

```

import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.preprocessing.image import ImageDataGenerator
IMG_SIZE = 224
BATCH_SIZE = 32

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

train_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/Flower_detection', # Path to the directory
with training images
    target_size=(IMG_SIZE,IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='binary',
    subset='training'
)

val_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/Flower_detection', # Removed the extra
space at the beginning
    target_size=(IMG_SIZE,IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='binary',
    subset='validation'
)

Found 125 images belonging to 2 classes.
Found 31 images belonging to 2 classes.

#defin 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(512, activation='relu'),
    layers.Dense(1, activation='sigmoid')
])

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

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

```

```

Epoch 1/5
4/4 [=====] - 83s 19s/step - loss: 6.1148 -
accuracy: 0.5200 - val_loss: 2.0128 - val_accuracy: 0.5161
Epoch 2/5
4/4 [=====] - 23s 5s/step - loss: 1.0993 -
accuracy: 0.4400 - val_loss: 0.6655 - val_accuracy: 0.6129
Epoch 3/5
4/4 [=====] - 21s 5s/step - loss: 0.6423 -
accuracy: 0.5600 - val_loss: 0.6267 - val_accuracy: 0.7097
Epoch 4/5
4/4 [=====] - 26s 6s/step - loss: 0.5973 -
accuracy: 0.7280 - val_loss: 0.6778 - val_accuracy: 0.5806
Epoch 5/5
4/4 [=====] - 21s 5s/step - loss: 0.4914 -
accuracy: 0.7200 - val_loss: 0.5991 - val_accuracy: 0.6129

<keras.src.callbacks.History at 0x7bae1edaa320>

model.save("flower_detection.h5", "label.text")

/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('flower_detection.h5')

test_image_path =
'/content/drive/MyDrive/Flower_detection/peonise/peonies_00002.jpg' #
Store the path as a string
img = image.load_img(test_image_path, target_size=(224, 224)) # Pass
the path to load_img
img_array = image.img_to_array(img) # Convert to numpy array
img_array /= 255.0 # Normalize
img_array = np.expand_dims(img_array, axis=0) # Add batch dimension
prediction = model.predict(img_array)
print(prediction)

1/1 [=====] - 0s 115ms/step
[[0.65871716]]

if prediction < 0.5:
    print("Prediction: This is lily(Probability:", prediction[0][0])
else:

```

```
print("Prediction: This is Peonise(Probability:", prediction[0][0])
```

Prediction: This is Peonise(Probability: 0.65871716

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

Mounted at /content/drive