In [2]:

```
!pip install efficientnet
!pip install iterative-stratification
!pip install gdown

import os
if not os.path.isfile('model_effnet_bo_087.h5'):
    !gdown https://drive.google.com/uc?id=1FXF1HymYbRf3OlThMTXAa74TRup3AhD_
```

```
Requirement already satisfied: efficientnet in /home/surya/anaconda3/lib/python3.6/site-packages (1.0.0)
Requirement already satisfied: scikit-image in /home/surya/anaconda3/lib/python3.6/site-packages (from effic
ientnet) (0.15.0)
Requirement already satisfied: keras-applications<=1.0.8,>=1.0.7 in /home/surya/anaconda3/lib/python3.6/site
-packages (from efficientnet) (1.0.8)
Requirement already satisfied: matplotlib!=3.0.0,>=2.0.0 in /home/surya/anaconda3/lib/python3.6/site-package
s (from scikit-image->efficientnet) (2.2.2)
Requirement already satisfied: scipy>=0.17.0 in /home/surya/anaconda3/lib/python3.6/site-packages (from scik
it-image->efficientnet) (1.1.0)
Requirement already satisfied: imageio>=2.0.1 in /home/surya/anaconda3/lib/python3.6/site-packages (from sci
kit-image->efficientnet) (2.3.0)
Requirement already satisfied: PyWavelets>=0.4.0 in /home/surya/anaconda3/lib/python3.6/site-packages (from
scikit-image->efficientnet) (0.5.2)
Requirement already satisfied: pillow>=4.3.0 in /home/surya/anaconda3/lib/python3.6/site-packages (from scik
it-image->efficientnet) (6.2.0)
Requirement already satisfied: networkx>=2.0 in /home/surya/anaconda3/lib/python3.6/site-packages (from scik
it-image->efficientnet) (2.1)
Requirement already satisfied: h5py in /home/surya/anaconda3/lib/python3.6/site-packages (from keras-applica
tions<=1.0.8,>=1.0.7->efficientnet) (2.7.1)
Requirement already satisfied: numpy>=1.9.1 in /home/surya/anaconda3/lib/python3.6/site-packages (from keras
-applications<=1.0.8,>=1.0.7->efficientnet) (1.17.3)
Requirement already satisfied: cycler>=0.10 in /home/surya/anaconda3/lib/python3.6/site-packages (from matpl
otlib!=3.0.0,>=2.0.0->scikit-image->efficientnet) (0.10.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /home/surya/anaconda3/lib/python
3.6/site-packages (from matplotlib!=3.0.0,>=2.0.0->scikit-image->efficientnet) (2.2.0)
Requirement already satisfied: python-dateutil>=2.1 in /home/surya/.local/lib/python3.6/site-packages (from
matplotlib!=3.0.0,>=2.0.0->scikit-image->efficientnet) (2.7.3)
Requirement already satisfied: pytz in /home/surya/.local/lib/python3.6/site-packages (from matplotlib!=3.0.
0,>=2.0.0->scikit-image->efficientnet) (2018.5)
Requirement already satisfied: six>=1.10 in /home/surya/.local/lib/python3.6/site-packages (from matplotlib!
=3.0.0,>=2.0.0->scikit-image->efficientnet) (1.11.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /home/surya/anaconda3/lib/python3.6/site-packages (from
matplotlib!=3.0.0,>=2.0.0->scikit-image->efficientnet) (1.0.1)
Requirement already satisfied: decorator>=4.1.0 in /home/surya/anaconda3/lib/python3.6/site-packages (from n
etworkx>=2.0->scikit-image->efficientnet) (4.3.0)
Requirement already satisfied: setuptools in /home/surya/anaconda3/lib/python3.6/site-packages (from kiwisol
ver>=1.0.1->matplotlib!=3.0.0,>=2.0.0->scikit-image->efficientnet) (39.1.0)
WARNING: You are using pip version 19.1.1, however version 19.3.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
Requirement already satisfied: iterative-stratification in /home/surya/anaconda3/lib/python3.6/site-packages
(0.1.6)
Requirement already satisfied: numpy in /home/surya/anaconda3/lib/python3.6/site-packages (from iterative-st
ratification) (1.17.3)
Requirement already satisfied: scipy in /home/surya/anaconda3/lib/python3.6/site-packages (from iterative-st
ratification) (1.1.0)
Requirement already satisfied: scikit-learn in /home/surya/anaconda3/lib/python3.6/site-packages (from itera
tive-stratification) (0.20.3)
WARNING: You are using pip version 19.1.1, however version 19.3.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
Requirement already satisfied: gdown in /home/surya/anaconda3/lib/python3.6/site-packages (3.8.1)
Requirement already satisfied: filelock in /home/surya/anaconda3/lib/python3.6/site-packages (from gdown)
 (3.0.4)
Requirement already satisfied: six in /home/surya/.local/lib/python3.6/site-packages (from gdown) (1.11.0)
Requirement already satisfied: tqdm in /home/surya/anaconda3/lib/python3.6/site-packages (from gdown) (4.28.
1)
Requirement already satisfied: requests in /home/surya/anaconda3/lib/python3.6/site-packages (from gdown)
 (2.18.4)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in /home/surya/anaconda3/lib/python3.6/site-packages (f
rom requests->gdown) (3.0.4)
Requirement already satisfied: idna<2.7,>=2.5 in /home/surya/anaconda3/lib/python3.6/site-packages (from req
uests->gdown) (2.6)
Requirement already satisfied: urllib3<1.23,>=1.21.1 in /home/surya/anaconda3/lib/python3.6/site-packages (f
rom requests->gdown) (1.22)
Requirement already satisfied: certifi>=2017.4.17 in /home/surya/anaconda3/lib/python3.6/site-packages (from
requests->gdown) (2018.4.16)
WARNING: You are using pip version 19.1.1, however version 19.3.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
```

In [4]:

```
import numpy as np
import pandas as pd
import pydicom
import cv2
import tensorflow as tf
import multiprocessing
from math import ceil, floor
import keras
import keras.backend as K
from keras.callbacks import Callback, ModelCheckpoint
from keras.layers import Dense, Flatten, Dropout
from keras.models import Model, load_model
from keras.utils import Sequence
from keras.losses import binary_crossentropy
from keras.optimizers import Adam
import efficientnet.keras as efn
from iterstrat.ml_stratifiers import MultilabelStratifiedShuffleSplit
import matplotlib.pyplot as plt
```

In [5]:

```
HEIGHT = 256
WIDTH = 256
CHANNELS = 3
SHAPE = (HEIGHT, WIDTH, CHANNELS)
```

In [6]:

```
def correct_dcm(dcm):
    x = dcm.pixel_array + 1000
    px_mode = 4096
    x[x>=px_mode] = x[x>=px_mode] - px_mode
    dcm.PixelData = x.tobytes()
    dcm.RescaleIntercept = -1000

def window_image(dcm, window_center, window_width):
    if (dcm.BitsStored == 12) and (dcm.PixelRepresentation == 0) and (int(dcm.RescaleIntercept) > -100):
        correct_dcm(dcm)
    img = dcm.pixel_array * dcm.RescaleSlope + dcm.RescaleIntercept

    # Resize
    img = cv2.resize(img, SHAPE[:2], interpolation = cv2.INTER_LINEAR)

    img_min = window_center - window_width // 2
    img_max = window_center + window_width // 2
    img = np.clip(img, img_min, img_max)
    return img

def bsb_window(dcm):
    brain_img = window_image(dcm, 40, 80)
    subdural_img = window_image(dcm, 80, 200)
    soft_img = window_image(dcm, 40, 380)

    brain_img = (brain_img - 0) / 80
    subdural_img = (subdural_img - (-20)) / 200
    soft_img = (soft_img - (-150)) / 380
    bsb_img = np.array([brain_img, subdural_img, soft_img]).transpose(1,2,0)
    return bsb_img

def _read(path, SHAPE):
    dcm = pydicom.dcmread(path)
    try:
        img = bsb_window(dcm)
    except:
        img = np.zeros(SHAPE)
    return img
```

In [7]:

```
import matplotlib.pyplot as plt
```

Import the training and test datasets.

In [8]:

```
base_model =  efn.EfficientNetB0(weights = 'imagenet', include_top = False, \
                                 pooling = 'avg', input_shape = (HEIGHT, WIDTH, 3))

x = base_model.output
x = Dropout(0.125)(x)
output_layer = Dense(6, activation = 'sigmoid')(x)
model = Model(inputs=base_model.input, outputs=output_layer)
model.compile(optimizer = Adam(lr = 0.0001),
              loss = 'binary_crossentropy',
              metrics = ['acc'])
```

In [9]:

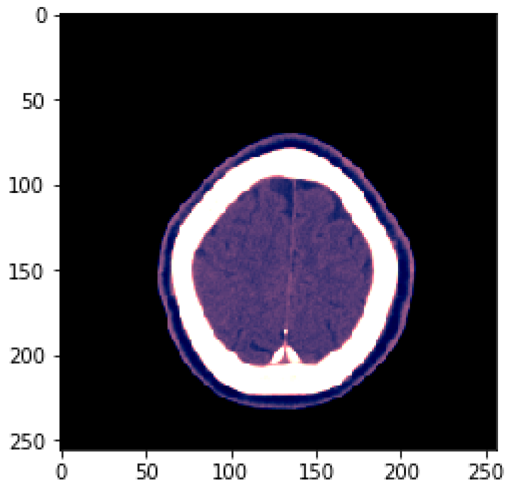
```
model.load_weights('model_effnet_bo_087.h5')
```

In [10]:
```
img_data = _read('ID_000000e27.dcm', (256, 256))
img_data.shape
```

Out[10]:
```
(256, 256, 3)
```

In [11]:
```
plt.imshow(img_data)
```

Out[11]:
```
<matplotlib.image.AxesImage at 0x7f9f047b2c18>
```



In [12]:
```
# Prepare the data
# 1 ==> Batch size
# 256, 256, 3 is the image shape
X = np.empty((1, 256,256, 3))
X[0] = img_data
X.shape
```

Out[12]:
```
(1, 256, 256, 3)
```

In [13]:
```
preds = model.predict(X)
preds
```

Out[13]:
```
array([[0.05049768, 0.00077422, 0.00192805, 0.00035753, 0.0012597 ,
        0.02818136]], dtype=float32)
```

In [14]:
```
preds.shape
```

Out[14]:
```
(1, 6)
```