Name: Anik Manik

Email address: iamanik4@gmail.com

Contact number: 9477672426

Anydesk address: 400 728 410

Years of Work Experience: 2.6 years

Date: 24th Jan 2021

In [1]:
```
import warnings
warnings.filterwarnings("ignore")
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import os
import datetime as dt
from datetime import datetime
from tqdm import tqdm
from glob import glob
import pandas as pd
import shutil
import glob2
from tensorflow.keras import models, layers
from tensorflow.keras.models import Model
from tensorflow.keras.layers import BatchNormalization, Activation, Flatten
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import *
from tensorflow.keras.layers import *
from tensorflow.keras.models import Model
import datetime
from sklearn.model_selection import train_test_split
from keras.losses import binary_crossentropy
import keras.backend as K
from keras.models import load_model
```

In [2]:
```
# install libraries to read dicom images
!pip install -q tensorflow-io
!pip install pydicom
```

```
     |████████████████████████████████| 25.3MB 171kB/s
Collecting pydicom
  Downloading https://files.pythonhosted.org/packages/f4/15/df16546bc59bfca390cf072d473fb2c8acd423163
6f64356593a63137e55/pydicom-2.1.2-py3-none-any.whl (1.9MB)
     |████████████████████████████████| 1.9MB 5.9MB/s
Installing collected packages: pydicom
Successfully installed pydicom-2.1.2
```

In [3]:
```
import pydicom as dicom
import tensorflow as tf
import tensorflow_io as tfio
```

In [4]:
```
# mount google drive
from google.colab import drive
drive.mount('gdrive',force_remount=True)
```

```
Mounted at gdrive
```

Download the dataset from kaggle

https://www.kaggle.com/seesee/siim-train-test (https://www.kaggle.com/seesee/siim-train-test)

In [5]:
```
# download the dataset from kaggle
# https://www.kaggle.com/seesee/siim-train-test
!wget --header="Host: storage.googleapis.com" --header="User-Agent: Mozilla/5.0 (Windows NT 10.0; Win6
4; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90 Safari/537.36" --header="Accept: te
xt/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applica
tion/signed-exchange;v=b3;q=0.9" --header="Accept-Language: en-US,en;q=0.9" --header="Referer: http
s://www.kaggle.com/" --header="Cookie: ext_name=ojplmecpdpgccookcobabopnaifgidhf" --header="Connectio
n: keep-alive" "https://storage.googleapis.com/kaggle-data-sets/245622/651264/bundle/archive.zip?X-Goo
g-Algorithm=GOOG4-RSA-SHA256&X-Goog-Credential=gcp-kaggle-com%40kaggle-161607.iam.gserviceaccount.com%
2F20210320%2Fauto%2Fstorage%2Fgoog4_request&X-Goog-Date=20210320T173401Z&X-Goog-Expires=259199&X-Goog-
SignedHeaders=host&X-Goog-Signature=076bc1656c616ecade51ffa59b801cd3675bbf5755235eb690b3273c41afd711a8
8627e0521333a5f538e727424b4f8c45f91d4a70767ea8fea30f88d03622b0338b9cf490c2d1ef69d535744e5698daaa2b01af
e213d433c214cfc72bd342e37987b34be29121dce371dd928eb0149094f459685f3944e37f3e36796730db9ab92e9799a1e1d7
f1727e54d21763ca98b863e964eaf0d1c3128817c1be94a107c59f19755b3f4479243ba474d3872e580eb643cb3a4f2b59bf32
3a06ac7873aefc2793e340c61c750b72adc69dd011b34c09aab5515847abc832ceacdd70bfc1eda0e8309b4c702745b641ec2e
6645188611b60f209897f2a266b2a2eb838b91" -c -O 'archive.zip'
```

```
--2021-03-21 02:12:49--  https://storage.googleapis.com/kaggle-data-sets/245622/651264/bundle/archiv
e.zip?X-Goog-Algorithm=GOOG4-RSA-SHA256&X-Goog-Credential=gcp-kaggle-com%40kaggle-161607.iam.gservice
account.com%2F20210320%2Fauto%2Fstorage%2Fgoog4_request&X-Goog-Date=20210320T173401Z&X-Goog-Expires=2
59199&X-Goog-SignedHeaders=host&X-Goog-Signature=076bc1656c616ecade51ffa59b801cd3675bbf5755235eb690b3
273c41afd711a88627e0521333a5f538e727424b4f8c45f91d4a70767ea8fea30f88d03622b0338b9cf490c2d1ef69d535744
e5698daaa2b01afe213d433c214cfc72bd342e37987b34be29121dce371dd928eb0149094f459685f3944e37f3e36796730db
9ab92e9799a1e1d7f1727e54d21763ca98b863e964eaf0d1c3128817c1be94a107c59f19755b3f4479243ba474d3872e580eb
643cb3a4f2b59bf323a06ac7873aefc2793e340c61c750b72adc69dd011b34c09aab5515847abc832ceacdd70bfc1eda0e830
9b4c702745b641ec2e6645188611b60f209897f2a266b2a2eb838b91
Resolving storage.googleapis.com (storage.googleapis.com)... 173.194.214.128, 173.194.216.128, 173.19
4.217.128, ...
Connecting to storage.googleapis.com (storage.googleapis.com)|173.194.214.128|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2059765561 (1.9G) [application/zip]
Saving to: ‘archive.zip’

archive.zip         100%[===================>]   1.92G   107MB/s    in 18s

2021-03-21 02:13:06 (112 MB/s) - ‘archive.zip’ saved [2059765561/2059765561]
```

In [6]:
```
# unzip the dataset
!unzip -qq 'archive.zip'
```

In [7]:
```
# read the given train csv file
image_df = pd.read_csv('siim/train-rle.csv')
image_df.head()
```

Out[7]:

	ImageId	EncodedPixels
0	1.2.276.0.7230010.3.1.4.8323329.6904.151787520...	-1
1	1.2.276.0.7230010.3.1.4.8323329.13666.15178752...	557374 2 1015 8 1009 14 1002 20 997 26 990 32 ...
2	1.2.276.0.7230010.3.1.4.8323329.11028.15178752...	-1
3	1.2.276.0.7230010.3.1.4.8323329.10366.15178752...	514175 10 1008 29 994 30 993 32 991 33 990 34 ...
4	1.2.276.0.7230010.3.1.4.8323329.10016.15178752...	592184 33 976 58 956 73 941 88 926 102 917 109...

In [8]:
```
# drop the duplicate ImageIDs
image_df.drop_duplicates(subset ="ImageId", keep = 'first', inplace = True)
```

In [9]:
```
# create a directory for dicom images
images_dicom = 'siim/images_dicom/'
if not os.path.isdir(images_dicom):
    os.makedirs(images_dicom)

# move all train dicom images from 'dicom-images-train' to 'images_dicom' in a single directory
existing_path = 'siim/dicom-images-train/'
dicom_list = glob2.glob(os.path.join(existing_path, '**/*.dcm'))
for filename in tqdm(dicom_list):
  shutil.move(str(filename), images_dicom)
```

```
100%|██████████| 12089/12089 [00:00<00:00, 17733.60it/s]
```

In [10]:
```
# remove extra space in EncodedPixels column
image_df.rename(columns = {' EncodedPixels':'EncodedPixels'}, inplace = True)

# add a column whether the image is with pneumothorax or without pneumothorax
image_df['is_pneumothorax'] = np.where(image_df['EncodedPixels']=='-1', 0, 1)

image_df.head()
```

Out[10]:

	ImageId	EncodedPixels	is_pneumothorax
0	1.2.276.0.7230010.3.1.4.8323329.6904.151787520...	-1	0
1	1.2.276.0.7230010.3.1.4.8323329.13666.15178752...	557374 2 1015 8 1009 14 1002 20 997 26 990 32 ...	1
2	1.2.276.0.7230010.3.1.4.8323329.11028.15178752...	-1	0
3	1.2.276.0.7230010.3.1.4.8323329.10366.15178752...	514175 10 1008 29 994 30 993 32 991 33 990 34 ...	1
4	1.2.276.0.7230010.3.1.4.8323329.10016.15178752...	592184 33 976 58 956 73 941 88 926 102 917 109...	1

In [11]:
```
# split the dataset and use val_df for final prediction
from sklearn.model_selection import train_test_split
train_df, val_df = train_test_split(image_df, test_size=0.2, random_state=42, stratify=image_df['is_pn
eumothorax'], shuffle=True)
```

In [12]:
```
# add full dicom path to image_df
val_df['dicom_path'] = images_dicom + val_df['ImageId']+'.dcm'
val_df.head()
```

Out[12]:

	ImageId	EncodedPixels	is_pneumothorax	
10812	1.2.276.0.7230010.3.1.4.8323329.11636.15178752...	-1	0	siim/images_dicom/1.2.27
7110	1.2.276.0.7230010.3.1.4.8323329.4471.151787518...	278724 1 1020 6 1016 9 1014 11 1011 13 1010 13...	1	siim/images_dicom/1.2.27
5130	1.2.276.0.7230010.3.1.4.8323329.5233.151787518...	-1	0	siim/images_dicom/1.2.27
5131	1.2.276.0.7230010.3.1.4.8323329.11260.15178752...	611609 30 992 33 989 36 987 40 982 44 978 49 9...	1	siim/images_dicom/1.2.27
5297	1.2.276.0.7230010.3.1.4.8323329.14511.15178752...	-1	0	siim/images_dicom/1.2.27

In [13]:
```
# Define function to convert RLE to mask, provided by organizers
def rle2mask(rle, width, height):
    mask= np.zeros(width* height)
    array = np.asarray([int(x) for x in rle.split()])
    starts = array[0::2]
    lengths = array[1::2]

    current_position = 0
    for index, start in enumerate(starts):
        current_position += start
        mask[current_position:current_position+lengths[index]] = 1
        current_position += lengths[index]

    return mask.reshape(width, height)
```

In [14]:
```
# Create Directories for mask png files
mask_png = 'siim/mask_png/'

if not os.path.isdir(mask_png):
    os.makedirs(mask_png)
```

In [15]:
```
import cv2
# define function to convert mask to png image
def masks_to_png(data, outdir):
  for img_id, enc_pix in tqdm(data.values):
    mask_path = outdir + str(img_id) + '_mask.png'
    # print(mask_path)
    if enc_pix != "-1":
      image_bytes = rle2mask(enc_pix, 1024, 1024).T
      mask = cv2.resize(image_bytes, (256, 256))
      cv2.imwrite(mask_path, mask)
    else:
      mask = np.zeros((256, 256), dtype=np.uint8)
      cv2.imwrite(mask_path, mask)
masks_to_png(val_df[['ImageId', 'EncodedPixels']], mask_png)
```

```
100%|██████████| 2410/2410 [00:05<00:00, 421.47it/s]
```

In [16]:
```
# add full png path to image_df
val_df['mask_path'] = mask_png + val_df['ImageId'] + '_mask.png'
val_df.head()
```

Out[16]:

	ImageId	EncodedPixels	is_pneumothorax	
10812	1.2.276.0.7230010.3.1.4.8323329.11636.15178752...	-1	0	siim/images_dicom/1.2.27
7110	1.2.276.0.7230010.3.1.4.8323329.4471.151787518...	278724 1 1020 6 1016 9 1014 11 1011 13 1010 13...	1	siim/images_dicom/1.2.27
5130	1.2.276.0.7230010.3.1.4.8323329.5233.151787518...	-1	0	siim/images_dicom/1.2.27
5131	1.2.276.0.7230010.3.1.4.8323329.11260.15178752...	611609 30 992 33 989 36 987 40 982 44 978 49 9...	1	siim/images_dicom/1.2.27
5297	1.2.276.0.7230010.3.1.4.8323329.14511.15178752...	-1	0	siim/images_dicom/1.2.27

01. Function-1

This function takes image path as input and returns whether the image contains pneumothorax. If yes, then the predicted segmentation.

In [59]:
```
def final_fun_1(image_path):
  # This function takes image path as input and returns whether the image contains pneumothorax.
  # If yes, then the predicted segmentation
  # preprocess the image
  size = 256
  image = tf.io.read_file(image_path)
  image = tfio.image.decode_dicom_image(image, dtype=tf.uint8,color_dim=True,scale='preserve')
  image = tf.image.convert_image_dtype(image, tf.float32)
  image =tf.squeeze(image,[0])
  image=tf.tile(image, tf.constant([1,1,3], tf.int32))
  image=tf.image.resize(image,size=[size,size])
  image = tf.expand_dims(image,axis=0)

  # define related functions
  def dice_loss(y_true, y_pred):
    smooth = 1.
    y_true_f = K.flatten(y_true)
    y_pred_f = K.flatten(y_pred)
    intersection = y_true_f * y_pred_f
    score = (2. * K.sum(intersection) + smooth) / (K.sum(y_true_f) + K.sum(y_pred_f) + smooth)
    return 1. - score

  def combined_bce_dice_loss(y_true, y_pred):
    return binary_crossentropy(y_true, y_pred) + dice_loss(y_true, y_pred)

  def iou_score(y_true, y_pred):
    smooth = 1.
    def func(y_true, y_pred):
      intersection = (y_true * y_pred).sum()
      union = y_true.sum() + y_pred.sum() - intersection
      x = (intersection + smooth) / (union + smooth)
      x = x.astype(np.float32)
      return x
    return tf.numpy_function(func, [y_true, y_pred], tf.float32)

  # load classification and segmentation model
  classification_model = load_model("gdrive/My Drive/Colab Notebooks/cs2_pneumothorax/classification/w
eights-07-0.6400.hdf5")
  segmentation_model = load_model("gdrive/My Drive/Colab Notebooks/cs2_pneumothorax/segmentation/weigh
ts-17-0.3066.hdf5",
                                  custom_objects={'combined_bce_dice_loss':combined_bce_dice_loss, "io
u_score":iou_score})

  # predict the image from the loaded models
  # first check if the image contains pneumothorax using classification model
  # predict the probability score of the image
  pred = classification_model.predict(image)
  # if the probabiliy score is greater than 0.5 then give class label=1 else 0
  if pred[0]>0.5:
    whether_pneumothorax = 1
  else:
    whether_pneumothorax = 0

  # if pneumothorax is predicted from classification model, then predict segmentation
  if whether_pneumothorax:
    print("\n\n" + "*"*20 + " THIS IMAGE CONTAINS PNEUMOTHORAX " + "*"*20 )
    # if the image contains pneumothorax, predict the mask segmentation
    pred_ms =  segmentation_model.predict(image)
    pred_mask = (pred_ms[0]>0.5).astype(np.uint8)

    plt.figure(figsize=(28,10))
    plt.subplot(131)
    plt.title("Original Image")
    plt.imshow(np.squeeze(image[0]),cmap='gray')

    plt.subplot(132)
    plt.title("Original Image with Predicted Mask")
    plt.imshow(np.squeeze(image[0]),cmap='gray',alpha=0.6)
    plt.imshow(np.squeeze(pred_mask).astype(np.uint8),cmap='Reds',alpha=0.4)
    return plt.show()
  else:
```

```
    print("\n\n" + "*"*20 + " THIS IMAGE DOES NOT CONTAIN PNEUMOTHORAX " + "*"*20 )
    plt.figure(figsize=(36,10))
    plt.subplot(131)
    plt.title("Original Image")
    plt.imshow(np.squeeze(image[0]),cmap='gray')
    return plt.show()
```

Print 1 sample image which doesn't contain pneumothorax

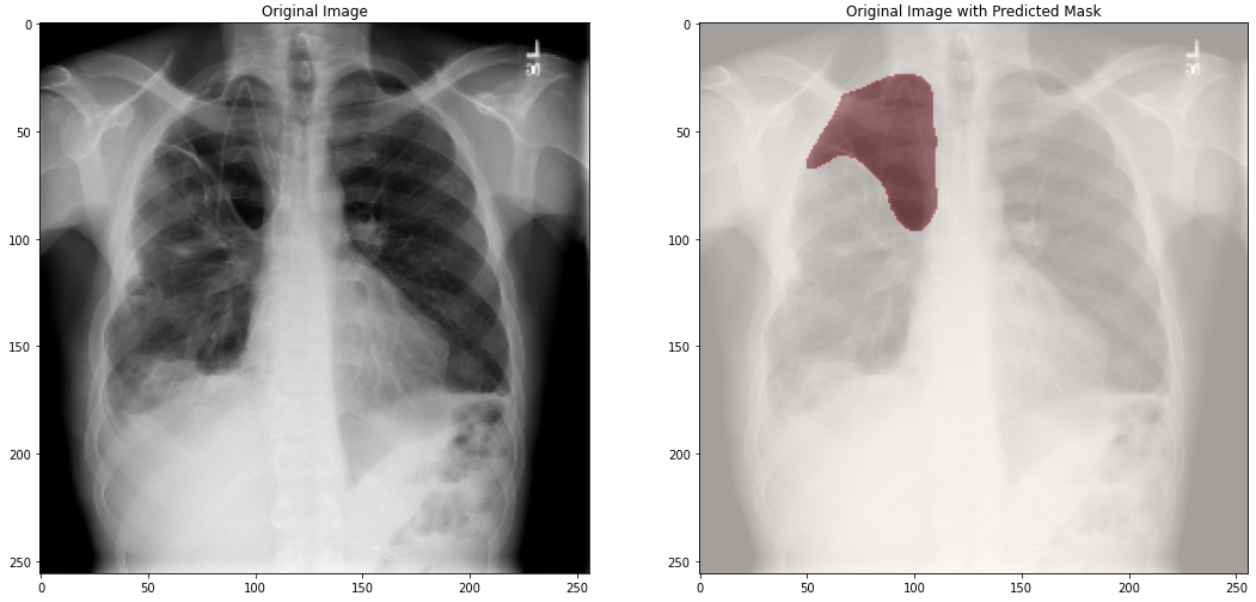
In [58]:
```
# print 1 sample image which doesn't contain pneumothorax
row_index = 48
image_path = val_df['dicom_path'].iloc[row_index]
plot_show = final_fun_1(image_path)
plot_show
```

```
WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.pred
ict_function at 0x7ff2fa3fb5f0> triggered tf.function retracing. Tracing is expensive and the excessi
ve number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing ten
sors with different shapes, (3) passing Python objects instead of tensors. For (1), please define you
r @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option t
hat relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://ww
w.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/
tf/function for  more details.


**********THIS IMAGE DOES NOT CONTAIN PNEUMOTHORAX**********
```

Print 1 sample image which contains pneumothorax

In [60]:
```
# print 1 sample image which contains pneumothorax
row_index = 45
image_path = val_df['dicom_path'].iloc[row_index]
plot_show = final_fun_1(image_path)
```

```
WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.pred
ict_function at 0x7ff2ffe780e0> triggered tf.function retracing. Tracing is expensive and the excessi
ve number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing ten
sors with different shapes, (3) passing Python objects instead of tensors. For (1), please define you
r @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option t
hat relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://ww
w.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/
tf/function for  more details.


**********THIS IMAGE CONTAINS PNEUMOTHORAX**********
WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.pred
ict_function at 0x7ff2fa5e8b00> triggered tf.function retracing. Tracing is expensive and the excessi
ve number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing ten
sors with different shapes, (3) passing Python objects instead of tensors. For (1), please define you
r @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option t
hat relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://ww
w.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/
tf/function for  more details.
```



Randomly check 10 images from the validation dataset

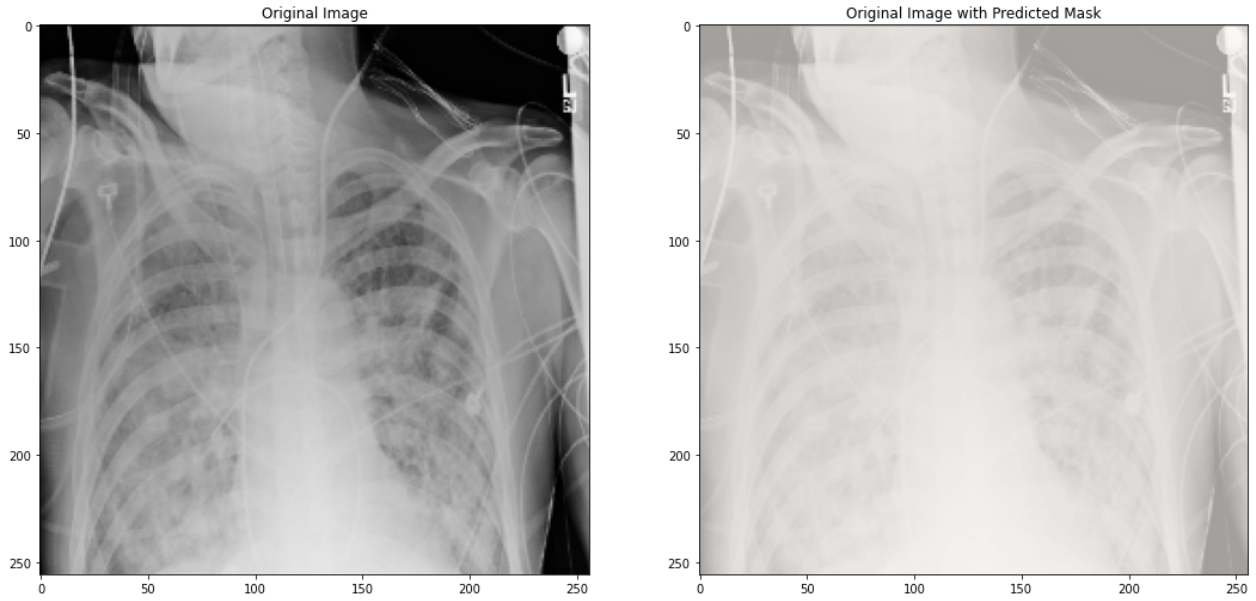
In [71]:
```
# Randomly check 10 images from the validation dataset
no_of_images = 10
for i in range(no_of_images):
  j = np.random.randint(0, len(val_df))
  image_path = val_df['dicom_path'].iloc[j]
  plot_show = final_fun_1(image_path)
  plot_show
```

WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.pred
ict_function at 0x7ff2fa77bc20> triggered tf.function retracing. Tracing is expensive and the excessi
ve number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing ten
sors with different shapes, (3) passing Python objects instead of tensors. For (1), please define you
r @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option t
hat relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://ww
w.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/
tf/function for more details.

**********THIS IMAGE DOES NOT CONTAIN PNEUMOTHORAX**********

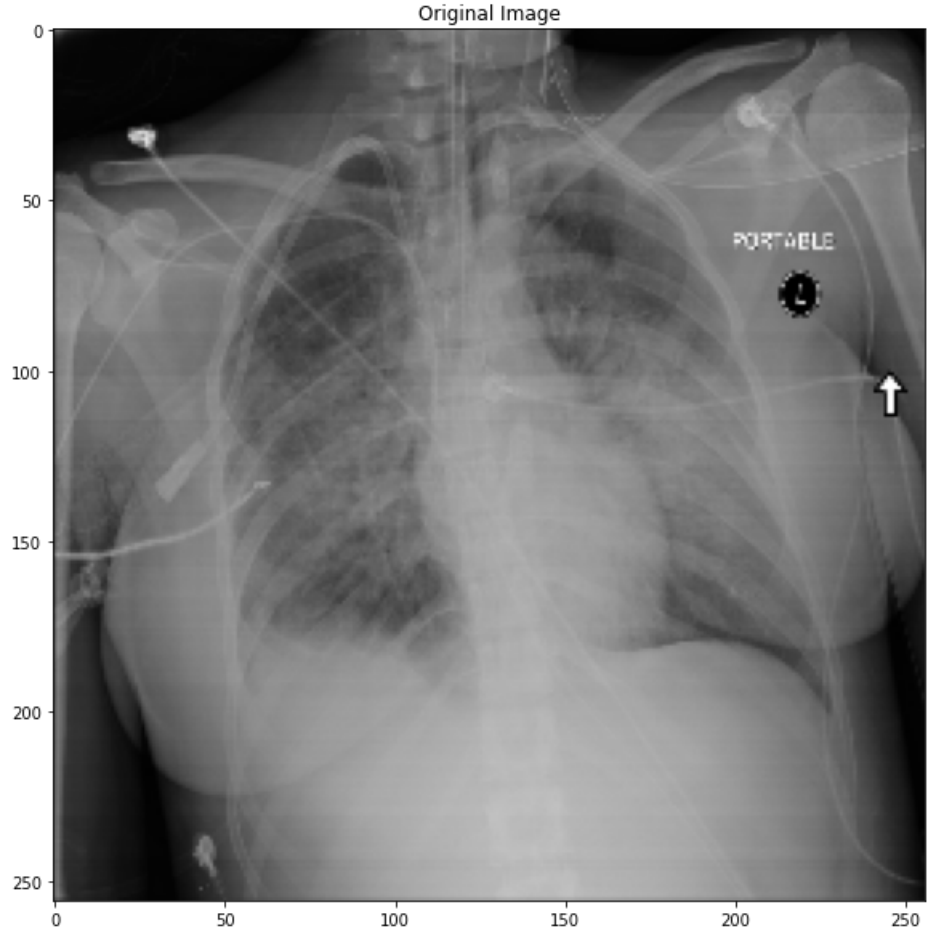
WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.pred
ict_function at 0x7ff2fa5c4f80> triggered tf.function retracing. Tracing is expensive and the excessi
ve number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing ten
sors with different shapes, (3) passing Python objects instead of tensors. For (1), please define you
r @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option t
hat relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://ww
w.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/
tf/function for more details.

**********THIS IMAGE CONTAINS PNEUMOTHORAX**********
WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.pred
ict_function at 0x7ff2fa6b4200> triggered tf.function retracing. Tracing is expensive and the excessi
ve number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing ten
sors with different shapes, (3) passing Python objects instead of tensors. For (1), please define you
r @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option t
hat relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://ww
w.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/
tf/function for more details.



WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7ff2fa64cd40> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.

**********THIS IMAGE DOES NOT CONTAIN PNEUMOTHORAX**********

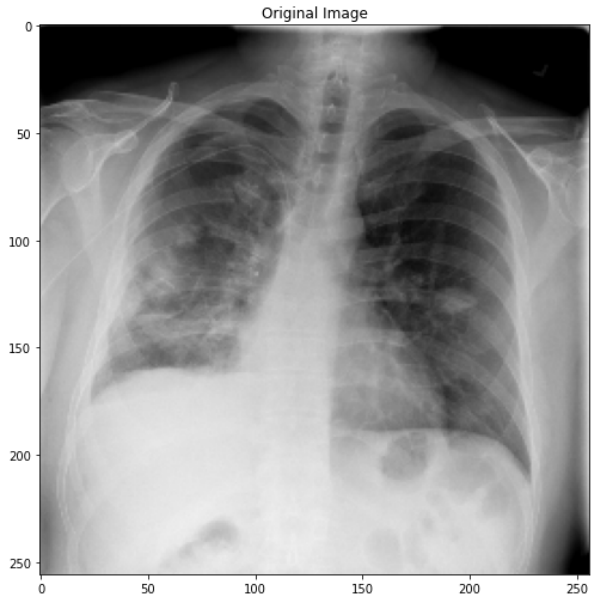


WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7ff3002740e0> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.

**********THIS IMAGE DOES NOT CONTAIN PNEUMOTHORAX**********

WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7ff2fa7f7170> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.

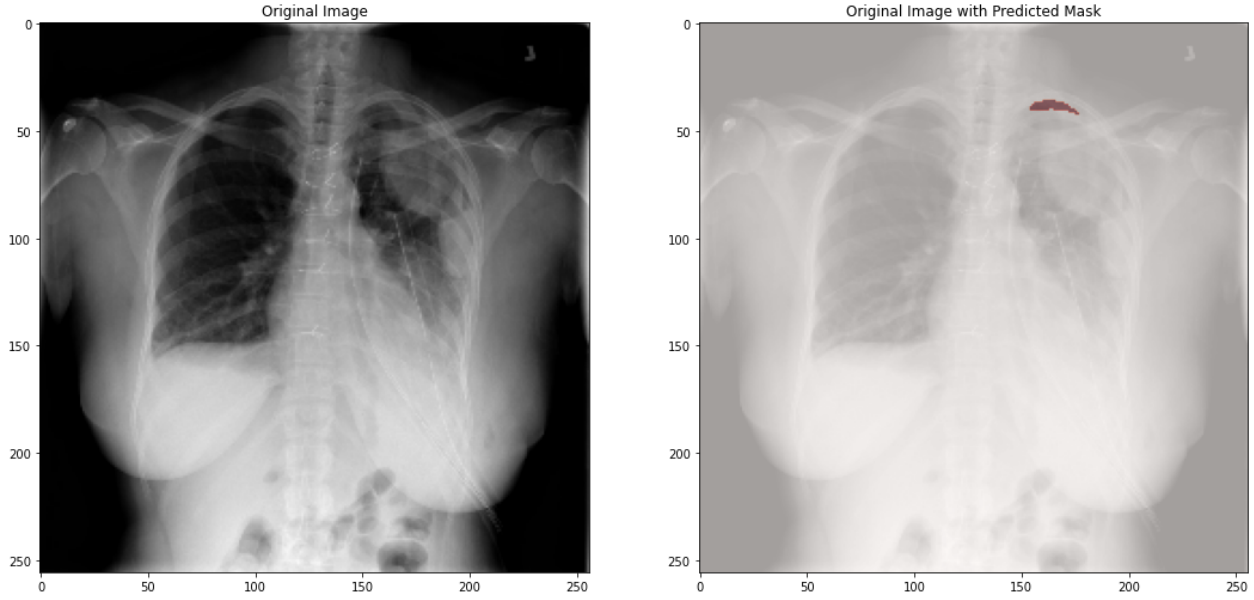
**********THIS IMAGE CONTAINS PNEUMOTHORAX**********
WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7ff2ff87f5f0> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.




```
WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.pred
ict_function at 0x7ff2ffe1b170> triggered tf.function retracing. Tracing is expensive and the excessi
ve number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing ten
sors with different shapes, (3) passing Python objects instead of tensors. For (1), please define you
r @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option t
hat relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://ww
w.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/
tf/function for  more details.


********************THIS IMAGE DOES NOT CONTAIN PNEUMOTHORAX********************
```

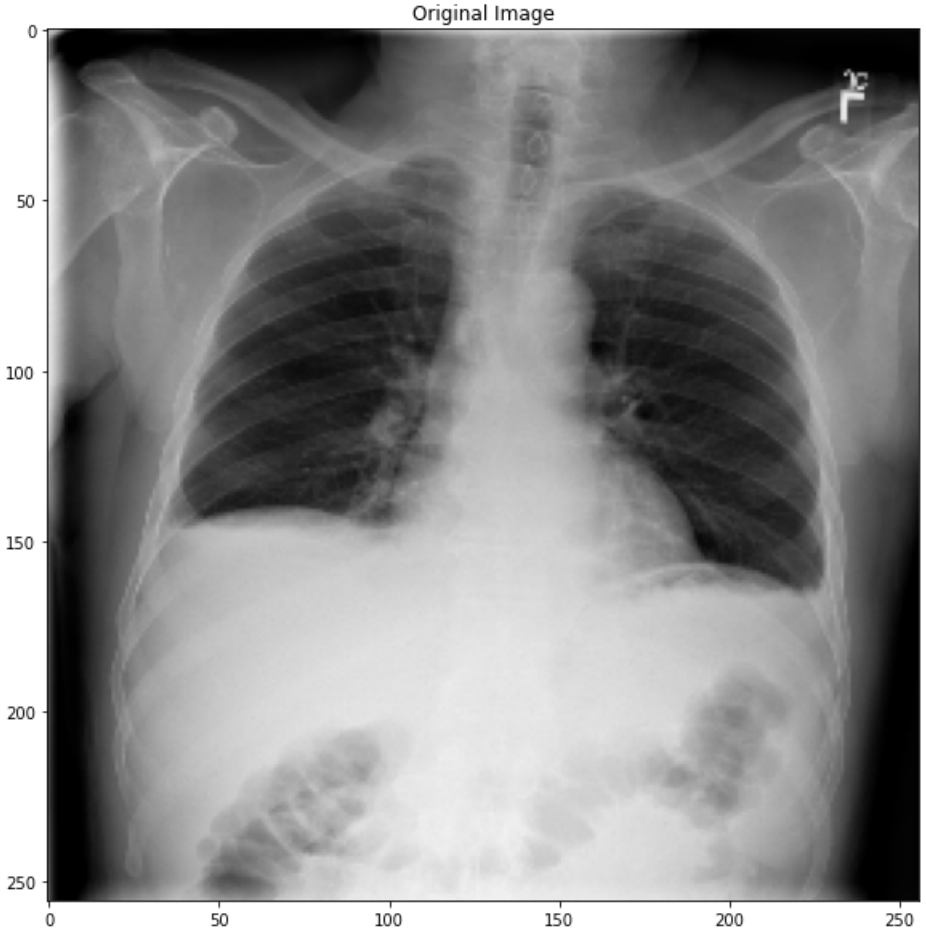
WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7ff2fc2e6ef0> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.

**********THIS IMAGE CONTAINS PNEUMOTHORAX**********
WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7ff2fa3fbc20> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.



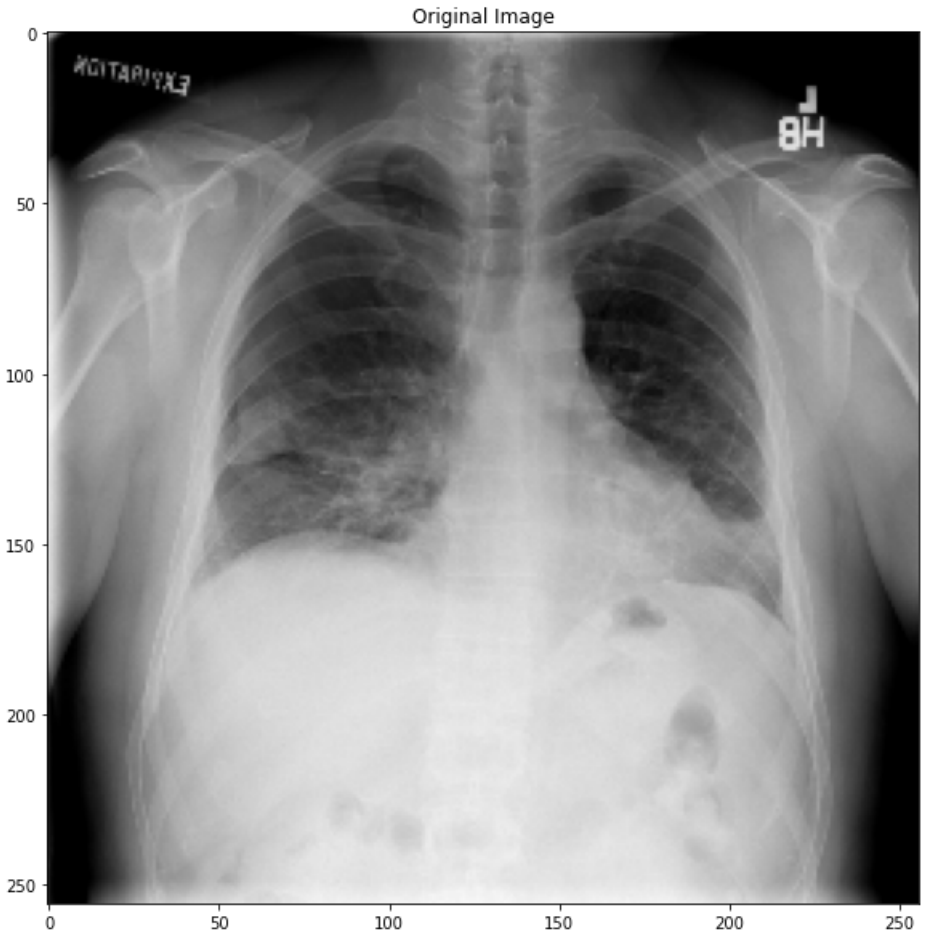
WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7ff2fc453ef0> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.

**********THIS IMAGE DOES NOT CONTAIN PNEUMOTHORAX**********



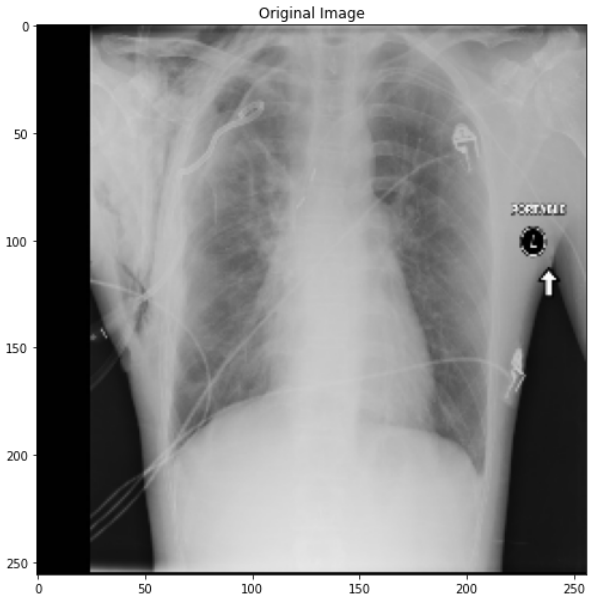
WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7ff2ffe70b00> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.

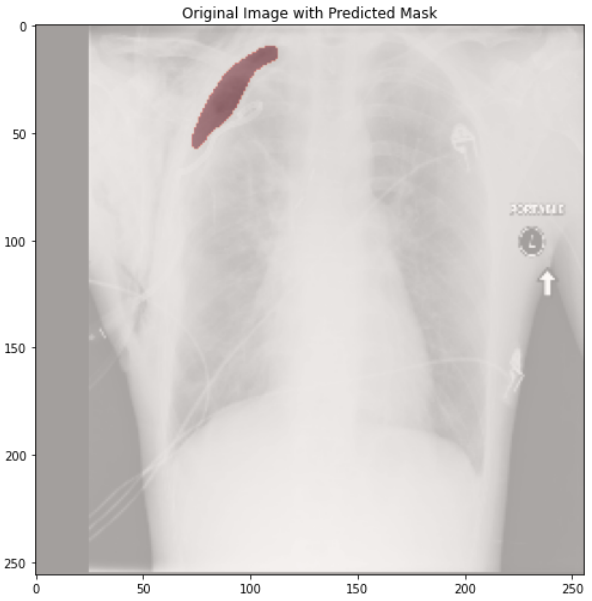
**********THIS IMAGE DOES NOT CONTAIN PNEUMOTHORAX**********



WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7ff303c29560> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.

**********THIS IMAGE CONTAINS PNEUMOTHORAX**********
WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7ff3003a3560> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.





02. Function-2

This function takes image_path and original mask_path as input.
Classification model predicts whether the image contains pneumothorax.
If yes, then segmentation model predicts the mask

In [63]:
```
def final_fun_2(image_path, mask_path):
  # This function takes image_path and original mask_path as input.
  # Classification model predicts whether the image contains pneumothorax
  # If yes, then segmentation model predicts the mask

  # preprocess the image
  size = 256
  image = tf.io.read_file(image_path)
  image = tfio.image.decode_dicom_image(image, dtype=tf.uint8,color_dim=True,scale='preserve')
  image = tf.image.convert_image_dtype(image, tf.float32)
  image =tf.squeeze(image,[0])
  image=tf.tile(image, tf.constant([1,1,3], tf.int32))
  image=tf.image.resize(image,size=[size,size])
  image = tf.expand_dims(image,axis=0)

  # preprocess mask
  mask = tf.io.read_file(mask_path)
  mask = tf.image.decode_png(mask, channels=1)
  mask = tf.image.resize(mask, [size, size])
  mask = tf.image.convert_image_dtype(mask, tf.float32)
  mask = tf.expand_dims(mask,axis=0)

  # define related functions
  def dice_loss(y_true, y_pred):
    smooth = 1.
    y_true_f = K.flatten(y_true)
    y_pred_f = K.flatten(y_pred)
    intersection = y_true_f * y_pred_f
    score = (2. * K.sum(intersection) + smooth) / (K.sum(y_true_f) + K.sum(y_pred_f) + smooth)
    return 1. - score

  def combined_bce_dice_loss(y_true, y_pred):
    return binary_crossentropy(y_true, y_pred) + dice_loss(y_true, y_pred)

  def iou_score(y_true, y_pred):
    smooth = 1.
    def func(y_true, y_pred):
      intersection = (y_true * y_pred).sum()
      union = y_true.sum() + y_pred.sum() - intersection
      x = (intersection + smooth) / (union + smooth)
      x = x.astype(np.float32)
      return x
    return tf.numpy_function(func, [y_true, y_pred], tf.float32)

  # load classification and segmentation model
  classification_model = load_model("gdrive/My Drive/Colab Notebooks/cs2_pneumothorax/classification/w
eights-07-0.6400.hdf5")
  segmentation_model = load_model("gdrive/My Drive/Colab Notebooks/cs2_pneumothorax/segmentation/weigh
ts-17-0.3066.hdf5",
                                  custom_objects={'combined_bce_dice_loss':combined_bce_dice_loss, "io
u_score":iou_score})

  # predict the image from the loaded models
  # first check if the image contains pneumothorax using classification model
  # predict the probability score of the image
  pred = classification_model.predict(image)
  # if the probabiliy score is greater than 0.5 then give class label=1 else 0
  if pred[0]>0.5:
    whether_pneumothorax = 1
  else:
    whether_pneumothorax = 0

  # If the classification model predicts the image contains pneumothorax then predict the mask
  if whether_pneumothorax:
    print("\n\n" + "*"*20 + " THIS IMAGE CONTAINS PNEUMOTHORAX " + "*"*20 )
    # if the image contains pneumothorax, predict the mask segmentation
    pred_ms =  segmentation_model.predict(image)
    pred_mask = (pred_ms[0]>0.5).astype(np.uint8)

    plt.figure(figsize=(26,8))
    plt.subplot(131)
```

```
    plt.title("Original Image")
    plt.imshow(np.squeeze(image[0]),cmap='gray')

    plt.subplot(132)
    plt.title("Original Image with Original Mask")
    plt.imshow(np.squeeze(image[0]),cmap='gray',alpha=0.6)
    plt.imshow(np.squeeze(mask[0]),cmap='Greens',alpha=0.4)

    plt.subplot(133)
    plt.title("Original Image with Predicted Mask")
    plt.imshow(np.squeeze(image[0]),cmap='gray',alpha=0.6)
    plt.imshow(np.squeeze(pred_mask).astype(np.uint8),cmap='Reds',alpha=0.4)
    return plt.show()
  else:
    print("\n\n" + "*"*20 + " THIS IMAGE DOES NOT CONTAIN PNEUMOTHORAX " + "*"*20 )
    plt.figure(figsize=(28,10))
    plt.subplot(131)
    plt.title("Original Image")
    plt.imshow(np.squeeze(image[0]),cmap='gray')

    plt.subplot(132)
    plt.title("Original Mask")
    plt.imshow(np.squeeze(mask[0]),cmap='Greens')
    return plt.show()
```

Predict 1 sample image which does't contain pneumothorax

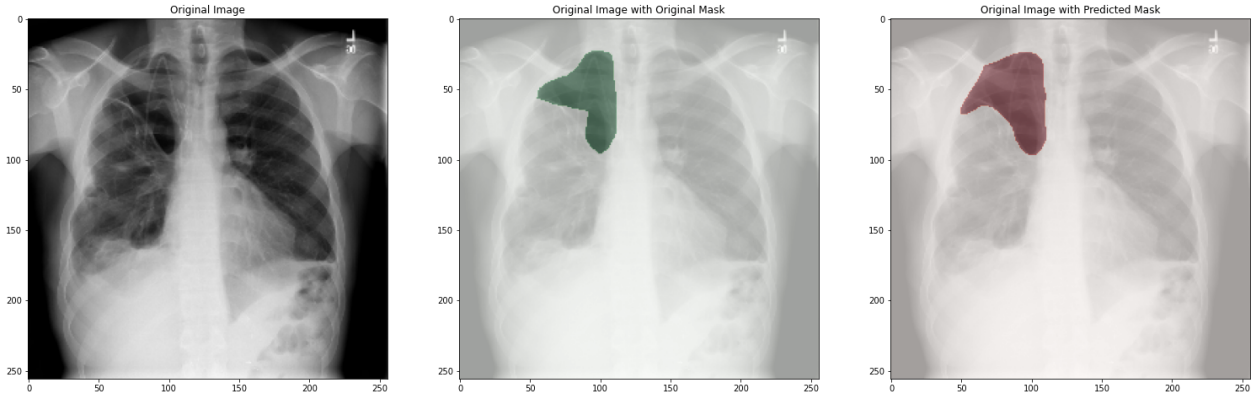
In [64]:
```
# Predict 1 sample image which does't contain pneumothorax
row_index = 48
image_path = val_df['dicom_path'].iloc[row_index]
mask_path = val_df['mask_path'].iloc[row_index]
plot_show = final_fun_2(image_path, mask_path)
plot_show
```

```
WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.pred
ict_function at 0x7ff3003a3d40> triggered tf.function retracing. Tracing is expensive and the excessi
ve number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing ten
sors with different shapes, (3) passing Python objects instead of tensors. For (1), please define you
r @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option t
hat relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://ww
w.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/
tf/function for  more details.


********** THIS IMAGE DOES NOT CONTAIN PNEUMOTHORAX **********
```

Predict 1 sample image which contains pneumothorax

In [66]:
```
# Predict 1 sample image which contains pneumothorax
row_index = 45
image_path = val_df['dicom_path'].iloc[row_index]
mask_path = val_df['mask_path'].iloc[row_index]
plot_show = final_fun_2(image_path, mask_path)
plot_show
```

```
WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.pred
ict_function at 0x7ff2fa579680> triggered tf.function retracing. Tracing is expensive and the excessi
ve number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing ten
sors with different shapes, (3) passing Python objects instead of tensors. For (1), please define you
r @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option t
hat relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://ww
w.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/
tf/function for  more details.


******************** THIS IMAGE CONTAINS PNEUMOTHORAX ********************
WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.pred
ict_function at 0x7ff300274d40> triggered tf.function retracing. Tracing is expensive and the excessi
ve number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing ten
sors with different shapes, (3) passing Python objects instead of tensors. For (1), please define you
r @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option t
hat relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://ww
w.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/
tf/function for  more details.
```

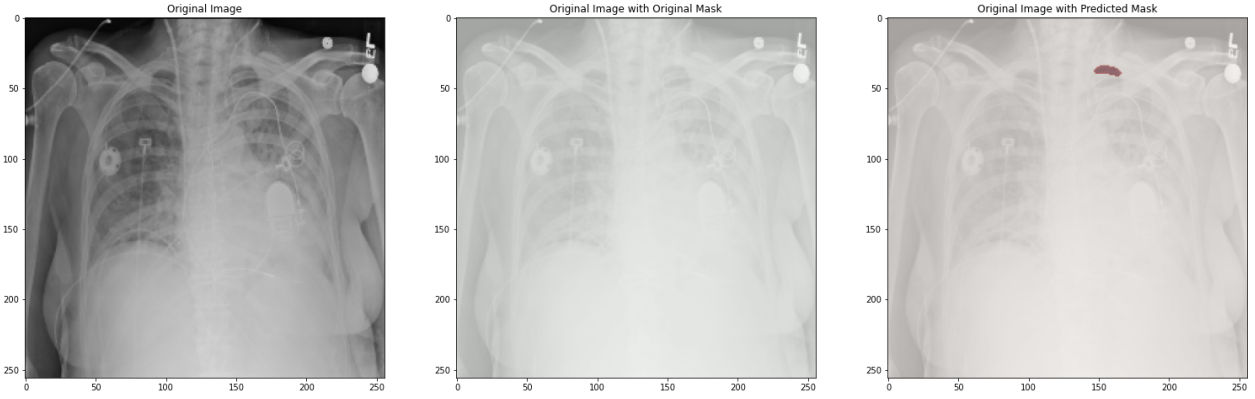


Randomly predict 10 images from the validation dataframe

In [73]:
```
# Randomly predict 10 images from the validation dataframe
no_of_images = 10
for i in range(no_of_images):
  j = np.random.randint(0, len(val_df))
  image_path = val_df['dicom_path'].iloc[j]
  mask_path = val_df['mask_path'].iloc[j]
  plot_show = final_fun_2(image_path, mask_path)
  plot_show
```

WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7ff2fa594a70> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.

********** THIS IMAGE CONTAINS PNEUMOTHORAX **********
WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7ff2fa6b4200> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.



WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7ff2ff87f440> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.

********** THIS IMAGE CONTAINS PNEUMOTHORAX **********
WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7ff300274680> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.

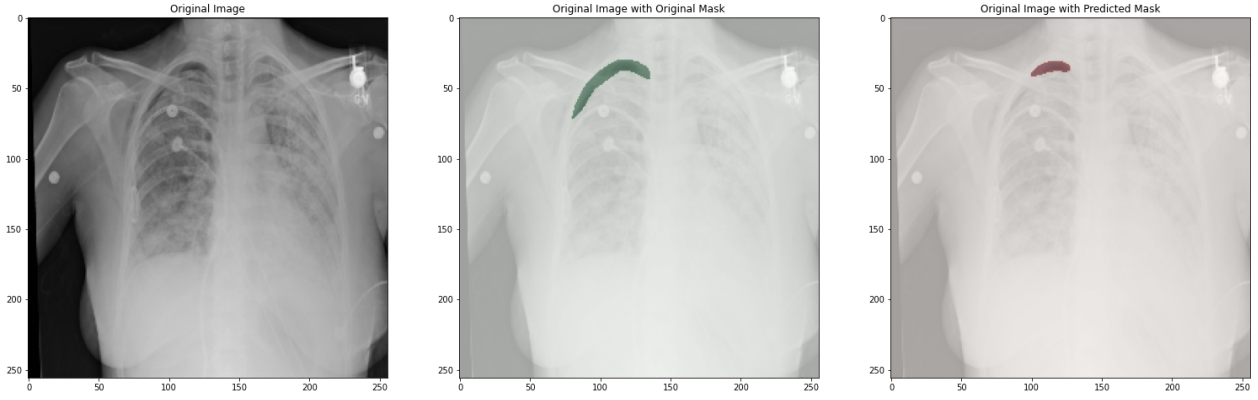
WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7ff2fa7f78c0> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.

********** THIS IMAGE DOES NOT CONTAIN PNEUMOTHORAX **********

WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7ff300275dd0> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.

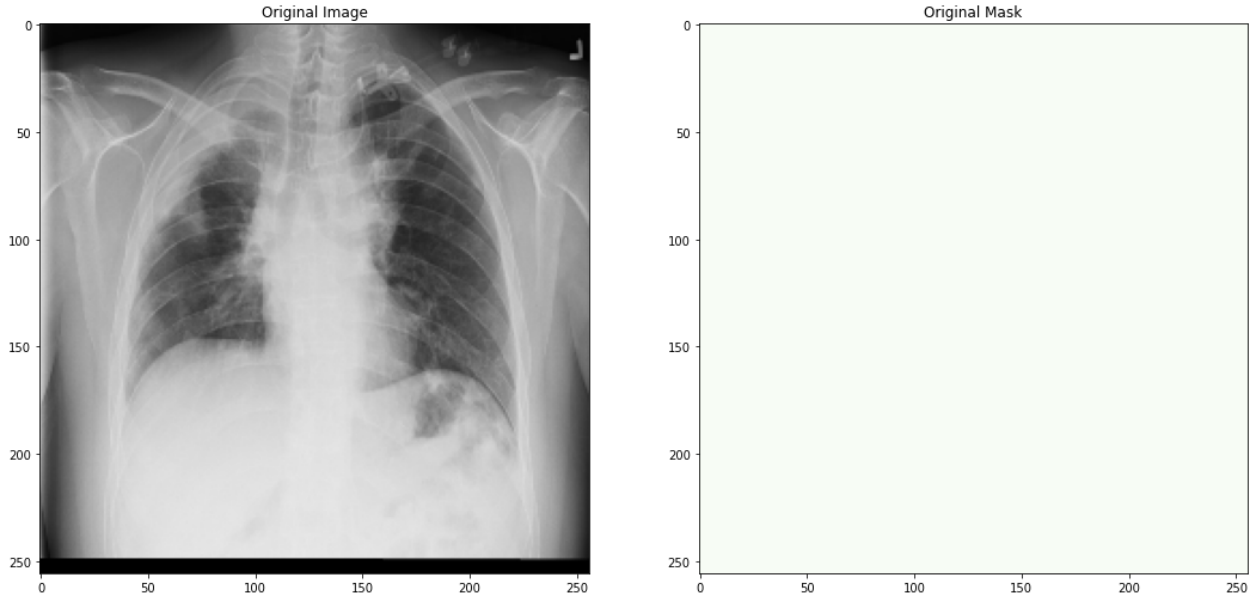
********** THIS IMAGE CONTAINS PNEUMOTHORAX **********

WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7ff3002744d0> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.



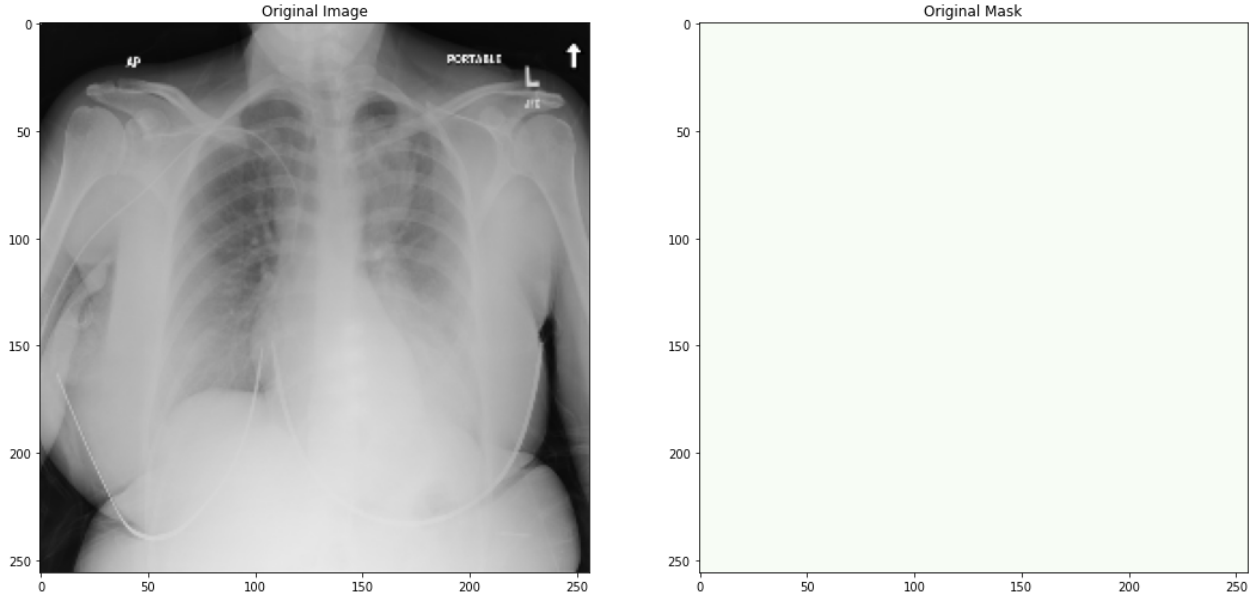
WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7ff2fa5798c0> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.

********** THIS IMAGE DOES NOT CONTAIN PNEUMOTHORAX **********



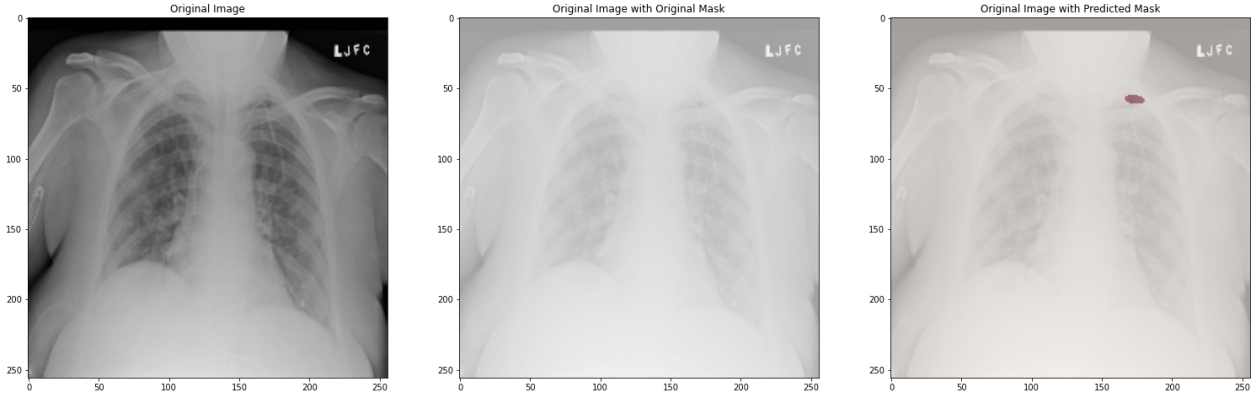
WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7ff2ffde1830> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.

********** THIS IMAGE DOES NOT CONTAIN PNEUMOTHORAX **********



WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7ff3003a3ef0> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.

********** THIS IMAGE CONTAINS PNEUMOTHORAX **********
WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7ff2fc420cb0> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.

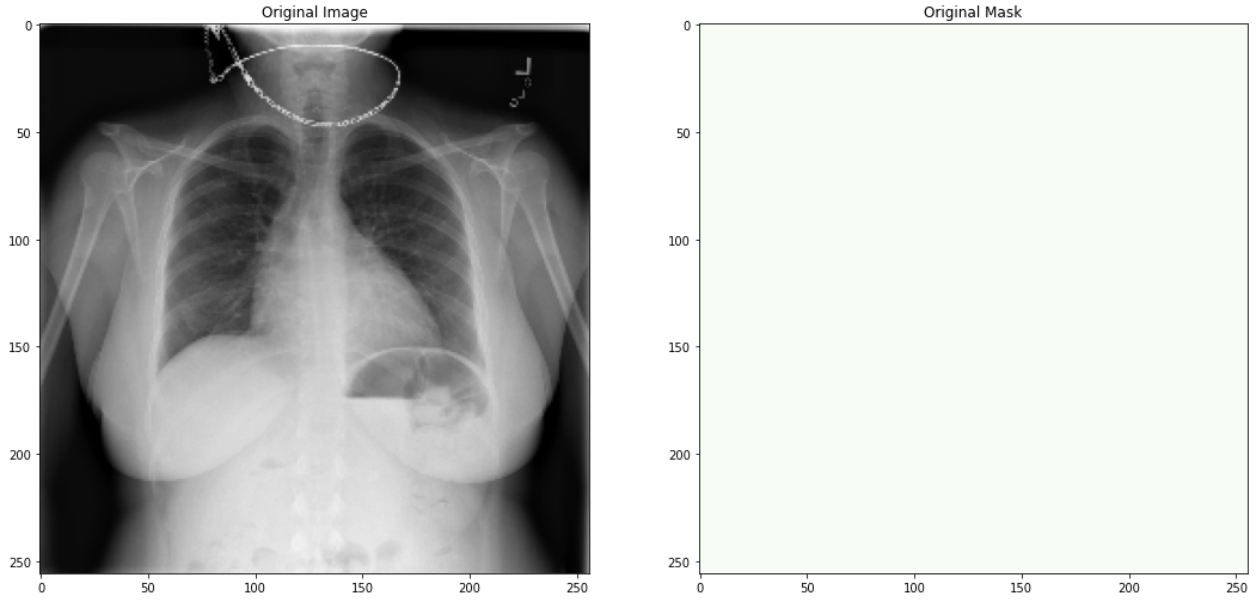


WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7ff2fa5c4710> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.

********** THIS IMAGE DOES NOT CONTAIN PNEUMOTHORAX **********

WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7ff2fd425cb0> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.

********** THIS IMAGE DOES NOT CONTAIN PNEUMOTHORAX **********



WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7ff2fa94bcb0> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.

********** THIS IMAGE DOES NOT CONTAIN PNEUMOTHORAX **********