

```
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/1SV21CS049 (1)/Lung X-Ray Image',
    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/1SV21CS049 (1)/Lung X-Ray Image',
    target_size=(IMG_SIZE,IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    subset='validation'
)
```

```
Found 2800 images belonging to 3 classes.
Found 700 images belonging to 3 classes.
```

```
# 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
])
```

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

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

```
Epoch 1/5
```

```
88/88 [=====] - 637s 7s/step - loss: 0.6909 -
```

```

accuracy: 0.6590 - val_loss: 0.6366 - val_accuracy: 0.6667
Epoch 2/5
88/88 [=====] - 522s 6s/step - loss: 0.6365 -
accuracy: 0.6667 - val_loss: 0.6365 - val_accuracy: 0.6667
Epoch 3/5
88/88 [=====] - 489s 6s/step - loss: 0.6365 -
accuracy: 0.6667 - val_loss: 0.6365 - val_accuracy: 0.6667
Epoch 4/5
88/88 [=====] - 471s 5s/step - loss: 0.6365 -
accuracy: 0.6667 - val_loss: 0.6365 - val_accuracy: 0.6667
Epoch 5/5
88/88 [=====] - 480s 5s/step - loss: 0.6365 -
accuracy: 0.6667 - val_loss: 0.6365 - val_accuracy: 0.6667

<keras.src.callbacks.History at 0x78476c584a00>

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

/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(

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

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

model = load_model('Model.h5')
test_image_path = '/content/drive/MyDrive/1SV21CS049 (1)/Lung X-Ray
Image/Lung_Opacity/1.jpg'
# Change target size to 244x244 to match model input
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)

img_array = img_array / 255.0

predictions = model.predict(img_array)
print(predictions)

1/1 [=====] - 0s 235ms/step
[[0.3327752]]

if predictions < 0.5:
    print('It is not Lung_opacity')

```

```
else:  
    print('It is a Lung_opacity')
```

It is not Lung\_opacity