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.notebook 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 128kB/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 10.9MB/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%
2F20210324%2Fauto%2Fstorage%2Fgoog4_request&X-Goog-Date=20210324T022759Z&X-Goog-Expires=259199&X-Goog-
SignedHeaders=host&X-Goog-Signature=6537e07b49380396cf2a8773c646d3e4847a77f3f9e6d24612c369ee3962e3aaab
5e69f6e9ea89f09026dea49c0ea2818d9a29f5e713e0b25cba7445cbfe806668b81034ec3b93f88942ec5770e0e69c7c2387a4
fcc6ea770aa548f4e84d1e7f7d789e8581e5a78883165555fc729dbfeeeca80c797157680c411dd8e045b95a5eb7b304d91f89
f4e56a9bc25d46f84a416d540b4aef097d7ac0512bcc6ca52495e135a86065aaec9e9fe7f0188a29d89f1c11775b84f8d64d8b
cb3a8641feb1f2e7473c02a91402da8df9784bd889855e0c274a65098a5abcccb5cc0f926f02ed52330b438bc2a538c77d0fb9
492927c1ec7296b0f9828950b2ffe6f6a12e76" -c -O 'archive.zip'
```

```
--2021-03-25 16:28:11--  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%2F20210324%2Fauto%2Fstorage%2Fgoog4_request&X-Goog-Date=20210324T022759Z&X-Goog-Expires=2
59199&X-Goog-SignedHeaders=host&X-Goog-Signature=6537e07b49380396cf2a8773c646d3e4847a77f3f9e6d24612c3
69ee3962e3aaab5e69f6e9ea89f09026dea49c0ea2818d9a29f5e713e0b25cba7445cbfe806668b81034ec3b93f88942ec577
0e0e69c7c2387a4fcc6ea770aa548f4e84d1e7f7d789e8581e5a78883165555fc729dbfeeeca80c797157680c411dd8e045b9
5a5eb7b304d91f89f4e56a9bc25d46f84a416d540b4aef097d7ac0512bcc6ca52495e135a86065aaec9e9fe7f0188a29d89f1
c11775b84f8d64d8bcb3a8641feb1f2e7473c02a91402da8df9784bd889855e0c274a65098a5abcccb5cc0f926f02ed52330b
438bc2a538c77d0fb9492927c1ec7296b0f9828950b2ffe6f6a12e76
Resolving storage.googleapis.com (storage.googleapis.com)... 64.233.166.128, 74.125.133.128, 74.125.1
40.128, ...
Connecting to storage.googleapis.com (storage.googleapis.com)|64.233.166.128|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2059765561 (1.9G) [application/zip]
Saving to: ‘archive.zip’

archive.zip         100%[===================>]   1.92G  14.6MB/s    in 3m 23s

2021-03-25 16:31:35 (9.66 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)
```

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]:

```
image_df = image_df.loc[image_df['is_pneumothorax'] == 1]
image_df.head()
```

Out[11]:

	ImageId	EncodedPixels	is_pneumothorax
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
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
10	1.2.276.0.7230010.3.1.4.8323329.3514.151787517...	759441 11 1010 15 1007 18 1005 19 1005 20 1003...	1
13	1.2.276.0.7230010.3.1.4.8323329.14008.15178752...	119368 98 923 102 908 118 903 126 896 133 889 ...	1

In [12]:

```
# 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, shuffle=True)
```

In [13]:

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

Out[13]:

	ImageId	EncodedPixels	is_pneumothorax	
1106	1.2.276.0.7230010.3.1.4.8323329.4703.151787518...	609496 23 991 48 971 83 936 103 918 111 910 11...	1	siim/images_dicom/1.2.27
7989	1.2.276.0.7230010.3.1.4.8323329.1033.151787516...	284490 38 976 58 956 73 941 87 877 149 845 182...	1	siim/images_dicom/1.2.27
8108	1.2.276.0.7230010.3.1.4.8323329.12294.15178752...	233827 23 994 37 981 47 971 57 960 66 955 70 9...	1	siim/images_dicom/1.2.27
11527	1.2.276.0.7230010.3.1.4.8323329.4628.151787518...	190212 2 1015 7 1009 13 992 12 1 18 980 19 100...	1	siim/images_dicom/1.2.27
3796	1.2.276.0.7230010.3.1.4.8323329.11498.15178752...	674944 3 1017 8 1014 10 1012 13 1008 16 1005 2...	1	siim/images_dicom/1.2.27

In [14]:

```
# 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 [15]:

```
# Create Directories for mask png files
mask_png = 'siim/mask_png/'

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

In [16]:

```python
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)
```

In [17]:

```python
# add full png path to image_df
val_df['mask_path'] = mask_png + val_df['ImageId'] + '_mask.png'
val_df = val_df.head(20)
val_df.head()
```

Out[17]:

	ImageId	EncodedPixels	is_pneumothorax	
1106	1.2.276.0.7230010.3.1.4.8323329.4703.151787518...	609496 23 991 48 971 83 936 103 918 111 910 11...	1	siim/images_dicom/1.2.27
7989	1.2.276.0.7230010.3.1.4.8323329.1033.151787516...	284490 38 976 58 956 73 941 87 877 149 845 182...	1	siim/images_dicom/1.2.27
8108	1.2.276.0.7230010.3.1.4.8323329.12294.15178752...	233827 23 994 37 981 47 971 57 960 66 955 70 9...	1	siim/images_dicom/1.2.27
11527	1.2.276.0.7230010.3.1.4.8323329.4628.151787518...	190212 2 1015 7 1009 13 992 12 1 18 980 19 100...	1	siim/images_dicom/1.2.27
3796	1.2.276.0.7230010.3.1.4.8323329.11498.15178752...	674944 3 1017 8 1014 10 1012 13 1008 16 1005 2...	1	siim/images_dicom/1.2.27

In [19]:

```
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 f(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(f, [y_true, y_pred], tf.float32)
```

In [20]:

```
# Load best weights to segmentation model
from keras.models import load_model
# custom_objects = optional dictionary mapping names (strings) to custom classes or functions to be co
nsidered during deserialization
model_seg = load_model("gdrive/My Drive/Colab Notebooks/cs2_pneumothorax/segmentation/weights-17-0.306
6.hdf5",
                                 custom_objects={'combined_bce_dice_loss':combined_bce_dice_loss, "iou
_score":iou_score})
```

Convert hdf5 file to tflite version

In [22]:

```
# Convert hdf5 file to tflite version
converter = tf.lite.TFLiteConverter.from_keras_model(model_seg)
tflite_model = converter.convert()
open("gdrive/My Drive/Colab Notebooks/cs2_pneumothorax/segmentation/converted_seg_model.tflite", "wb")
.write(tflite_model)
```

```
INFO:tensorflow:Assets written to: /tmp/tmpebwlcx_h/assets
```

Out[22]:

```
48311532
```

Convert hdf5 to quantized tflite version

In [23]:

```
# Convert the model to quantized version with post-training quantization
converter = tf.lite.TFLiteConverter.from_keras_model(model_seg)
converter.optimizations = [tf.lite.Optimize.OPTIMIZE_FOR_SIZE]
tflite_quant_model = converter.convert()
open("gdrive/My Drive/Colab Notebooks/cs2_pneumothorax/segmentation/converted_seg_quant_model.tflite",
"wb").write(tflite_quant_model)
```

```
INFO:tensorflow:Assets written to: /tmp/tmpz7bju42z/assets

INFO:tensorflow:Assets written to: /tmp/tmpz7bju42z/assets
```

Out[23]:

```
12626992
```

Load tflite model

In [24]:
```
# https://colab.research.google.com/github/frogermcs/TFLite-Tester/blob/master/notebooks/Testing_TFLite_model.ipynb#scrollTo=OoBmFmXlHVhj
tflite_interpreter = tf.lite.Interpreter(model_path="gdrive/My Drive/Colab Notebooks/cs2_pneumothorax/segmentation/converted_seg_model.tflite")

# Learn about its input and output details
input_details = tflite_interpreter.get_input_details()
output_details = tflite_interpreter.get_output_details()
tflite_interpreter.allocate_tensors()
```

Load quantized tflite model

In [25]:
```
# Load quantized TFLite model
tflite_interpreter_quant = tf.lite.Interpreter(model_path="gdrive/My Drive/Colab Notebooks/cs2_pneumothorax/segmentation/converted_seg_quant_model.tflite")

# Learn about its input and output details
input_details = tflite_interpreter_quant.get_input_details()
output_details = tflite_interpreter_quant.get_output_details()
tflite_interpreter_quant.allocate_tensors()
```

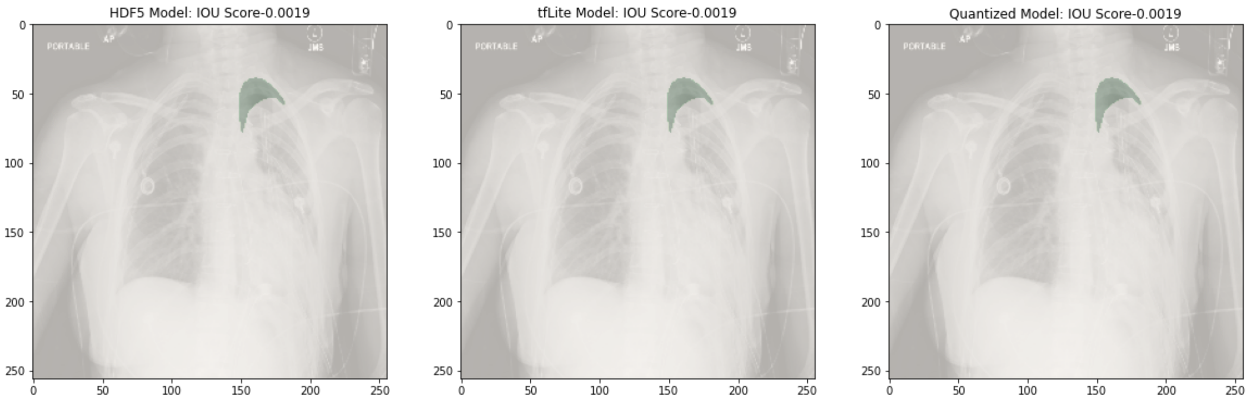
Define iou_score for a single image mask

In [26]:
```
# define iou_score for a single image mask
def iou_score_single_image(pred_mask, mask):
  intersection = np.sum(np.logical_and(pred_mask, mask))
  union = np.sum(np.logical_or(pred_mask, mask))
  smooth = 1.
  iou = (intersection + smooth) / (union + smooth)
  iou = np.mean(iou)
  return iou
```

In [39]:

```
# print 20 images with best iou score
iou_score_hdf5 = []
iou_score_tflite = []
iou_score_quantized = []
for row_no in tqdm(range(20)):
  img_path = val_df['dicom_path'].iloc[row_no]
  msk_path = val_df['mask_path'].iloc[row_no]
  print("ImageID = " + str(img_path))

  size = 256
  image = tf.io.read_file(img_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)

  mask = tf.io.read_file(msk_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)

  # predict mask using unet_imagenet_model
  pred_hdf5 =  model_seg.predict(image)
  pred_mask_hdf5 = (pred_hdf5[0]>0.5).astype(np.uint8)
  iou_hdf5 = iou_score_single_image(pred_mask_hdf5, mask)
  iou_score_hdf5.append(iou_hdf5)

  # plot for hdf5 model
  plt.figure(figsize=(20,6))
  plt.subplot(131)
  plt.title("HDF5 Model: IOU Score-{:.4f}".format(iou_hdf5))
  plt.imshow(np.squeeze(image[0]),cmap='gray',alpha=0.8)
  plt.imshow(np.squeeze(mask[0]),cmap='Greens',alpha=0.4)
  plt.imshow(np.squeeze(pred_mask_hdf5).astype(np.uint8),cmap='Reds',alpha=0.4)
  # plt.show()

  # predict using tflite model
  tflite_interpreter.set_tensor(input_details[0]['index'], image)
  tflite_interpreter.invoke()
  tflite_model_predictions = tflite_interpreter.get_tensor(output_details[0]['index'])
  pred_mask_tflite = (tflite_model_predictions[0]>0.5).astype(np.uint8)
  pred_mask_tflite = tf.convert_to_tensor(pred_mask_tflite)
  iou_tflite = iou_score_single_image(pred_mask_tflite, mask)
  iou_score_tflite.append(iou_tflite)

  # plot for tflite model
  plt.subplot(132)
  plt.title("tfLite Model: IOU Score-{:.4f}".format(iou_tflite))
  plt.imshow(np.squeeze(image[0]),cmap='gray',alpha=0.8)
  plt.imshow(np.squeeze(mask[0]),cmap='Greens',alpha=0.4)
  plt.imshow(np.squeeze(pred_mask_tflite).astype(np.uint8),cmap='Reds',alpha=0.4)
  # plt.show()

  # predict using tflite quantized model
  tflite_interpreter_quant.set_tensor(input_details[0]['index'], image)
  tflite_interpreter_quant.invoke()
  quantized_model_predictions = tflite_interpreter_quant.get_tensor(output_details[0]['index'])
  pred_mask_quantized = (quantized_model_predictions[0]>0.5).astype(np.uint8)
  pred_mask_quantized = tf.convert_to_tensor(pred_mask_quantized)
  iou_quantized = iou_score_single_image(pred_mask_quantized, mask)
  iou_score_quantized.append(iou_quantized)

  # plot for tflite quantized model
  plt.subplot(133)
  plt.title("Quantized Model: IOU Score-{:.4f}".format(iou_quantized))
  plt.imshow(np.squeeze(image[0]),cmap='gray',alpha=0.8)
  plt.imshow(np.squeeze(mask[0]),cmap='Greens',alpha=0.4)
  plt.imshow(np.squeeze(pred_mask_quantized).astype(np.uint8),cmap='Reds',alpha=0.4)
plt.show()
print("\n\n")
```

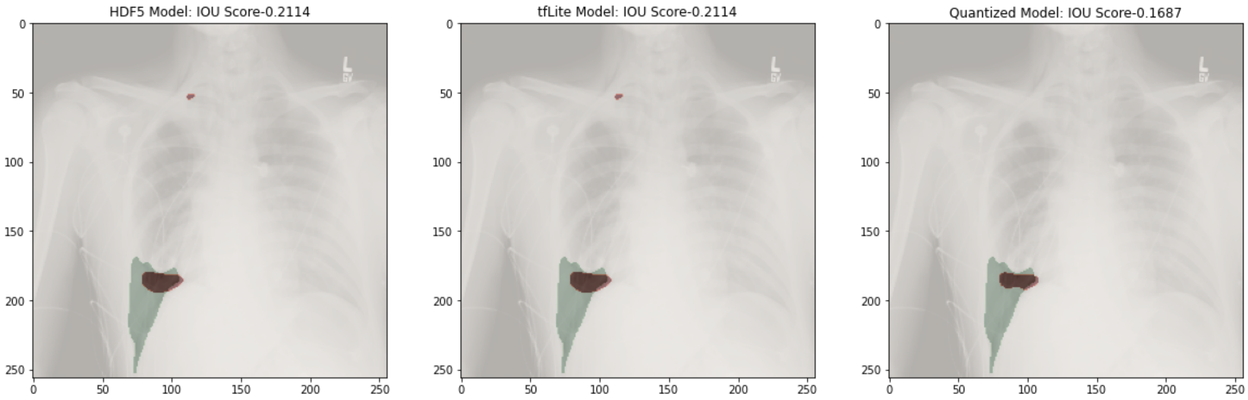
ImageID = siim/images_dicom/1.2.276.0.7230010.3.1.4.8323329.4703.1517875184.225849.dcm


HDF5 Model: IOU Score-0.0019

tfLite Model: IOU Score-0.0019

Quantized Model: IOU Score-0.0019

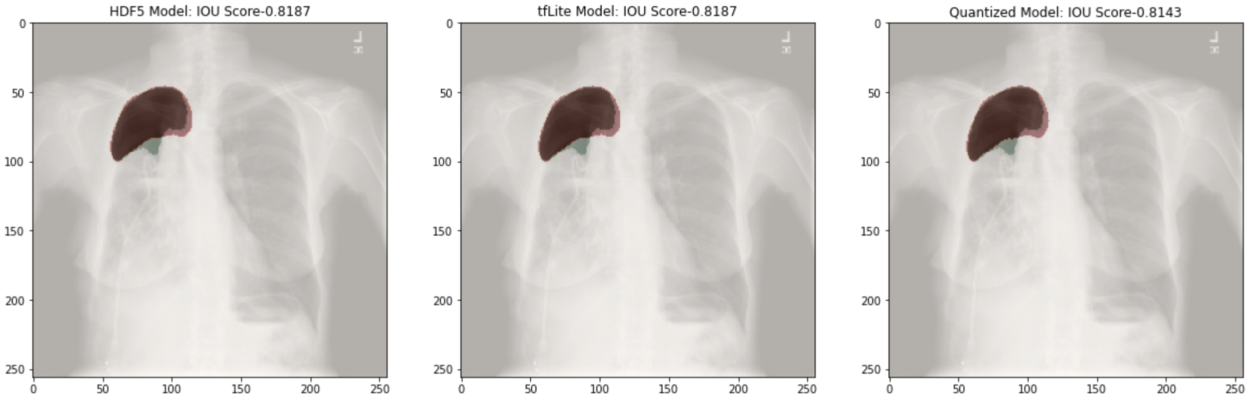
ImageID = siim/images_dicom/1.2.276.0.7230010.3.1.4.8323329.1033.1517875166.9366.dcm


HDF5 Model: IOU Score-0.2114

tfLite Model: IOU Score-0.2114

Quantized Model: IOU Score-0.1687

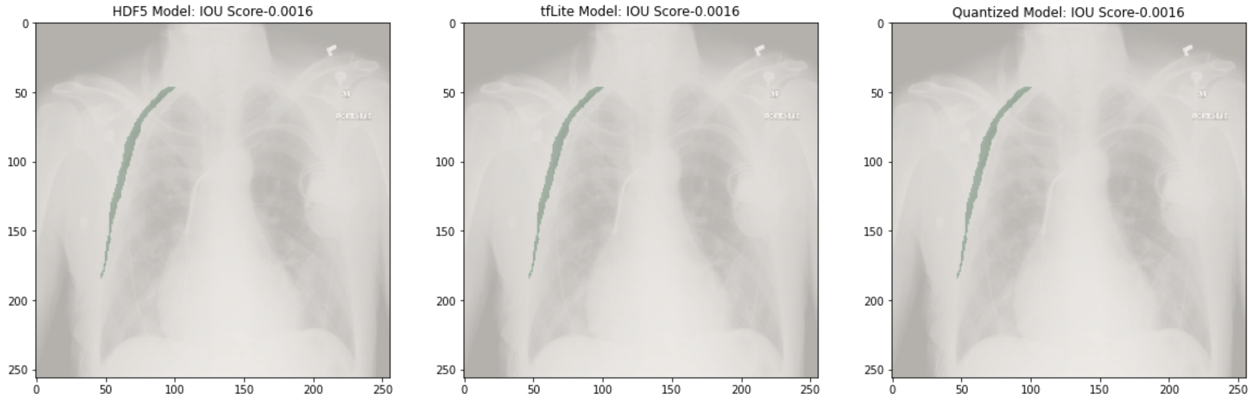
ImageID = siim/images_dicom/1.2.276.0.7230010.3.1.4.8323329.12294.1517875238.418455.dcm


HDF5 Model: IOU Score-0.8187

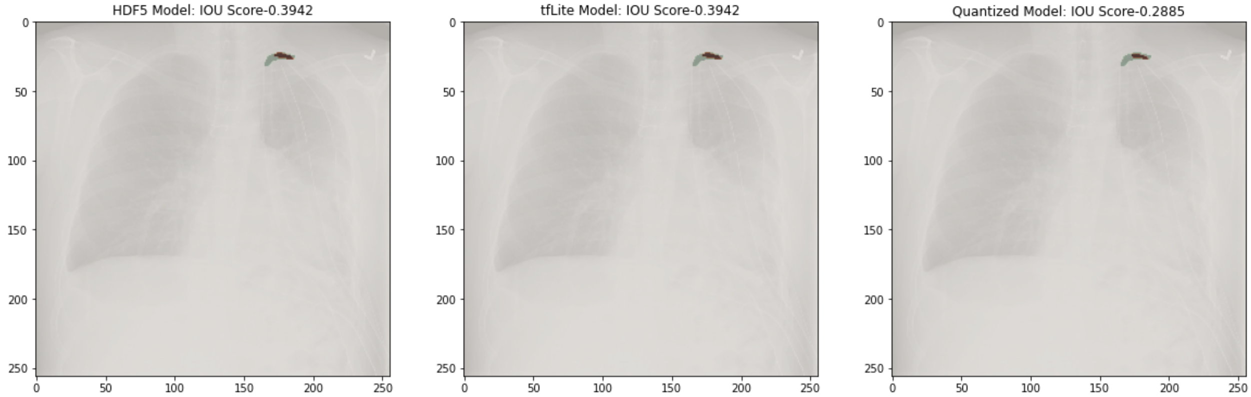
tfLite Model: IOU Score-0.8187

Quantized Model: IOU Score-0.8143

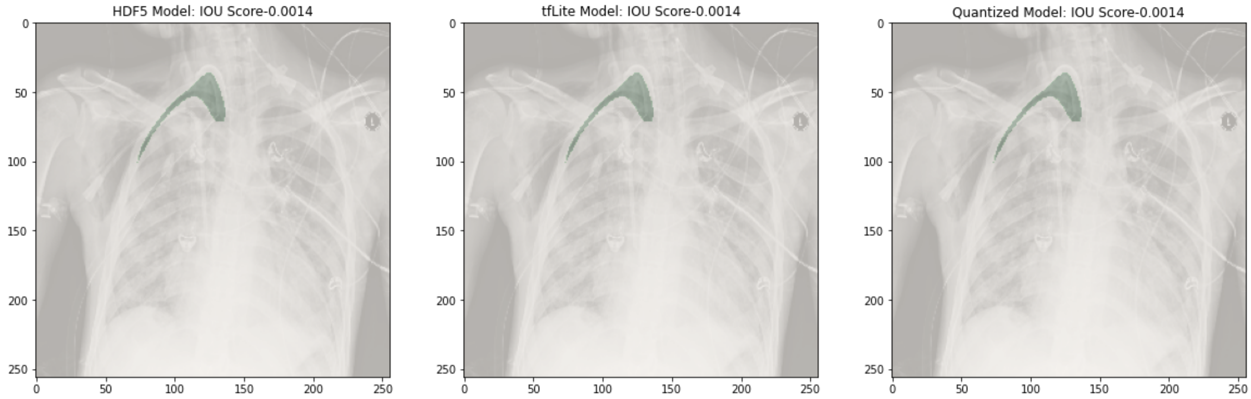
ImageID = siim/images_dicom/1.2.276.0.7230010.3.1.4.8323329.4628.1517875183.816936.dcm


HDF5 Model: IOU Score-0.0016
tfLite Model: IOU Score-0.0016
Quantized Model: IOU Score-0.0016
0
50
100
150
200
250

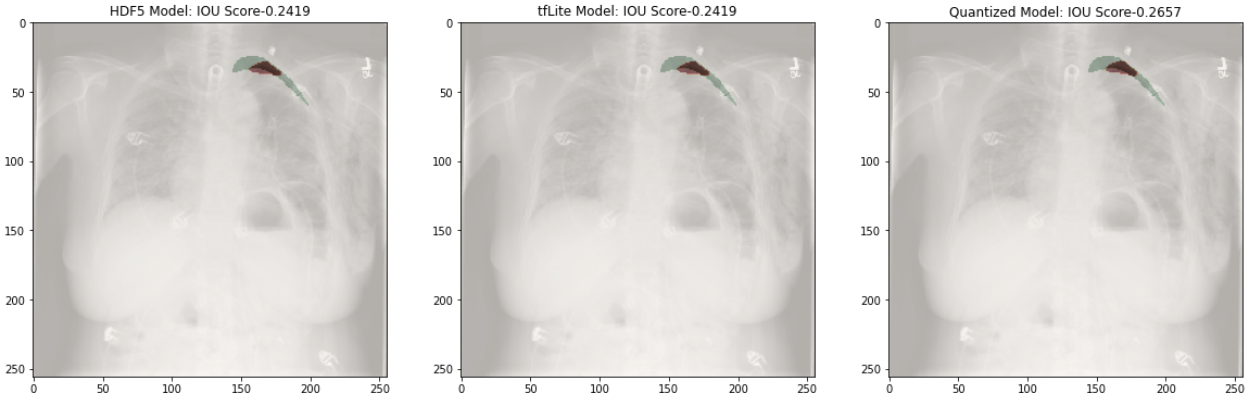
```
ImageID = siim/images_dicom/1.2.276.0.7230010.3.1.4.8323329.11498.1517875233.226620.dcm
```


HDF5 Model: IOU Score-0.3942
tfLite Model: IOU Score-0.3942
Quantized Model: IOU Score-0.2885
0
50
100
150
200
250

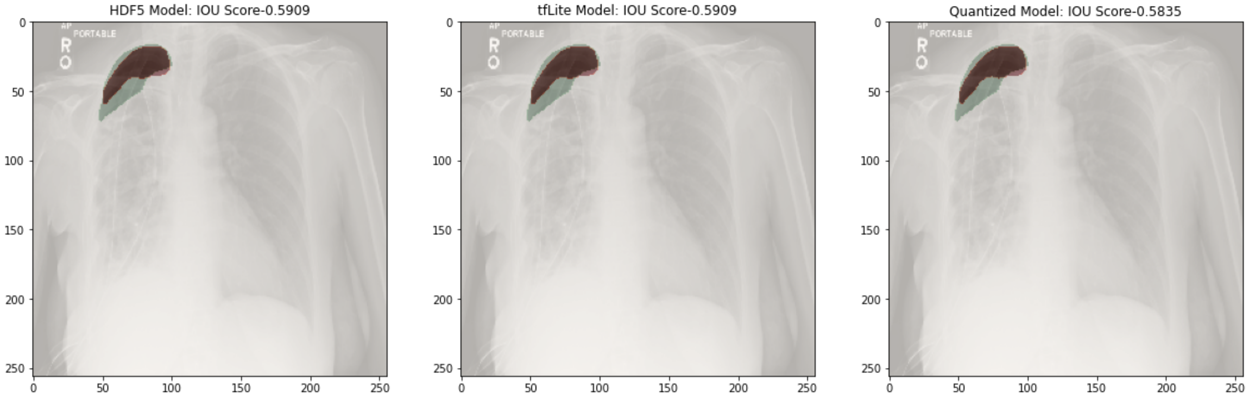
```
ImageID = siim/images_dicom/1.2.276.0.7230010.3.1.4.8323329.4874.1517875185.188564.dcm
```


HDF5 Model: IOU Score-0.0014
tfLite Model: IOU Score-0.0014
Quantized Model: IOU Score-0.0014
0
50
100
150
200
250

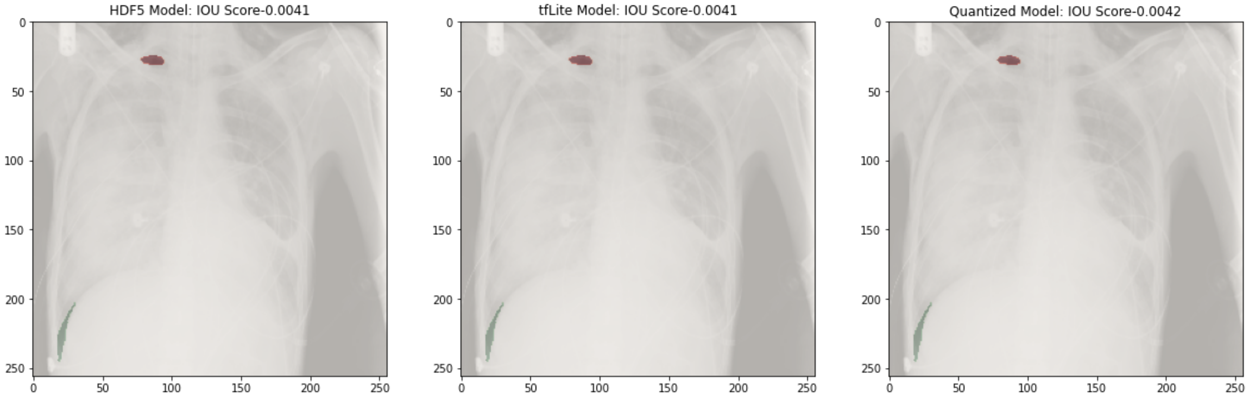
```
ImageID = siim/images_dicom/1.2.276.0.7230010.3.1.4.8323329.1824.1517875169.753287.dcm
```


HDF5 Model: IOU Score-0.2419
tfLite Model: IOU Score-0.2419
Quantized Model: IOU Score-0.2657
0
50
100
150
200
250

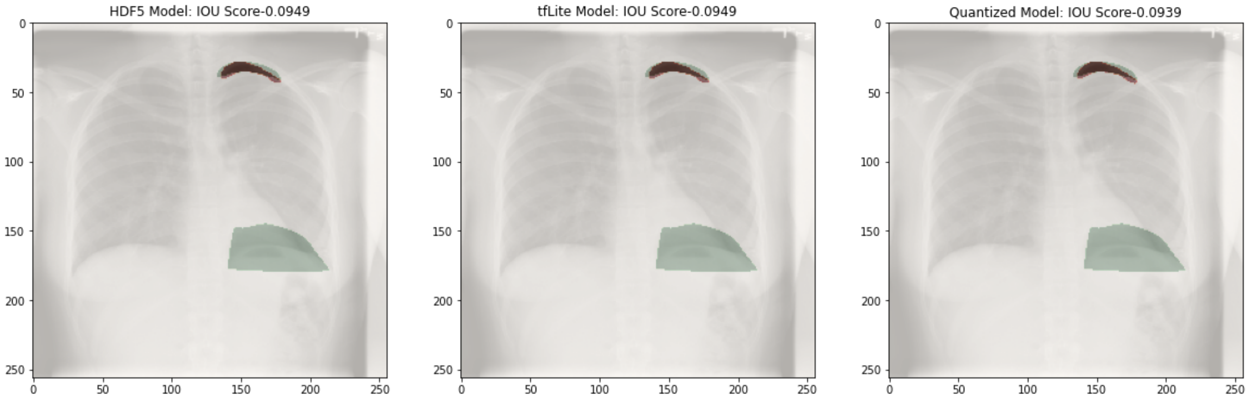
ImageID = siim/images_dicom/1.2.276.0.7230010.3.1.4.8323329.13439.1517875245.328664.dcm


HDF5 Model: IOU Score-0.5909
tfLite Model: IOU Score-0.5909
Quantized Model: IOU Score-0.5835
0
50
100
150
200
250

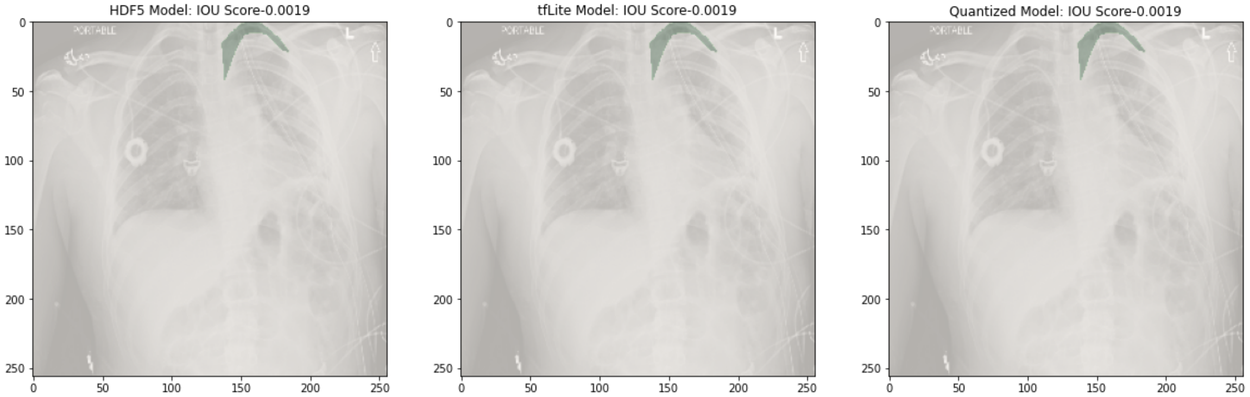
ImageID = siim/images_dicom/1.2.276.0.7230010.3.1.4.8323329.2428.1517875172.652980.dcm


HDF5 Model: IOU Score-0.0041
tfLite Model: IOU Score-0.0041
Quantized Model: IOU Score-0.0042
0
50
100
150
200
250

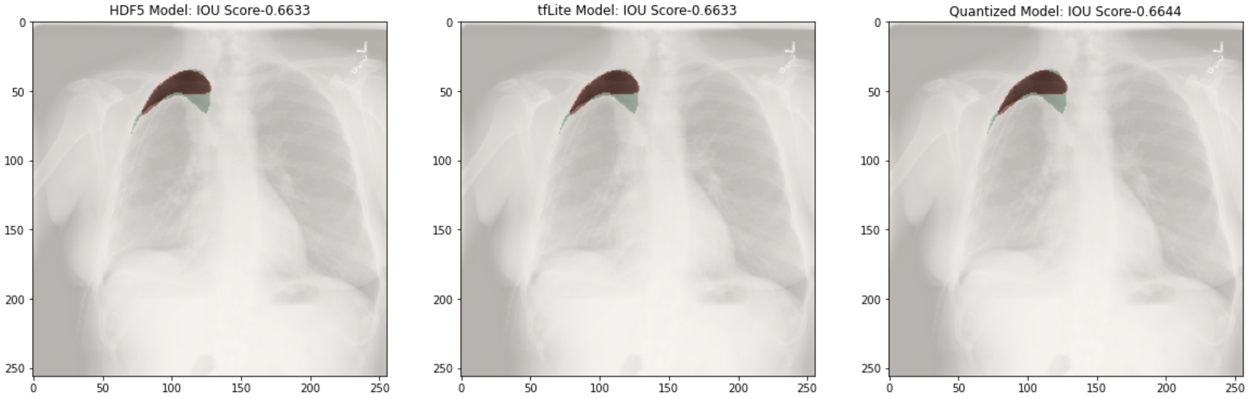
ImageID = siim/images_dicom/1.2.276.0.7230010.3.1.4.8323329.6616.1517875199.238776.dcm


HDF5 Model: IOU Score-0.0949
tfLite Model: IOU Score-0.0949
Quantized Model: IOU Score-0.0939
0
50
100
150
200
250

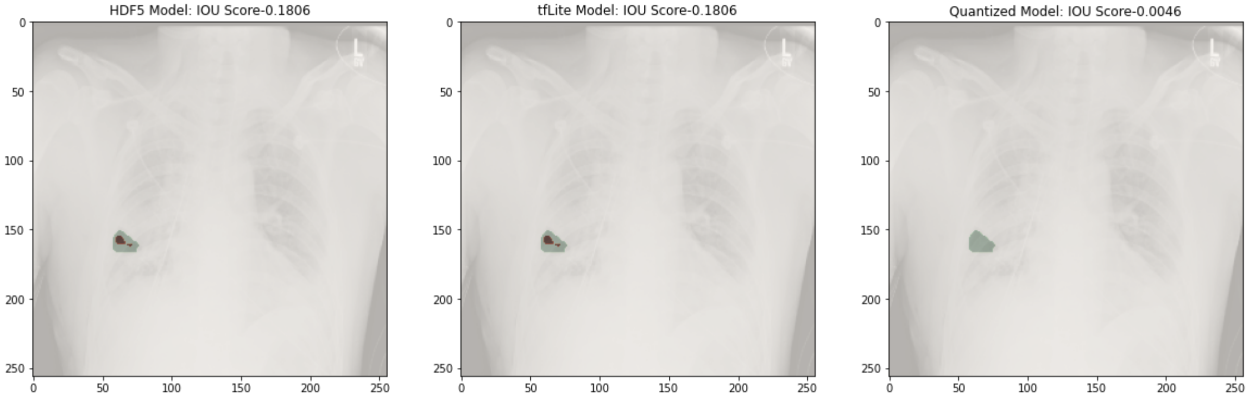
ImageID = siim/images_dicom/1.2.276.0.7230010.3.1.4.8323329.14150.1517875249.852043.dcm


HDF5 Model: IOU Score-0.0019
tfLite Model: IOU Score-0.0019
Quantized Model: IOU Score-0.0019
PORTABLE
L
0
50
100
150
200
250

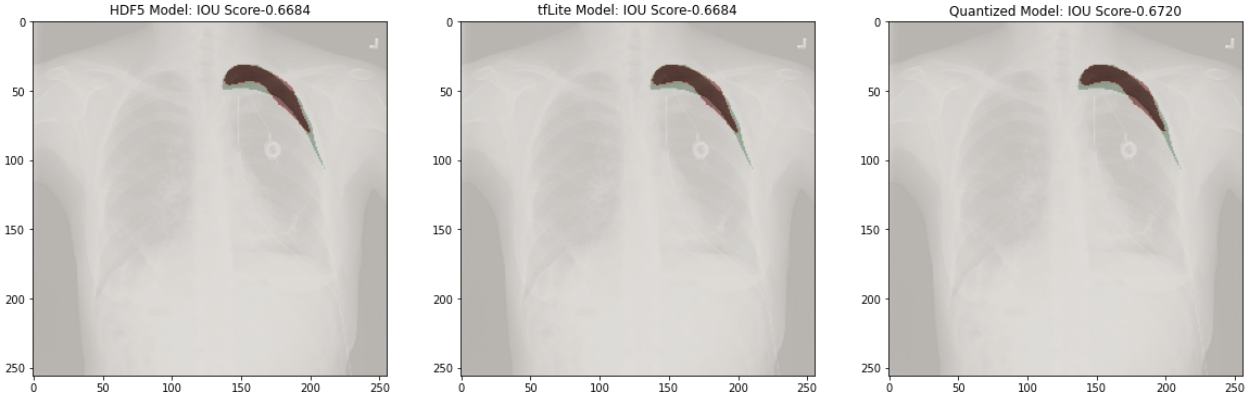
ImageID = siim/images_dicom/1.2.276.0.7230010.3.1.4.8323329.2084.1517875171.71353.dcm


HDF5 Model: IOU Score-0.6633
tfLite Model: IOU Score-0.6633
Quantized Model: IOU Score-0.6644
0
50
100
150
200
250

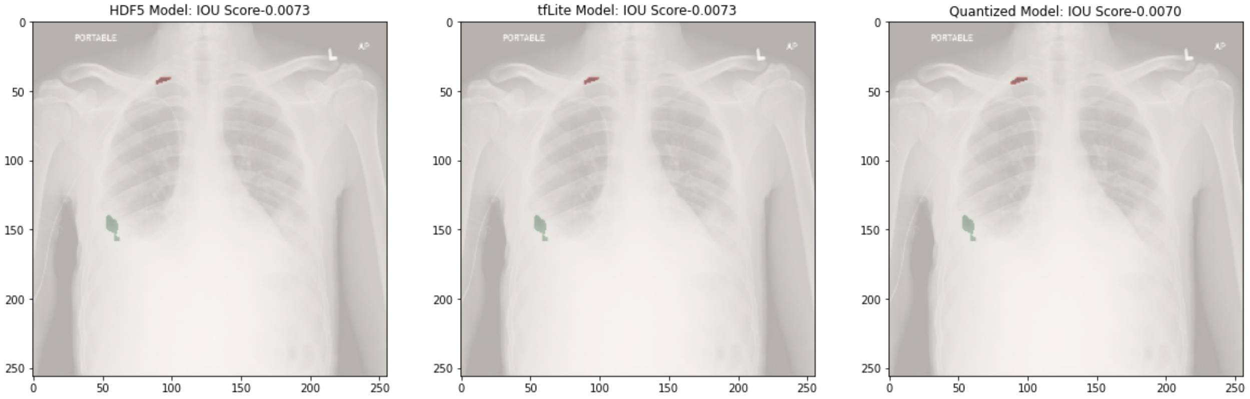
ImageID = siim/images_dicom/1.2.276.0.7230010.3.1.4.8323329.1140.1517875166.465295.dcm


HDF5 Model: IOU Score-0.1806
tfLite Model: IOU Score-0.1806
Quantized Model: IOU Score-0.0046
0
50
100
150
200
250

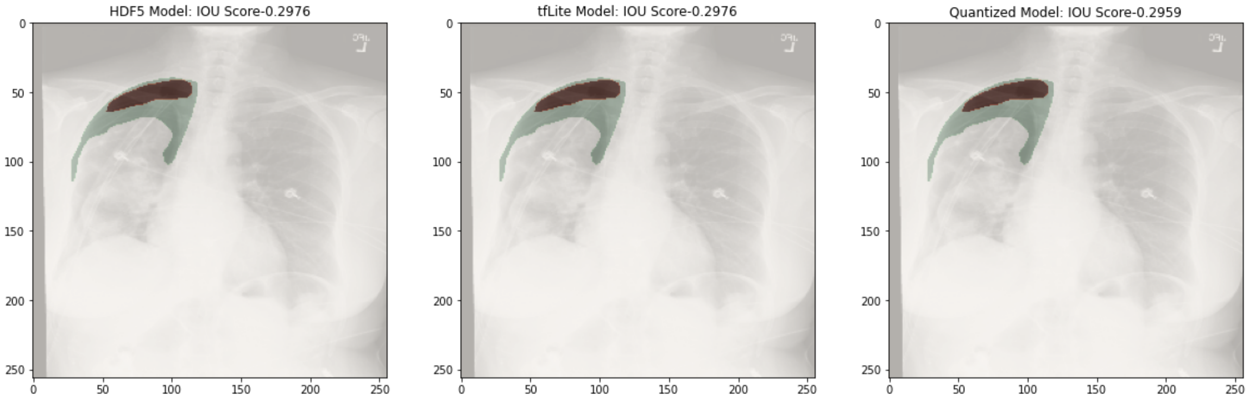
```
ImageID = siim/images_dicom/1.2.276.0.7230010.3.1.4.8323329.1168.1517875166.599906.dcm
```


HDF5 Model: IOU Score-0.6684
tfLite Model: IOU Score-0.6684
Quantized Model: IOU Score-0.6720
0
50
100
150
200
250

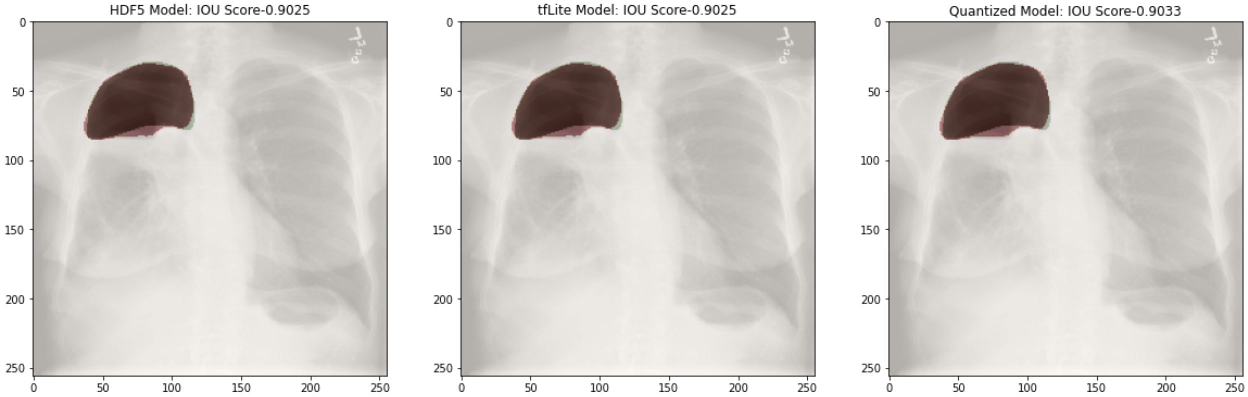
```
ImageID = siim/images_dicom/1.2.276.0.7230010.3.1.4.8323329.32433.1517875160.571504.dcm
```


HDF5 Model: IOU Score-0.0073
tfLite Model: IOU Score-0.0073
Quantized Model: IOU Score-0.0070
PORTABLE
AP
0
50
100
150
200
250

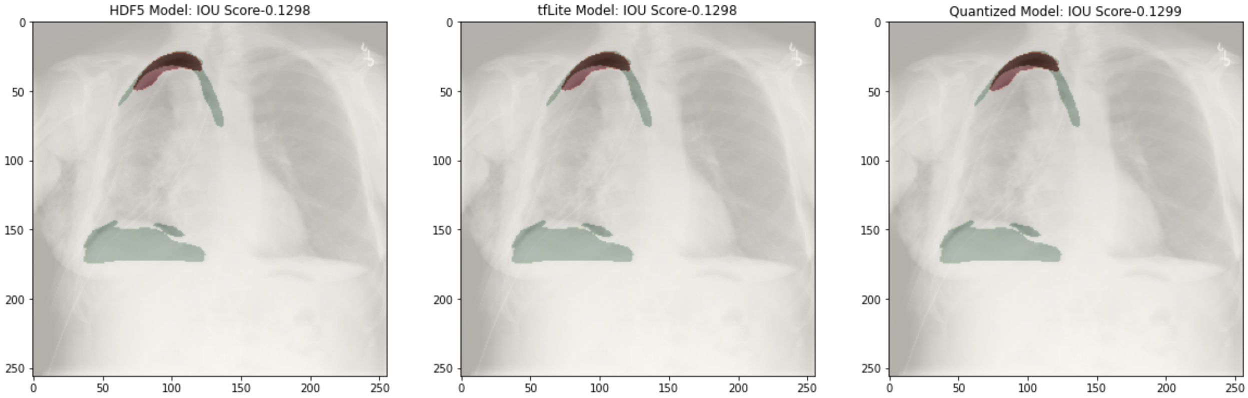
```
ImageID = siim/images_dicom/1.2.276.0.7230010.3.1.4.8323329.10643.1517875224.698414.dcm
```


HDF5 Model: IOU Score-0.2976
tfLite Model: IOU Score-0.2976
Quantized Model: IOU Score-0.2959
0
50
100
150
200
250

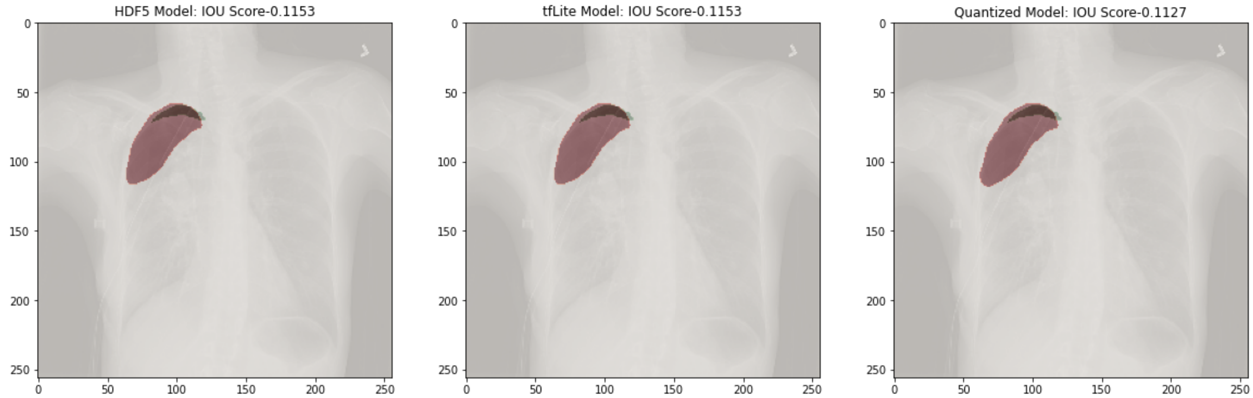
```
ImageID = siim/images_dicom/1.2.276.0.7230010.3.1.4.8323329.14439.1517875252.132263.dcm
```


HDF5 Model: IOU Score-0.9025
tfLite Model: IOU Score-0.9025
Quantized Model: IOU Score-0.9033
0
50
100
150
200
250

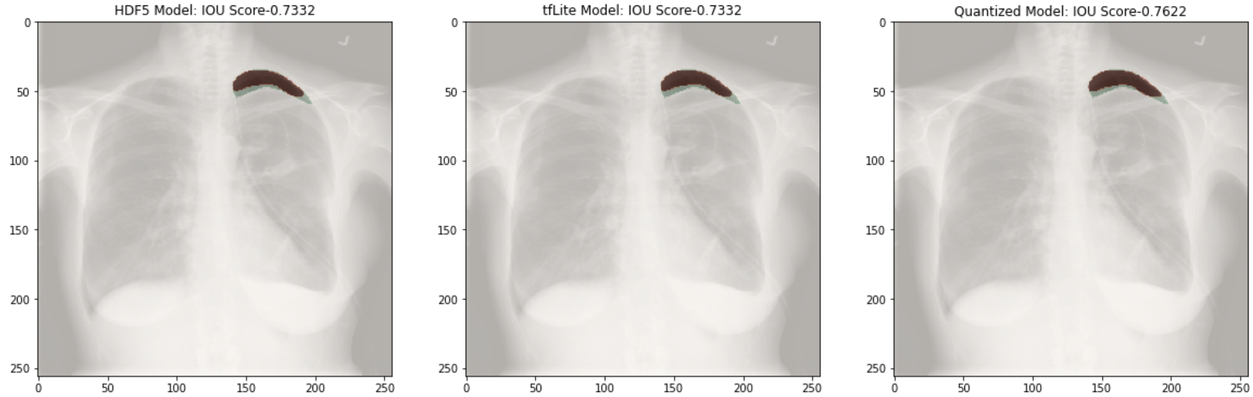
```
ImageID = siim/images_dicom/1.2.276.0.7230010.3.1.4.8323329.6776.1517875201.61444.dcm
```


HDF5 Model: IOU Score-0.1298
tfLite Model: IOU Score-0.1298
Quantized Model: IOU Score-0.1299
0
50
100
150
200
250

```
ImageID = siim/images_dicom/1.2.276.0.7230010.3.1.4.8323329.3984.1517875180.510136.dcm
```


HDF5 Model: IOU Score-0.1153
tfLite Model: IOU Score-0.1153
Quantized Model: IOU Score-0.1127
0
50
100
150
200
250
0
50
100
150
200
250

```
ImageID = siim/images_dicom/1.2.276.0.7230010.3.1.4.8323329.2665.1517875173.947393.dcm
```


HDF5 Model: IOU Score-0.7332
tfLite Model: IOU Score-0.7332
Quantized Model: IOU Score-0.7622
0
50
100
150
200
250
0
50
100
150
200
250

In [40]:

```
# add a new column in val_df dataframe with the iou scores
val_df['iou_score_hdf5'] = iou_score_hdf5
val_df['iou_score_tflite'] = iou_score_tflite
val_df['iou_score_quantized'] = iou_score_quantized
val_df.head(20)
```

Out[40]:

	ImageId	EncodedPixels	is_pneumothorax	
1106	1.2.276.0.7230010.3.1.4.8323329.4703.151787518...	609496 23 991 48 971 83 936 103 918 111 910 11...	1	siim/images_dicom/1.2.27
7989	1.2.276.0.7230010.3.1.4.8323329.1033.151787516...	284490 38 976 58 956 73 941 87 877 149 845 182...	1	siim/images_dicom/1.2.27
8108	1.2.276.0.7230010.3.1.4.8323329.12294.15178752...	233827 23 994 37 981 47 971 57 960 66 955 70 9...	1	siim/images_dicom/1.2.27
11527	1.2.276.0.7230010.3.1.4.8323329.4628.151787518...	190212 2 1015 7 1009 13 992 12 1 18 980 19 100...	1	siim/images_dicom/1.2.27
3796	1.2.276.0.7230010.3.1.4.8323329.11498.15178752...	674944 3 1017 8 1014 10 1012 13 1008 16 1005 2...	1	siim/images_dicom/1.2.27
2071	1.2.276.0.7230010.3.1.4.8323329.4874.151787518...	300436 6 1013 11 1008 13 1006 14 1005 17 1004 ...	1	siim/images_dicom/1.2.27
8173	1.2.276.0.7230010.3.1.4.8323329.1824.151787516...	588947 1 1018 7 1011 13 1005 20 1000 24 998 26...	1	siim/images_dicom/1.2.27
2022	1.2.276.0.7230010.3.1.4.8323329.13439.15178752...	195850 17 1002 27 992 33 986 39 980 45 976 48 ...	1	siim/images_dicom/1.2.27
6614	1.2.276.0.7230010.3.1.4.8323329.2428.151787517...	74657 28 972 57 16 3 945 90 931 91 929 92 929 ...	1	siim/images_dicom/1.2.27
4609	1.2.276.0.7230010.3.1.4.8323329.6616.151787519...	544921 6 1014 11 1011 14 1008 17 1006 18 1004 ...	1	siim/images_dicom/1.2.27
12084	1.2.276.0.7230010.3.1.4.8323329.14150.15178752...	557123 1 1022 12 1010 24 999 35 988 46 976 63 ...	1	siim/images_dicom/1.2.27
3203	1.2.276.0.7230010.3.1.4.8323329.2084.151787517...	291143 3 1016 9 1010 9 1011 9 1012 9 1012 10 1...	1	siim/images_dicom/1.2.27
6633	1.2.276.0.7230010.3.1.4.8323329.1140.151787516...	237173 20 1002 29 993 33 989 37 985 41 982 43 ...	1	siim/images_dicom/1.2.27

	ImageId	EncodedPixels	is_pneumothorax	
12814	1.2.276.0.7230010.3.1.4.8323329.1168.151787516...	558270 3 1011 16 1003 22 998 27 994 31 991 33 ...	1	siim/images_dicom/1.2.27
1079	1.2.276.0.7230010.3.1.4.8323329.32433.15178751...	216636 26 993 32 990 35 988 36 987 38 985 40 9...	1	siim/images_dicom/1.2.27
5577	1.2.276.0.7230010.3.1.4.8323329.10643.15178752...	113052 42 977 52 967 59 961 64 957 67 955 69 9...	1	siim/images_dicom/1.2.27
10870	1.2.276.0.7230010.3.1.4.8323329.14439.15178752...	161066 32 977 52 962 63 951 74 945 80 939 86 9...	1	siim/images_dicom/1.2.27
2814	1.2.276.0.7230010.3.1.4.8323329.6776.151787520...	151179 2 1018 10 31 2 978 13 25 12 972 15 19 2...	1	siim/images_dicom/1.2.27
11516	1.2.276.0.7230010.3.1.4.8323329.3984.151787518...	333094 1 1022 1 1021 2 1021 3 1020 3 1019 5 10...	1	siim/images_dicom/1.2.27
7965	1.2.276.0.7230010.3.1.4.8323329.2665.151787517...	574649 3 1017 12 1011 18 1004 24 999 29 994 32...	1	siim/images_dicom/1.2.27

Observation:

Model loaded from hdf5 file and tflite file giving same iou score and their size also same. Post quantized tflite model size is very less and it is giving approximately same iou score. As the size is reduced a lot, we can easily deploy it in iot devices.