```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.preprocessing.image import ImageDataGenerator

IMG_SIZE = 244
BATCH_SIZE = 32

train_datagen =
ImageDataGenerator(rescale=1./255,validation_split=0.2)
train_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/flower',
    target_size=(IMG_SIZE,IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    subset='training'
)

val_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/flower',
    target_size=(IMG_SIZE,IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    subset='validation'
)
```

```
Found 41 images belonging to 1 classes.
Found 10 images belonging to 1 classes.
```

```
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).
```

```
# Define the model
model = keras.Sequential([
    layers.Conv2D(32,
(3,3),activation='relu',input_shape=(IMG_SIZE,IMG_SIZE,3)),
    layers.MaxPooling2D(2,2),
    layers.Conv2D(64,(3,3),activation='relu'),
    layers.MaxPooling2D(2,2),
    layers.Conv2D(128,(3,3),activation='relu'),
    layers.MaxPooling2D(2,2),
    layers.Flatten(),
    layers.Dense(128,activation='relu'),
    layers.Dense(1,activation='sigmoid') #output layer
])
```

```
#compile the model
model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['ac
curacy'])

model.fit(train_generator,validation_data=val_generator,epochs=5)

Epoch 1/5
2/2 [==============================] - 11s 4s/step - loss: 0.5187 -
accuracy: 1.0000 - val_loss: 8.4070e-14 - val_accuracy: 1.0000
Epoch 2/5
2/2 [==============================] - 7s 2s/step - loss: 6.4720e-20 -
accuracy: 1.0000 - val_loss: 8.7206e-28 - val_accuracy: 1.0000
Epoch 3/5
2/2 [==============================] - 9s 7s/step - loss: 0.0000e+00 -
accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 4/5
2/2 [==============================] - 8s 2s/step - loss: 0.0000e+00 -
accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 5/5
2/2 [==============================] - 9s 2s/step - loss: 0.0000e+00 -
accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000

<keras.src.callbacks.History at 0x7d0bb6c51150>

model.save("Model.h5","label.txt")

/usr/local/lib/python3.10/dist-packages/keras/src/engine/
training.py:3103: UserWarning: You are saving your model as an HDF5
file via `model.save()`. This file format is considered legacy. We
recommend using instead the native Keras format, e.g.
`model.save('my_model.keras')`.
  saving_api.save_model(

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

model = load_model('/content/Model h5')
test_image_path
='/content/drive/MyDrive/flower/flowers/bougainvillea_00002.jpg'
img = image.load_img(test_image_path, target_size=(244,244))
img_array = image.img_to_array(img)
img_array = np.expand_dims(img_array,axis=0)
# Add batch dimension
img_array /= 255. #Normalize the pixel values
#Make predictions
prediction = model.predict(img_array)
#Print the prediction
print(prediction)
```

```
1/1 [==============================] - 0s 160ms/step
[[1.]]
```

```
if prediction > 0.5:
    print("flower")
else:
    print( "No flower")
```

```
flower
```

```
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).
```