

```

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/Dataset_BUSI_with_GT',
    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/Dataset_BUSI_with_GT',
    target_size=(IMG_SIZE,IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    subset='validation'
)

Found 1263 images belonging to 3 classes.
Found 315 images belonging to 3 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
40/40 [=====] - 340s 9s/step - loss: 0.7329 - accuracy: 0.6144 - val_loss: 0.6652 - val_accuracy: 0.6667
Epoch 2/5
40/40 [=====] - 219s 6s/step - loss: 0.6430 - accuracy: 0.6667 - val_loss: 0.6369 - val_accuracy: 0.6667
Epoch 3/5
40/40 [=====] - 196s 5s/step - loss: 0.6368 - accuracy: 0.6667 - val_loss: 0.6366 - val_accuracy: 0.6667
Epoch 4/5
40/40 [=====] - 197s 5s/step - loss: 0.6366 - accuracy: 0.6667 - val_loss: 0.6366 - val_accuracy: 0.6667
Epoch 5/5
40/40 [=====] - 203s 5s/step - loss: 0.6366 - accuracy: 0.6667 - val_loss: 0.6366 - val_accuracy: 0.6667

<keras.src.callbacks.History at 0x7c3a434c4880>

model.save("model.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('/content/drive/MyDrive/model.h5')
test_image_path = '/content/drive/MyDrive/Dataset_BUSI_with_GT/normal/normal (1).png'
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 180ms/step
[[0.48843867]]

if prediction < 0.5:
    print("present")

```

```
else:  
    print("not present")  
present
```