```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np
import matplotlib.pyplot as plt


img_s=224
batch_s=32


train_datagen = ImageDataGenerator(rescale=1./255,validation_split=0.2)
```

Double-click (or enter) to edit

```
train_gene=train_datagen.flow_from_directory('/content/drive/MyDrive/flower_images',target_size=(img_s,img_s),batch_size=batch_s,class_m
```

```
Found 4000 images belonging to 2 classes.
```

```
val_generator=train_datagen.flow_from_directory('/content/drive/MyDrive/flower_images',target_size=(img_s,img_s),batch_size=batch_s,clas
```

```
Found 1000 images belonging to 2 classes.
```

```
model=keras.Sequential([
    layers.Conv2D(32,(3,3),activation='relu',input_shape=(img_s,img_s,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')



])
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/convolutional/base_conv.py:107: UserWarning: Do not pass an `input_shape`/`
super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

```
model.summary()
```

```
                                                                 Model: "sequential"
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━┓
┃ Layer (type)                     ┃ Output Shape                ┃         Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━┩
│ conv2d (Conv2D)                  │ (None, 222, 222, 32)        │             896 │
├──────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ max_pooling2d (MaxPooling2D)     │ (None, 111, 111, 32)        │               0 │
├──────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ conv2d_1 (Conv2D)                │ (None, 109, 109, 64)        │          18,496 │
├──────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ max_pooling2d_1 (MaxPooling2D)   │ (None, 54, 54, 64)          │               0 │
├──────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ conv2d_2 (Conv2D)                │ (None, 52, 52, 128)         │          73,856 │
├──────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ max_pooling2d_2 (MaxPooling2D)   │ (None, 26, 26, 128)         │               0 │
├──────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ flatten (Flatten)                │ (None, 86528)               │               0 │
├──────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ dense (Dense)                    │ (None, 128)                 │      11,075,712 │
├──────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ dense_1 (Dense)                  │ (None, 1)                   │             129 │
└──────────────────────────────────┴─────────────────────────────┴─────────────────┘
                                                     Total params: 11,169,089 (42.61 MB)
```

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

```
model.fit(train_gene,epochs=5,validation_data=val_generator)
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:121: UserWarning: Your `PyDataset` cl
  self._warn_if_super_not_called()
Epoch 1/5
125/125 ━━━━━━━━━━━━━━━━━━━━ 873s 7s/step - accuracy: 0.8951 - loss: -86104.0000 - val_accuracy: 0.8000 - val_loss: -136496.0
Epoch 2/5
125/125 ━━━━━━━━━━━━━━━━━━━━ 499s 4s/step - accuracy: 0.8027 - loss: -28980.0000 - val_accuracy: 0.8000 - val_loss: -4239076
Epoch 3/5
125/125 ━━━━━━━━━━━━━━━━━━━━ 492s 4s/step - accuracy: 0.8009 - loss: -17298.0000 - val_accuracy: 0.8000 - val_loss: -1189625
Epoch 4/5
125/125 ━━━━━━━━━━━━━━━━━━━━ 492s 4s/step - accuracy: 0.8937 - loss: -15736.0000 - val_accuracy: 0.8000 - val_loss: -1623551
```

```
Epoch 5/5 125/125 ━━━━━━━━━━━━━━━━━━━━ 512s 4s/step - accuracy: 0.8098 - loss: -323728.0000 - val_accuracy: 0.8000 - val_loss: -125079
<keras.src.callbacks.history.History at 0x79624d558450>
```

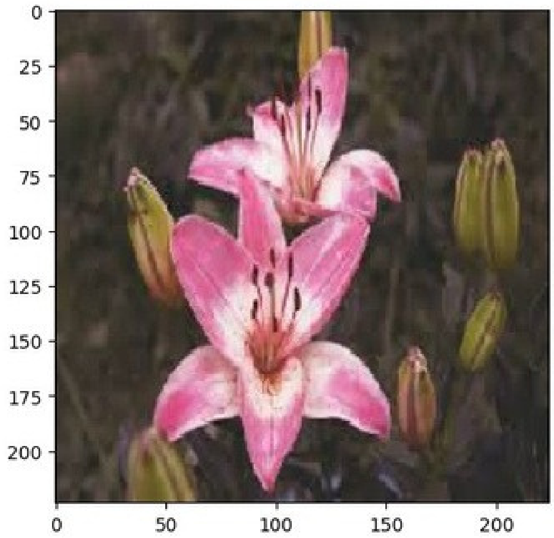
```
model.save('/content/drive/MyDrive/flower_images/tt.h5')
```

```
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is c
```

```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import  numpy as np
import matplotlib.pyplot as plt
model=load_model('/content/drive/MyDrive/flower_images/tt.h5')
print("Model Loaded Successfully")
```

```
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty until y
Model Loaded Successfully
```

```
test_image_path="/content/drive/MyDrive/flower_images/Lilly/00048a5c76.jpg"
img=image.load_img(test_image_path,target_size=(224,224))     plt.imshow(img)
plt.axis() plt.show()
```



```
img_array=image.img_to_array(img)
img_array=np.expand_dims(img_array,axis=0)
img_array=img_array/255.
```

```
prediction=model.predict(img_array)
if prediction>0.5:
print("Lilly")
else  :
  print("Rose")
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 63ms/step
Lilly
```