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 129kB/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 9.0MB/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 15:04:59--  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)... 74.125.197.128, 74.125.142.128, 74.125.1
95.128, ...
Connecting to storage.googleapis.com (storage.googleapis.com)|74.125.197.128|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2059765561 (1.9G) [application/zip]
Saving to: ‘archive.zip’

archive.zip         100%[===================>]   1.92G  78.0MB/s    in 27s

2021-03-25 15:05:26 (71.5 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]:

```
# 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

Load Classification model from google drive

```
In [13]: from keras.models import load_model
         model_clf = load_model("gdrive/My Drive/Colab Notebooks/cs2_pneumothorax/classification/weights-07-0.6
         400.hdf5")
```

Convert hdf5 file to tflite version

```
In [14]: # convert the model into tflite version
         converter = tf.lite.TFLiteConverter.from_keras_model(model_clf)
         tflite_model = converter.convert()
         open("gdrive/My Drive/Colab Notebooks/cs2_pneumothorax/classification/converted_clf_model.tflite", "w
         b").write(tflite_model)

         INFO:tensorflow:Assets written to: /tmp/tmp8dcmwehs/assets

Out[14]: 81127036
```

Convert hdf5 to quantized tflite version

```
In [15]: # Convert the model to quantized version with post-training quantization
         converter = tf.lite.TFLiteConverter.from_keras_model(model_clf)
         converter.optimizations = [tf.lite.Optimize.OPTIMIZE_FOR_SIZE]
         tflite_quant_model = converter.convert()
         open("gdrive/My Drive/Colab Notebooks/cs2_pneumothorax/classification/converted_clf_quant_model.tflit
         e", "wb").write(tflite_quant_model)

         INFO:tensorflow:Assets written to: /tmp/tmpwhhmkd17/assets

         INFO:tensorflow:Assets written to: /tmp/tmpwhhmkd17/assets

Out[15]: 20420624
```

Load tflite model

```
In [16]: # https://colab.research.google.com/github/frogermcs/TFLite-Tester/blob/master/notebooks/Testing_TFLit
         e_model.ipynb#scrollTo=OoBmFmXLHVhj
         tflite_interpreter = tf.lite.Interpreter(model_path="gdrive/My Drive/Colab Notebooks/cs2_pneumothorax/
         classification/converted_clf_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 [21]: # Load quantized TFLite model
         tflite_interpreter_quant = tf.lite.Interpreter(model_path="gdrive/My Drive/Colab Notebooks/cs2_pneumot
         horax/classification/converted_clf_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()
```

Predict using all the models

In [23]:

```
# predict using classification model from hdf5 file
val_df = val_df.head(20)
val_image_path = val_df['dicom_path'].values
val_image_prob_hdf5 = []
val_image_prob_tflite = []
val_image_prob_tflite_quantized = []
for file in tqdm(val_image_path):
  size = 256
  image = tf.io.read_file(file)
  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)

  # predict using model loaded from hdf5 file
  pred = model_clf.predict(image)
  val_image_prob_hdf5.append(pred[0])

  # 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'])
  val_image_prob_tflite.append(tflite_model_predictions)

  # predict using quantized tflite model
  tflite_interpreter_quant.set_tensor(input_details[0]['index'], image)
  tflite_interpreter_quant.invoke()
  tflite_model_predictions = tflite_interpreter_quant.get_tensor(output_details[0]['index'])
  val_image_prob_tflite_quantized.append(tflite_model_predictions)
```

In [30]:

```
# convert array to list
pred_prob_hdf5 = [ele[0] for ele in val_image_prob_hdf5]
pred_prob_tflite = [ele[0][0] for ele in val_image_prob_tflite]
pred_prob_quantized = [ele[0][0] for ele in val_image_prob_tflite_quantized]
```

In [31]:

```
# add a new column in val_df dataframe with the predicted probabilities
val_df['pred_prob_hdf5'] = pred_prob_hdf5
val_df['pred_prob_tflite'] = pred_prob_tflite
val_df['pred_prob_quantized'] = pred_prob_quantized
val_df.head(20)
```

Out[31]:

	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
78	1.2.276.0.7230010.3.1.4.8323329.4000.151787518...	-1	0	siim/images_dicom/1.2.27
4061	1.2.276.0.7230010.3.1.4.8323329.10051.15178752...	-1	0	siim/images_dicom/1.2.27
11758	1.2.276.0.7230010.3.1.4.8323329.32405.15178751...	-1	0	siim/images_dicom/1.2.27
11737	1.2.276.0.7230010.3.1.4.8323329.392.1517875162...	222519 7 1013 12 1010 15 1007 17 1005 19 1004 ...	1	siim/images_dicom/1.2.27
5761	1.2.276.0.7230010.3.1.4.8323329.12145.15178752...	-1	0	siim/images_dicom/1.2.27
11300	1.2.276.0.7230010.3.1.4.8323329.13483.15178752...	146908 31 988 47 971 64 956 72 949 76 946 78 9...	1	siim/images_dicom/1.2.27
377	1.2.276.0.7230010.3.1.4.8323329.6244.151787519...	-1	0	siim/images_dicom/1.2.27
7523	1.2.276.0.7230010.3.1.4.8323329.6992.151787520...	630988 1 1024 1 3073 1 1024 1 1023 2 1023 2 10...	1	siim/images_dicom/1.2.27
12724	1.2.276.0.7230010.3.1.4.8323329.11685.15178752...	-1	0	siim/images_dicom/1.2.27
7974	1.2.276.0.7230010.3.1.4.8323329.31767.15178751...	-1	0	siim/images_dicom/1.2.27
5787	1.2.276.0.7230010.3.1.4.8323329.31788.15178751...	-1	0	siim/images_dicom/1.2.27
3328	1.2.276.0.7230010.3.1.4.8323329.11362.15178752...	-1	0	siim/images_dicom/1.2.27
7098	1.2.276.0.7230010.3.1.4.8323329.31930.15178751...	-1	0	siim/images_dicom/1.2.27
5182	1.2.276.0.7230010.3.1.4.8323329.2301.151787517...	-1	0	siim/images_dicom/1.2.27
7016	1.2.276.0.7230010.3.1.4.8323329.10831.15178752...	-1	0	siim/images_dicom/1.2.27

Observation:

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