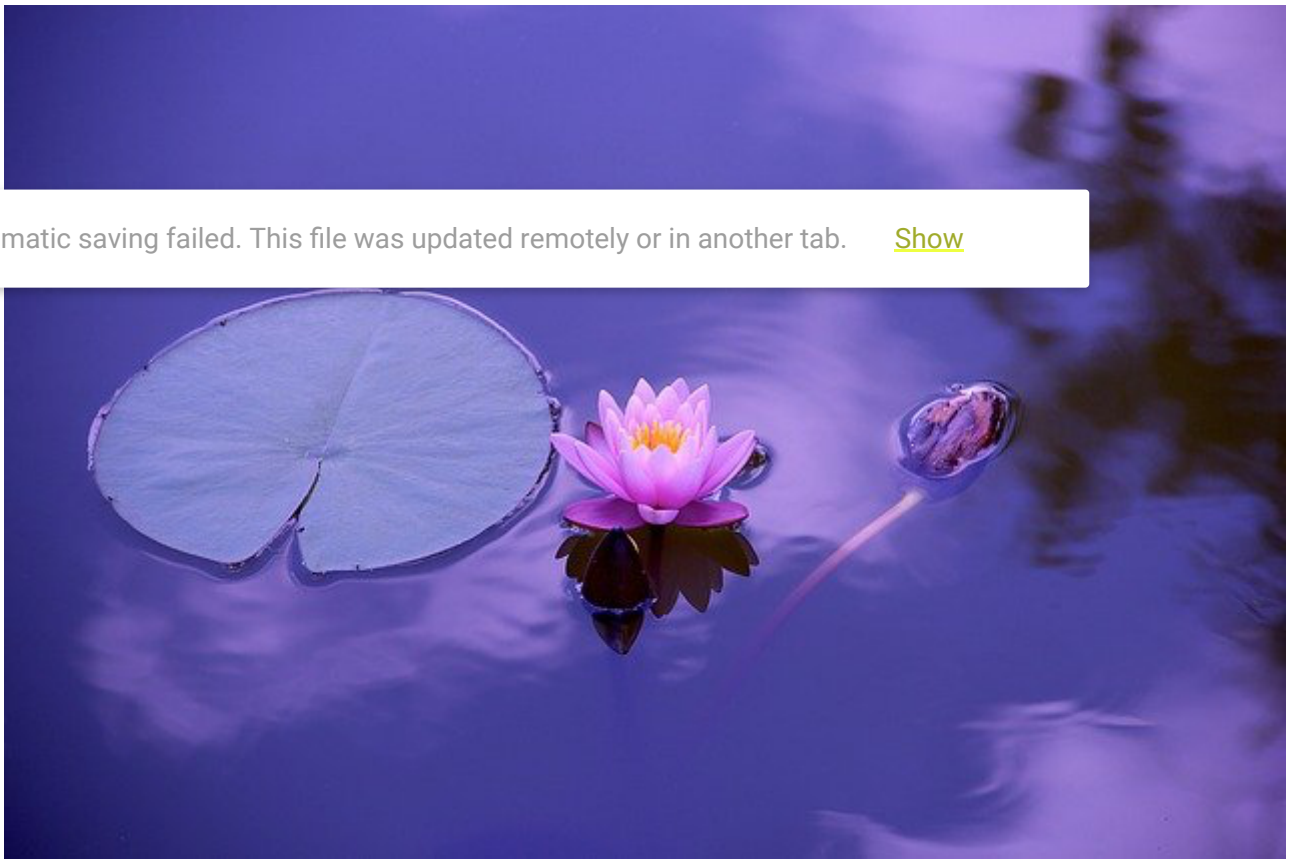
```
model.save("flower.h5")
```

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

```
model = load_model("flower.h5")
```

```
img = image.load_img(r"/content/drive/MyDrive/IBM_PROJECTS/images/f3.jpg")
```

```
img
```



```
img = image.load_img(r"/content/drive/MyDrive/IBM_PROJECTS/images/f5.jpg",target_size=(64,
```

```
img
```

```
x = image.img_to_array(img)
```

```
x
```

```
array([[[  6.,  57., 114.],
        [ 13.,  97., 161.],
        [ 14., 113., 180.],
        ...,
        [  7.,  47., 106.],
        [  9.,  57., 121.],
        [  7.,  60., 126.]],

       [[ 11.,  52., 104.],
        [ 12.,  90., 156.],
        [ 16., 115., 182.],
        ...,
        [  5.,  62., 129.],
        [  9.,  66., 135.],
        [  9.,  72., 143.]],

       [[  4.,  65., 128.],
        [ 12., 100., 172.],
        [ 17., 126., 195.],
        ...,
        [ 10.,  73., 144.],
        [ 10.,  79., 154.],
```



```
       [[  7.,  57., 120.],
        [  8.,  54., 113.],
        [  6.,  46.,  97.],
        ...,
        [ 12.,  63., 154.],
        [ 40., 142., 224.],
        [ 38., 131., 211.]],

       [[  7.,  55., 117.],
        [  6.,  51., 108.],
        [  7.,  45.,  92.],
        ...,
        [ 10.,  24.,  69.],
        [ 31., 116., 207.],
        [ 33., 122., 204.]],
```

```
             [[  6.,  55., 114.],
              [  7.,  48., 104.],
              [  8.,  45.,  90.],
              ...,
              [  1.,   7.,  29.],
              [ 21.,  83., 170.],
              [ 36., 126., 204.]]], dtype=float32)
```

```
x = np.expand_dims(x,axis = 0)
```

```
x
```

```
array([[[[  6.,  57., 114.],
         [ 13.,  97., 161.],
         [ 14., 113., 180.],
         ...,
         [  7.,  47., 106.],
         [  9.,  57., 121.],
         [  7.,  60., 126.]],

        [[ 11.,  52., 104.],
         [ 12.,  90., 156.],
         [ 16., 115., 182.],
         ...,
         [  5.,  62., 129.],
         [  9.,  66., 135.],
         [  9.,  72., 143.]],

        [[  4.,  65., 128.],
         [ 12., 100., 172.],
         [ 17., 126., 195.],
         ...,
         [ 10.,  73., 144.],
         [ 10.,  79., 154.],
         [  6.,  84., 156.]],

        ...,
```



```
         [  6.,  46.,  97.],
         ...,
         [ 12.,  63., 154.],
         [ 40., 142., 224.],
         [ 38., 131., 211.]],

        [[  7.,  55., 117.],
         [  6.,  51., 108.],
         [  7.,  45.,  92.],
         ...,
         [ 10.,  24.,  69.],
         [ 31., 116., 207.],
         [ 33., 122., 204.]],

        [[  6.,  55., 114.],
         [  7.,  48., 104.],
         [  8.,  45.,  90.],
```

```
               ...,
               [  1.,   7.,  29.],
               [ 21.,  83., 170.],
               [ 36., 126., 204.]]]], dtype=float32)
```

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

```
     1/1 [==============================] - 0s 184ms/step
```

```
pred
```

```
     array([[0., 0., 0., 1., 0.]], dtype=float32)
```

```
x_test.class_indices
```

```
index = ["","images"]
```

```
img = image.load_img(r"/content/drive/MyDrive/IBM_PROJECTS/images/f2.jpg",target_size=(64,
```

```
img
```

```
img = image.load_img(r"/content/drive/MyDrive/IBM_PROJECTS/images/f2.jpg")
```

```
img
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

Colab paid products - Cancel contracts here

✓ 0s completed at 12:23

Automatic saving failed. This file was updated remotely or in another tab. Show diff