Skin Lesion Segmentation

Connecting Google Drive

In [ ]:

```
!pip install -U -q PyDrive
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from google.colab import auth
from oauth2client.client import GoogleCredentials
```

In [ ]:

```
auth.authenticate_user()
gauth = GoogleAuth()
gauth.credentials = GoogleCredentials.get_application_default()
drive = GoogleDrive(gauth)
```

Extracting the Training data

In [ ]:

```
# Importing train data from google drive
fid = drive.ListFile({'q':"title='Train_Data.rar'"}).GetList()[0]['id']
f = drive.CreateFile({'id': fid})
f.GetContentFile('Train_Data.rar')
```

In [ ]:

```
# unzipping the contents of zip folder
!pip install unrar
!unrar x Train_Data
```

Importing the required Libraries

In [ ]:

```
import os
import random
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
plt.style.use("ggplot")
%matplotlib inline

from tqdm import tqdm_notebook, tnrange
from itertools import chain
from skimage.io import imread, imshow, concatenate_images
from skimage.transform import resize
from skimage.morphology import label
from sklearn.model_selection import train_test_split

import tensorflow as tf

from keras.models import Model, load_model
from keras.layers import Input, BatchNormalization, Activation, Dense, Dropout
from keras.layers.core import Lambda, RepeatVector, Reshape
from keras.layers.convolutional import Conv2D, Conv2DTranspose
from keras.layers.pooling import MaxPooling2D, GlobalMaxPool2D
from keras.layers.merge import concatenate, add
from keras.callbacks import EarlyStopping, ModelCheckpoint, ReduceLROnPlateau
from keras.optimizers import Adam
from keras.preprocessing.image import ImageDataGenerator, array_to_img, img_to_array, load_img
```

In [ ]:

```
seed = 42
np.random.seed = seed

IMG_WIDTH = 384
IMG_HEIGHT = 256
IMG_CHANNELS = 3
```

In [ ]:

```
TRAIN_PATH = 'Train_Data/BWT/'
TRAIN_PATH_MASK='Train_Data/MASK/'
```

In [ ]:

```
# removing unwanted files
test = os.listdir(TRAIN_PATH)
for item in test:
    if item.endswith(".ini"):
        os.remove(os.path.join(TRAIN_PATH, item))
```

In [ ]:

```
train_ids = next(os.walk(TRAIN_PATH))[2]
train_ids = np.sort(train_ids)
print("No. of images = ", len(train_ids))
#test_ids = next(os.walk(TEST_PATH))[2]
#train_ids = np.delete(train_ids,(0),axis=0)
X = np.zeros((len(train_ids),IMG_HEIGHT, IMG_WIDTH,IMG_CHANNELS),dtype=np.uint8)
Y = np.zeros((len(train_ids),IMG_HEIGHT, IMG_WIDTH, 1), dtype=np.bool)
```

```
No. of images =  6452
```

In [ ]:

```
# tqdm is used to display the progress bar
for n, id_ in tqdm_notebook(enumerate(train_ids), total=len(train_ids)):
    path = TRAIN_PATH + id_
    img = imread(path)[:,:,:IMG_CHANNELS]
    X[n] = img
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:2: TqdmDeprecationWarning: This function will be removed in tqdm==5.
0.0
Please use `tqdm.notebook.tqdm` instead of `tqdm.tqdm_notebook`
```

In [ ]:

```
test = os.listdir(TRAIN_PATH_MASK)
for item in test:
    if item.endswith(".ini"):
        os.remove(os.path.join(TRAIN_PATH_MASK, item))

train_mask_ids = next(os.walk(TRAIN_PATH_MASK))[2]
train_mask_ids = np.sort(train_mask_ids)
print("No. of images = ", len(train_mask_ids))
for n1, id1_ in tqdm_notebook(enumerate(train_mask_ids), total=len(train_mask_ids)):
    mask = img_to_array(load_img("Train_Data/MASK/"+id1_, grayscale=True))
    Y[n1] = mask
```

```
No. of images =  6452
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:9: TqdmDeprecationWarning: This function will be removed in tqdm==5.
0.0
Please use `tqdm.notebook.tqdm` instead of `tqdm.tqdm_notebook`
  if __name__ == '__main__':

/usr/local/lib/python3.6/dist-packages/keras_preprocessing/image/utils.py:107: UserWarning: grayscale is deprecated. Please use co
lor_mode = "grayscale"
```

```
  warnings.warn('grayscale is deprecated. Please use '
```

Data spliting for Training and Validation

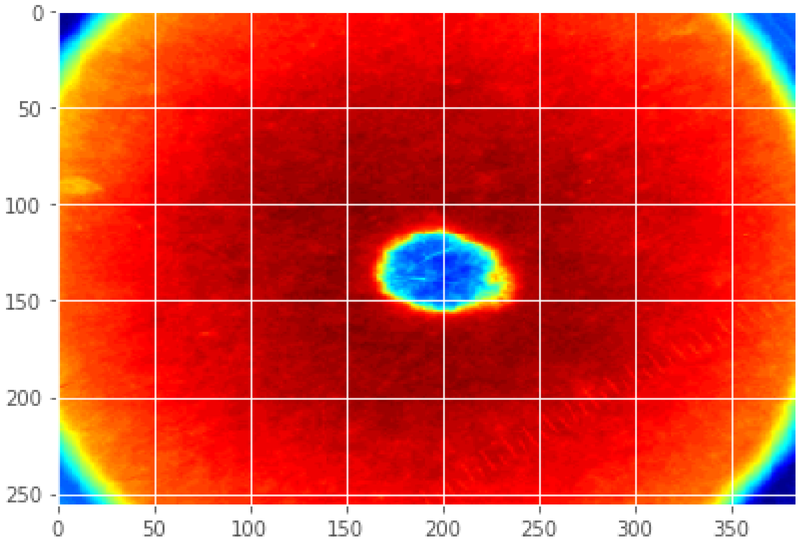
In [ ]:

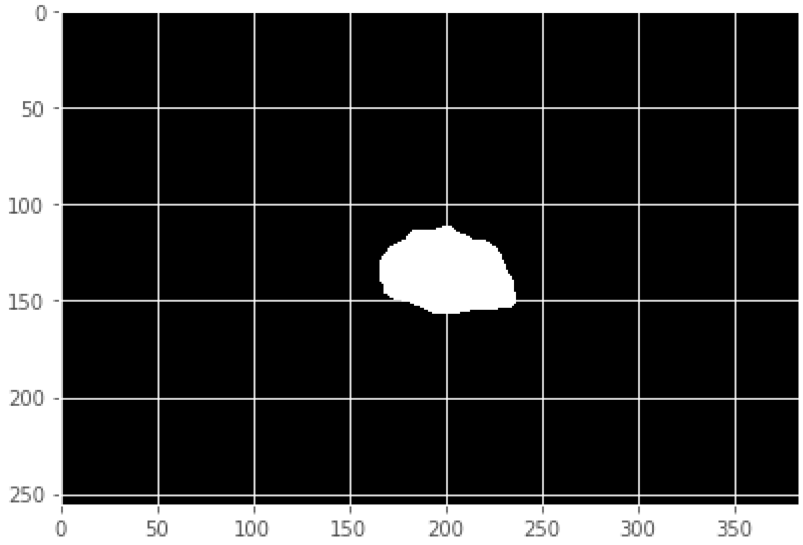
```
X_train, X_valid, y_train, y_valid = train_test_split(X, Y, test_size=0.2, random_state=42)
```

Random dermoscopic image with its corresponding Segmentation map

In [ ]:

```
image_x = random.randint(0, len(train_ids))
imshow(X[image_x])
plt.show()
imshow(np.squeeze(Y[image_x]))
plt.show()
```





Network hidden layers

In [ ]:

```
def conv2d_block(input_tensor, n_filters, kernel_size = 3, batchnorm = True):
    """Function to add 2 convolutional layers with the parameters passed to it"""
    # first layer
    x = Conv2D(filters = n_filters, kernel_size = (kernel_size, kernel_size),\
              kernel_initializer = 'he_normal', padding = 'same')(input_tensor)
    if batchnorm:
        x = BatchNormalization()(x)
    x = Activation('relu')(x)

    # second layer
    x = Conv2D(filters = n_filters, kernel_size = (kernel_size, kernel_size),\
              kernel_initializer = 'he_normal', padding = 'same')(input_tensor)
    if batchnorm:
        x = BatchNormalization()(x)
    x = Activation('relu')(x)

    return x
```

In [ ]:

```
def get_unet(input_img, n_filters = 16, dropout = 0.1, batchnorm = True):
    """Function to define the UNET Model"""
    # Contracting Path
    c1 = conv2d_block(input_img, n_filters * 1, kernel_size = 3, batchnorm = batchnorm)
    p1 = MaxPooling2D((2, 2))(c1)
    p1 = Dropout(dropout)(p1)

    c2 = conv2d_block(p1, n_filters * 2, kernel_size = 3, batchnorm = batchnorm)
    p2 = MaxPooling2D((2, 2))(c2)
    p2 = Dropout(dropout)(p2)

    c3 = conv2d_block(p2, n_filters * 4, kernel_size = 3, batchnorm = batchnorm)
    p3 = MaxPooling2D((2, 2))(c3)
    p3 = Dropout(dropout)(p3)

    c4 = conv2d_block(p3, n_filters * 8, kernel_size = 3, batchnorm = batchnorm)
    p4 = MaxPooling2D((2, 2))(c4)
    p4 = Dropout(dropout)(p4)

    c5 = conv2d_block(p4, n_filters = n_filters * 16, kernel_size = 3, batchnorm = batchnorm)

    # Expansive Path
    u6 = Conv2DTranspose(n_filters * 8, (3, 3), strides = (2, 2), padding = 'same')(c5)
    u6 = concatenate([u6, c4])
    u6 = Dropout(dropout)(u6)
    c6 = conv2d_block(u6, n_filters * 8, kernel_size = 3, batchnorm = batchnorm)

    u7 = Conv2DTranspose(n_filters * 4, (3, 3), strides = (2, 2), padding = 'same')(c6)
    u7 = concatenate([u7, c3])
    u7 = Dropout(dropout)(u7)
    c7 = conv2d_block(u7, n_filters * 4, kernel_size = 3, batchnorm = batchnorm)

    u8 = Conv2DTranspose(n_filters * 2, (3, 3), strides = (2, 2), padding = 'same')(c7)
    u8 = concatenate([u8, c2])
    u8 = Dropout(dropout)(u8)
    c8 = conv2d_block(u8, n_filters * 2, kernel_size = 3, batchnorm = batchnorm)

    u9 = Conv2DTranspose(n_filters * 1, (3, 3), strides = (2, 2), padding = 'same')(c8)
    u9 = concatenate([u9, c1])
    u9 = Dropout(dropout)(u9)
    c9 = conv2d_block(u9, n_filters * 1, kernel_size = 3, batchnorm = batchnorm)

    outputs = Conv2D(1, (1, 1), activation='sigmoid')(c9)
    model = Model(inputs=[input_img], outputs=[outputs])
    return model
```

In [ ]:

```
input_img = Input((IMG_HEIGHT,IMG_WIDTH, 3), name='img')
model = get_unet(input_img, n_filters=16, dropout=0.05, batchnorm=True)
model.compile(optimizer=Adam(), loss="binary_crossentropy", metrics=["accuracy"])
```

Network Structure

In [ ]:

```
model.summary()
```

```
Model: "functional_2"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
img (InputLayer)                [(None, 256, 384, 3) 0
__________________________________________________________________________________________________
conv2d_20 (Conv2D)              (None, 256, 384, 16) 448         img[0][0]
__________________________________________________________________________________________________
batch_normalization_19 (BatchNo (None, 256, 384, 16) 64          conv2d_20[0][0]
__________________________________________________________________________________________________
activation_19 (Activation)      (None, 256, 384, 16) 0           batch_normalization_19[0][0]
__________________________________________________________________________________________________
max_pooling2d_4 (MaxPooling2D)  (None, 128, 192, 16) 0           activation_19[0][0]
__________________________________________________________________________________________________
dropout_8 (Dropout)             (None, 128, 192, 16) 0           max_pooling2d_4[0][0]
__________________________________________________________________________________________________
conv2d_22 (Conv2D)              (None, 128, 192, 32) 4640        dropout_8[0][0]
__________________________________________________________________________________________________
batch_normalization_21 (BatchNo (None, 128, 192, 32) 128         conv2d_22[0][0]
__________________________________________________________________________________________________
activation_21 (Activation)      (None, 128, 192, 32) 0           batch_normalization_21[0][0]
__________________________________________________________________________________________________
max_pooling2d_5 (MaxPooling2D)  (None, 64, 96, 32)   0           activation_21[0][0]
__________________________________________________________________________________________________
dropout_9 (Dropout)             (None, 64, 96, 32)   0           max_pooling2d_5[0][0]
__________________________________________________________________________________________________
conv2d_24 (Conv2D)              (None, 64, 96, 64)   18496       dropout_9[0][0]
__________________________________________________________________________________________________
batch_normalization_23 (BatchNo (None, 64, 96, 64)   256         conv2d_24[0][0]
__________________________________________________________________________________________________
activation_23 (Activation)      (None, 64, 96, 64)   0           batch_normalization_23[0][0]
__________________________________________________________________________________________________
max_pooling2d_6 (MaxPooling2D)  (None, 32, 48, 64)   0           activation_23[0][0]
__________________________________________________________________________________________________
dropout_10 (Dropout)            (None, 32, 48, 64)   0           max_pooling2d_6[0][0]
__________________________________________________________________________________________________
```

```
conv2d_26 (Conv2D)              (None, 32, 48, 128)  73856       dropout_10[0][0]                 
__________________________________________________________________________________________________
batch_normalization_25 (BatchNo (None, 32, 48, 128)  512         conv2d_26[0][0]                  
__________________________________________________________________________________________________
activation_25 (Activation)      (None, 32, 48, 128)  0           batch_normalization_25[0][0]     
__________________________________________________________________________________________________
max_pooling2d_7 (MaxPooling2D)  (None, 16, 24, 128)  0           activation_25[0][0]              
__________________________________________________________________________________________________
dropout_11 (Dropout)            (None, 16, 24, 128)  0           max_pooling2d_7[0][0]            
__________________________________________________________________________________________________
conv2d_28 (Conv2D)              (None, 16, 24, 256)  295168      dropout_11[0][0]                 
__________________________________________________________________________________________________
batch_normalization_27 (BatchNo (None, 16, 24, 256)  1024        conv2d_28[0][0]                  
__________________________________________________________________________________________________
activation_27 (Activation)      (None, 16, 24, 256)  0           batch_normalization_27[0][0]     
__________________________________________________________________________________________________
conv2d_transpose_4 (Conv2DTrans (None, 32, 48, 128)  295040      activation_27[0][0]              
__________________________________________________________________________________________________
concatenate_4 (Concatenate)     (None, 32, 48, 256)  0           conv2d_transpose_4[0][0]         
                                                                 activation_25[0][0]              
__________________________________________________________________________________________________
dropout_12 (Dropout)            (None, 32, 48, 256)  0           concatenate_4[0][0]              
__________________________________________________________________________________________________
conv2d_30 (Conv2D)              (None, 32, 48, 128)  295040      dropout_12[0][0]                 
__________________________________________________________________________________________________
batch_normalization_29 (BatchNo (None, 32, 48, 128)  512         conv2d_30[0][0]                  
__________________________________________________________________________________________________
activation_29 (Activation)      (None, 32, 48, 128)  0           batch_normalization_29[0][0]     
__________________________________________________________________________________________________
conv2d_transpose_5 (Conv2DTrans (None, 64, 96, 64)   73792       activation_29[0][0]              
__________________________________________________________________________________________________
concatenate_5 (Concatenate)     (None, 64, 96, 128)  0           conv2d_transpose_5[0][0]         
                                                                 activation_23[0][0]              
__________________________________________________________________________________________________
dropout_13 (Dropout)            (None, 64, 96, 128)  0           concatenate_5[0][0]              
__________________________________________________________________________________________________
conv2d_32 (Conv2D)              (None, 64, 96, 64)   73792       dropout_13[0][0]                 
__________________________________________________________________________________________________
batch_normalization_31 (BatchNo (None, 64, 96, 64)   256         conv2d_32[0][0]                  
__________________________________________________________________________________________________
activation_31 (Activation)      (None, 64, 96, 64)   0           batch_normalization_31[0][0]     
__________________________________________________________________________________________________
conv2d_transpose_6 (Conv2DTrans (None, 128, 192, 32) 18464       activation_31[0][0]              
__________________________________________________________________________________________________
concatenate_6 (Concatenate)     (None, 128, 192, 64) 0           conv2d_transpose_6[0][0]         
                                                                 activation_21[0][0]              
__________________________________________________________________________________________________
```

```
dropout_14 (Dropout)            (None, 128, 192, 64) 0           concatenate_6[0][0]
__________________________________________________________________________________________________
conv2d_34 (Conv2D)              (None, 128, 192, 32) 18464       dropout_14[0][0]
__________________________________________________________________________________________________
batch_normalization_33 (BatchNo (None, 128, 192, 32) 128         conv2d_34[0][0]
__________________________________________________________________________________________________
activation_33 (Activation)      (None, 128, 192, 32) 0           batch_normalization_33[0][0]
__________________________________________________________________________________________________
conv2d_transpose_7 (Conv2DTrans (None, 256, 384, 16) 4624        activation_33[0][0]
__________________________________________________________________________________________________
concatenate_7 (Concatenate)     (None, 256, 384, 32) 0           conv2d_transpose_7[0][0]
                                                                 activation_19[0][0]
__________________________________________________________________________________________________
dropout_15 (Dropout)            (None, 256, 384, 32) 0           concatenate_7[0][0]
__________________________________________________________________________________________________
conv2d_36 (Conv2D)              (None, 256, 384, 16) 4624        dropout_15[0][0]
__________________________________________________________________________________________________
batch_normalization_35 (BatchNo (None, 256, 384, 16) 64          conv2d_36[0][0]
__________________________________________________________________________________________________
activation_35 (Activation)      (None, 256, 384, 16) 0           batch_normalization_35[0][0]
__________________________________________________________________________________________________
conv2d_37 (Conv2D)              (None, 256, 384, 1)  17          activation_35[0][0]
==================================================================================================
Total params: 1,179,409
Trainable params: 1,177,937
Non-trainable params: 1,472
__________________________________________________________________________________________________
```

In [ ]:

```
callbacks = [
    EarlyStopping(patience=10, verbose=1),
    ReduceLROnPlateau(factor=0.1, patience=5, min_lr=0.00001, verbose=1),
    ModelCheckpoint('model-skin-lesion-segmentation.h5', verbose=1, save_best_only=True, save_weights_only=True)
]
```

Network training

In [ ]:

```
results = model.fit(X_train, y_train, batch_size=32, epochs=50, callbacks=callbacks,\
                    validation_data=(X_valid, y_valid))
```

```
Epoch 1/50
162/162 [==============================] - ETA: 0s - loss: 0.1720 - accuracy: 0.9542
Epoch 00001: val_loss improved from inf to 0.12847, saving model to model-skin-lesion-segmentation.h5
162/162 [==============================] - 66s 406ms/step - loss: 0.1720 - accuracy: 0.9542 - val_loss: 0.1285 - val_accuracy: 0.9
```

```
588
Epoch 2/50
162/162 [==============================] - ETA: 0s - loss: 0.0970 - accuracy: 0.9681
Epoch 00002: val_loss improved from 0.12847 to 0.08693, saving model to model-skin-lesion-segmentation.h5
162/162 [==============================] - 65s 400ms/step - loss: 0.0970 - accuracy: 0.9681 - val_loss: 0.0869 - val_accuracy: 0.9
703
Epoch 3/50
162/162 [==============================] - ETA: 0s - loss: 0.0822 - accuracy: 0.9709
Epoch 00003: val_loss improved from 0.08693 to 0.08025, saving model to model-skin-lesion-segmentation.h5
162/162 [==============================] - 65s 402ms/step - loss: 0.0822 - accuracy: 0.9709 - val_loss: 0.0803 - val_accuracy: 0.9
714
Epoch 4/50
162/162 [==============================] - ETA: 0s - loss: 0.0755 - accuracy: 0.9725
Epoch 00004: val_loss improved from 0.08025 to 0.07597, saving model to model-skin-lesion-segmentation.h5
162/162 [==============================] - 65s 402ms/step - loss: 0.0755 - accuracy: 0.9725 - val_loss: 0.0760 - val_accuracy: 0.9
725
Epoch 5/50
162/162 [==============================] - ETA: 0s - loss: 0.0737 - accuracy: 0.9727
Epoch 00005: val_loss did not improve from 0.07597
162/162 [==============================] - 65s 401ms/step - loss: 0.0737 - accuracy: 0.9727 - val_loss: 0.0956 - val_accuracy: 0.9
657
Epoch 6/50
162/162 [==============================] - ETA: 0s - loss: 0.0704 - accuracy: 0.9737
Epoch 00006: val_loss improved from 0.07597 to 0.07452, saving model to model-skin-lesion-segmentation.h5
162/162 [==============================] - 65s 401ms/step - loss: 0.0704 - accuracy: 0.9737 - val_loss: 0.0745 - val_accuracy: 0.9
736
Epoch 7/50
162/162 [==============================] - ETA: 0s - loss: 0.0680 - accuracy: 0.9746
Epoch 00007: val_loss improved from 0.07452 to 0.06550, saving model to model-skin-lesion-segmentation.h5
162/162 [==============================] - 65s 401ms/step - loss: 0.0680 - accuracy: 0.9746 - val_loss: 0.0655 - val_accuracy: 0.9
757
Epoch 8/50
162/162 [==============================] - ETA: 0s - loss: 0.0660 - accuracy: 0.9753
Epoch 00008: val_loss did not improve from 0.06550
162/162 [==============================] - 65s 401ms/step - loss: 0.0660 - accuracy: 0.9753 - val_loss: 0.0696 - val_accuracy: 0.9
735
Epoch 9/50
162/162 [==============================] - ETA: 0s - loss: 0.0651 - accuracy: 0.9756
Epoch 00009: val_loss did not improve from 0.06550
162/162 [==============================] - 65s 401ms/step - loss: 0.0651 - accuracy: 0.9756 - val_loss: 0.0660 - val_accuracy: 0.9
758
Epoch 10/50
162/162 [==============================] - ETA: 0s - loss: 0.0639 - accuracy: 0.9760
Epoch 00010: val_loss did not improve from 0.06550
162/162 [==============================] - 65s 401ms/step - loss: 0.0639 - accuracy: 0.9760 - val_loss: 0.0734 - val_accuracy: 0.9
741
Epoch 11/50
```

```
162/162 [==============================] - ETA: 0s - loss: 0.0636 - accuracy: 0.9759
Epoch 00011: val_loss did not improve from 0.06550
162/162 [==============================] - 65s 401ms/step - loss: 0.0636 - accuracy: 0.9759 - val_loss: 0.0783 - val_accuracy: 0.9
713
Epoch 12/50
162/162 [==============================] - ETA: 0s - loss: 0.0620 - accuracy: 0.9766
Epoch 00012: ReduceLROnPlateau reducing learning rate to 0.00010000000474974513.

Epoch 00012: val_loss did not improve from 0.06550
162/162 [==============================] - 65s 400ms/step - loss: 0.0620 - accuracy: 0.9766 - val_loss: 0.0762 - val_accuracy: 0.9
732
Epoch 13/50
162/162 [==============================] - ETA: 0s - loss: 0.0566 - accuracy: 0.9785
Epoch 00013: val_loss improved from 0.06550 to 0.06008, saving model to model-skin-lesion-segmentation.h5
162/162 [==============================] - 65s 401ms/step - loss: 0.0566 - accuracy: 0.9785 - val_loss: 0.0601 - val_accuracy: 0.9
776
Epoch 14/50
162/162 [==============================] - ETA: 0s - loss: 0.0549 - accuracy: 0.9791
Epoch 00014: val_loss improved from 0.06008 to 0.05982, saving model to model-skin-lesion-segmentation.h5
162/162 [==============================] - 65s 401ms/step - loss: 0.0549 - accuracy: 0.9791 - val_loss: 0.0598 - val_accuracy: 0.9
777
Epoch 15/50
162/162 [==============================] - ETA: 0s - loss: 0.0551 - accuracy: 0.9791
Epoch 00015: val_loss did not improve from 0.05982
162/162 [==============================] - 65s 402ms/step - loss: 0.0551 - accuracy: 0.9791 - val_loss: 0.0609 - val_accuracy: 0.9
775
Epoch 16/50
162/162 [==============================] - ETA: 0s - loss: 0.0543 - accuracy: 0.9794
Epoch 00016: val_loss did not improve from 0.05982
162/162 [==============================] - 65s 400ms/step - loss: 0.0543 - accuracy: 0.9794 - val_loss: 0.0604 - val_accuracy: 0.9
777
Epoch 17/50
162/162 [==============================] - ETA: 0s - loss: 0.0541 - accuracy: 0.9794
Epoch 00017: val_loss did not improve from 0.05982
162/162 [==============================] - 65s 401ms/step - loss: 0.0541 - accuracy: 0.9794 - val_loss: 0.0610 - val_accuracy: 0.9
775
Epoch 18/50
162/162 [==============================] - ETA: 0s - loss: 0.0547 - accuracy: 0.9792
Epoch 00018: val_loss did not improve from 0.05982
162/162 [==============================] - 65s 400ms/step - loss: 0.0547 - accuracy: 0.9792 - val_loss: 0.0621 - val_accuracy: 0.9
775
Epoch 19/50
162/162 [==============================] - ETA: 0s - loss: 0.0536 - accuracy: 0.9796
Epoch 00019: ReduceLROnPlateau reducing learning rate to 1.0000000474974514e-05.

Epoch 00019: val_loss did not improve from 0.05982
162/162 [==============================] - 65s 401ms/step - loss: 0.0536 - accuracy: 0.9796 - val_loss: 0.0613 - val_accuracy: 0.9
```

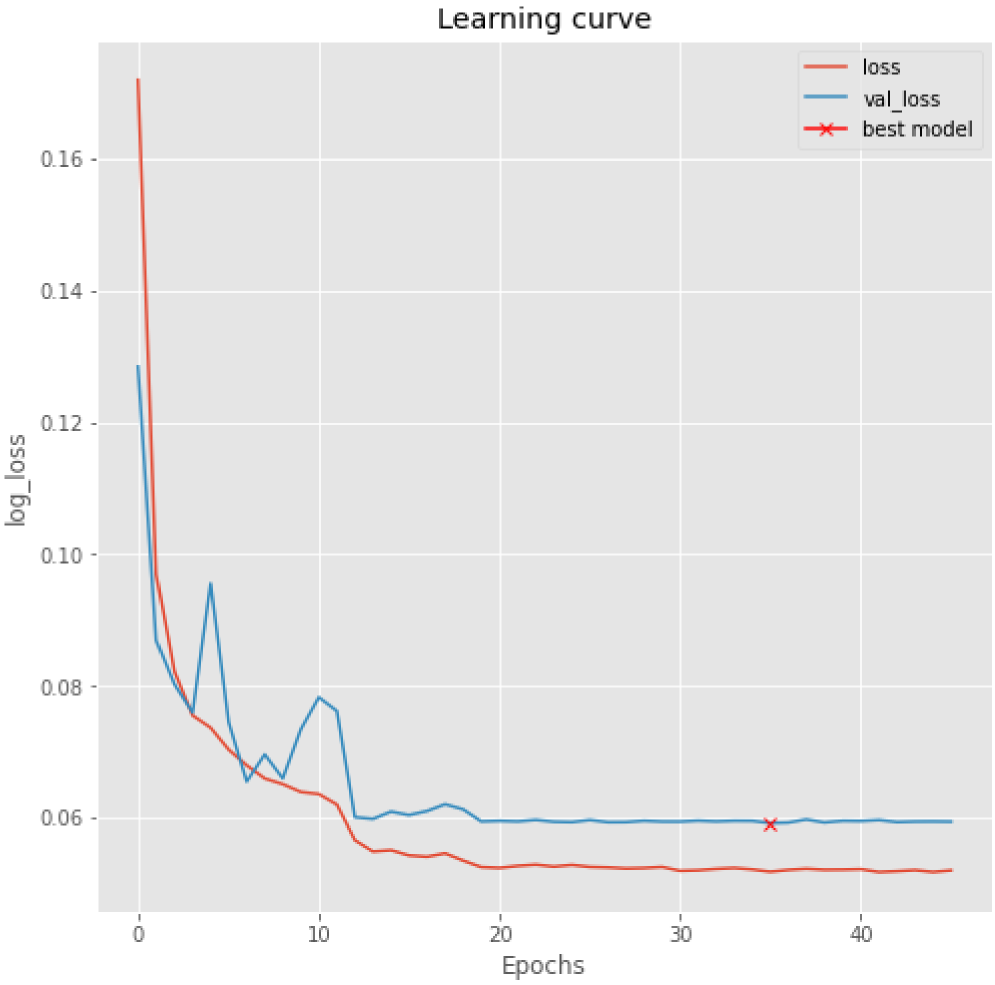
```
776
Epoch 20/50
162/162 [==============================] - ETA: 0s - loss: 0.0526 - accuracy: 0.9800
Epoch 00020: val_loss improved from 0.05982 to 0.05944, saving model to model-skin-lesion-segmentation.h5
162/162 [==============================] - 65s 401ms/step - loss: 0.0526 - accuracy: 0.9800 - val_loss: 0.0594 - val_accuracy: 0.9
780
Epoch 21/50
162/162 [==============================] - ETA: 0s - loss: 0.0524 - accuracy: 0.9800
Epoch 00021: val_loss did not improve from 0.05944
162/162 [==============================] - 65s 401ms/step - loss: 0.0524 - accuracy: 0.9800 - val_loss: 0.0595 - val_accuracy: 0.9
780
Epoch 22/50
162/162 [==============================] - ETA: 0s - loss: 0.0528 - accuracy: 0.9799
Epoch 00022: val_loss improved from 0.05944 to 0.05944, saving model to model-skin-lesion-segmentation.h5
162/162 [==============================] - 65s 401ms/step - loss: 0.0528 - accuracy: 0.9799 - val_loss: 0.0594 - val_accuracy: 0.9
781
Epoch 23/50
162/162 [==============================] - ETA: 0s - loss: 0.0529 - accuracy: 0.9798
Epoch 00023: val_loss did not improve from 0.05944
162/162 [==============================] - 65s 400ms/step - loss: 0.0529 - accuracy: 0.9798 - val_loss: 0.0597 - val_accuracy: 0.9
780
Epoch 24/50
162/162 [==============================] - ETA: 0s - loss: 0.0526 - accuracy: 0.9799
Epoch 00024: val_loss improved from 0.05944 to 0.05939, saving model to model-skin-lesion-segmentation.h5
162/162 [==============================] - 65s 402ms/step - loss: 0.0526 - accuracy: 0.9799 - val_loss: 0.0594 - val_accuracy: 0.9
781
Epoch 25/50
162/162 [==============================] - ETA: 0s - loss: 0.0529 - accuracy: 0.9799
Epoch 00025: val_loss improved from 0.05939 to 0.05933, saving model to model-skin-lesion-segmentation.h5
162/162 [==============================] - 65s 402ms/step - loss: 0.0529 - accuracy: 0.9799 - val_loss: 0.0593 - val_accuracy: 0.9
781
Epoch 26/50
162/162 [==============================] - ETA: 0s - loss: 0.0526 - accuracy: 0.9800
Epoch 00026: val_loss did not improve from 0.05933
162/162 [==============================] - 65s 401ms/step - loss: 0.0526 - accuracy: 0.9800 - val_loss: 0.0597 - val_accuracy: 0.9
780
Epoch 27/50
162/162 [==============================] - ETA: 0s - loss: 0.0525 - accuracy: 0.9800
Epoch 00027: val_loss improved from 0.05933 to 0.05932, saving model to model-skin-lesion-segmentation.h5
162/162 [==============================] - 65s 401ms/step - loss: 0.0525 - accuracy: 0.9800 - val_loss: 0.0593 - val_accuracy: 0.9
781
Epoch 28/50
162/162 [==============================] - ETA: 0s - loss: 0.0524 - accuracy: 0.9800
Epoch 00028: val_loss did not improve from 0.05932
162/162 [==============================] - 65s 401ms/step - loss: 0.0524 - accuracy: 0.9800 - val_loss: 0.0593 - val_accuracy: 0.9
781
Epoch 29/50
```

```
162/162 [==============================] - ETA: 0s - loss: 0.0524 - accuracy: 0.9800
Epoch 00029: val_loss did not improve from 0.05932
162/162 [==============================] - 65s 401ms/step - loss: 0.0524 - accuracy: 0.9800 - val_loss: 0.0595 - val_accuracy: 0.9
780
Epoch 30/50
162/162 [==============================] - ETA: 0s - loss: 0.0526 - accuracy: 0.9799
Epoch 00030: ReduceLROnPlateau reducing learning rate to 1e-05.

Epoch 00030: val_loss did not improve from 0.05932
162/162 [==============================] - 65s 401ms/step - loss: 0.0526 - accuracy: 0.9799 - val_loss: 0.0594 - val_accuracy: 0.9
781
Epoch 31/50
162/162 [==============================] - ETA: 0s - loss: 0.0520 - accuracy: 0.9801
Epoch 00031: val_loss did not improve from 0.05932
162/162 [==============================] - 65s 401ms/step - loss: 0.0520 - accuracy: 0.9801 - val_loss: 0.0594 - val_accuracy: 0.9
781
Epoch 32/50
162/162 [==============================] - ETA: 0s - loss: 0.0521 - accuracy: 0.9801
Epoch 00032: val_loss did not improve from 0.05932
162/162 [==============================] - 65s 402ms/step - loss: 0.0521 - accuracy: 0.9801 - val_loss: 0.0596 - val_accuracy: 0.9
780
Epoch 33/50
162/162 [==============================] - ETA: 0s - loss: 0.0523 - accuracy: 0.9801
Epoch 00033: val_loss did not improve from 0.05932
162/162 [==============================] - 65s 402ms/step - loss: 0.0523 - accuracy: 0.9801 - val_loss: 0.0594 - val_accuracy: 0.9
781
Epoch 34/50
162/162 [==============================] - ETA: 0s - loss: 0.0524 - accuracy: 0.9801
Epoch 00034: val_loss did not improve from 0.05932
162/162 [==============================] - 65s 402ms/step - loss: 0.0524 - accuracy: 0.9801 - val_loss: 0.0595 - val_accuracy: 0.9
780
Epoch 35/50
162/162 [==============================] - ETA: 0s - loss: 0.0522 - accuracy: 0.9801
Epoch 00035: val_loss did not improve from 0.05932
162/162 [==============================] - 65s 402ms/step - loss: 0.0522 - accuracy: 0.9801 - val_loss: 0.0595 - val_accuracy: 0.9
780
Epoch 36/50
162/162 [==============================] - ETA: 0s - loss: 0.0519 - accuracy: 0.9802
Epoch 00036: val_loss improved from 0.05932 to 0.05919, saving model to model-skin-lesion-segmentation.h5
162/162 [==============================] - 65s 402ms/step - loss: 0.0519 - accuracy: 0.9802 - val_loss: 0.0592 - val_accuracy: 0.9
781
Epoch 37/50
162/162 [==============================] - ETA: 0s - loss: 0.0521 - accuracy: 0.9801
Epoch 00037: val_loss did not improve from 0.05919
162/162 [==============================] - 65s 401ms/step - loss: 0.0521 - accuracy: 0.9801 - val_loss: 0.0593 - val_accuracy: 0.9
781
Epoch 38/50
```

```
162/162 [==============================] - ETA: 0s - loss: 0.0523 - accuracy: 0.9801
Epoch 00038: val_loss did not improve from 0.05919
162/162 [==============================] - 65s 401ms/step - loss: 0.0523 - accuracy: 0.9801 - val_loss: 0.0597 - val_accuracy: 0.9
780
Epoch 39/50
162/162 [==============================] - ETA: 0s - loss: 0.0522 - accuracy: 0.9802
Epoch 00039: val_loss did not improve from 0.05919
162/162 [==============================] - 65s 401ms/step - loss: 0.0522 - accuracy: 0.9802 - val_loss: 0.0593 - val_accuracy: 0.9
781
Epoch 40/50
162/162 [==============================] - ETA: 0s - loss: 0.0522 - accuracy: 0.9802
Epoch 00040: val_loss did not improve from 0.05919
162/162 [==============================] - 65s 401ms/step - loss: 0.0522 - accuracy: 0.9802 - val_loss: 0.0595 - val_accuracy: 0.9
781
Epoch 41/50
162/162 [==============================] - ETA: 0s - loss: 0.0523 - accuracy: 0.9801
Epoch 00041: val_loss did not improve from 0.05919
162/162 [==============================] - 65s 401ms/step - loss: 0.0523 - accuracy: 0.9801 - val_loss: 0.0595 - val_accuracy: 0.9
781
Epoch 42/50
162/162 [==============================] - ETA: 0s - loss: 0.0518 - accuracy: 0.9802
Epoch 00042: val_loss did not improve from 0.05919
162/162 [==============================] - 65s 401ms/step - loss: 0.0518 - accuracy: 0.9802 - val_loss: 0.0596 - val_accuracy: 0.9
780
Epoch 43/50
162/162 [==============================] - ETA: 0s - loss: 0.0519 - accuracy: 0.9802
Epoch 00043: val_loss did not improve from 0.05919
162/162 [==============================] - 65s 401ms/step - loss: 0.0519 - accuracy: 0.9802 - val_loss: 0.0593 - val_accuracy: 0.9
781
Epoch 44/50
162/162 [==============================] - ETA: 0s - loss: 0.0521 - accuracy: 0.9802
Epoch 00044: val_loss did not improve from 0.05919
162/162 [==============================] - 65s 401ms/step - loss: 0.0521 - accuracy: 0.9802 - val_loss: 0.0594 - val_accuracy: 0.9
781
Epoch 45/50
162/162 [==============================] - ETA: 0s - loss: 0.0518 - accuracy: 0.9802
Epoch 00045: val_loss did not improve from 0.05919
162/162 [==============================] - 65s 401ms/step - loss: 0.0518 - accuracy: 0.9802 - val_loss: 0.0594 - val_accuracy: 0.9
781
Epoch 46/50
162/162 [==============================] - ETA: 0s - loss: 0.0521 - accuracy: 0.9801
Epoch 00046: val_loss did not improve from 0.05919
162/162 [==============================] - 65s 401ms/step - loss: 0.0521 - accuracy: 0.9801 - val_loss: 0.0594 - val_accuracy: 0.9
781
Epoch 00046: early stopping
```

In [ ]:

```
plt.figure(figsize=(8, 8))
plt.title("Learning curve")
plt.plot(results.history["loss"], label="loss")
plt.plot(results.history["val_loss"], label="val_loss")
plt.plot( np.argmin(results.history["val_loss"]), np.min(results.history["val_loss"]), marker="x", color="r", label="best model")
plt.xlabel("Epochs")
plt.ylabel("log_loss")
plt.legend();
```



Model Validation on the test data

In [ ]:

```
# load the best model after training for testing on test dataset
model.load_weights('model-skin-lesion-segmentation.h5')
```

In [ ]:

```
fid = drive.ListFile({'q':"title='PH2.rar'"}).GetList()[0]['id']
f = drive.CreateFile({'id': fid})
f.GetContentFile('PH2.rar')
```

In [ ]:

```
#!pip install unrar
!unrar x PH2
```

In [ ]:

```
Test_Path='/content/BWT/'
Test_ids = next(os.walk(Test_Path))[2]
Test_ids = np.sort(Test_ids)

test = os.listdir(Test_Path)
for item in test:
    if item.endswith(".ini"):
        os.remove(os.path.join(Test_Path, item))

Test_Mask_Path='/content/MASK/'
Test_Mask_ids = next(os.walk(Test_Mask_Path))[2]
Test_Mask_ids = np.sort(Test_Mask_ids)

print("No. of Test images = ", len(Test_ids))
print("No. of Test Mask images = ", len(Test_Mask_ids))
#test_ids = next(os.walk(TEST_PATH))[2]
#train_ids = np.delete(train_ids,(0),axis=0)
X = np.zeros((len(Test_ids),IMG_HEIGHT, IMG_WIDTH,IMG_CHANNELS),dtype=np.uint8)
Y = np.zeros((len(Test_ids),IMG_HEIGHT, IMG_WIDTH, 1), dtype=np.bool)
```

```
No. of Test images =  200
No. of Test Mask images =  200
```

In [ ]:

```
for n, id_ in tqdm_notebook(enumerate(Test_ids), total=len(Test_ids)):
    path = Test_Path + id_
    img = imread(path)[:,:,:IMG_CHANNELS]
    x_img = resize(img, (IMG_HEIGHT, IMG_WIDTH, 3), mode = 'constant', preserve_range = True)
    X[n] = x_img

# tqdm is used to display the progress bar
```

```
for n1, id1_ in tqdm_notebook(enumerate(Test_Mask_ids), total=len(Test_Mask_ids)):
     mask = img_to_array(load_img("/content/MASK/"+id1_, grayscale=True))
     mask = resize(mask, (IMG_HEIGHT, IMG_WIDTH, 1), mode = 'constant', preserve_range = True)
     Y[n1] = mask
```

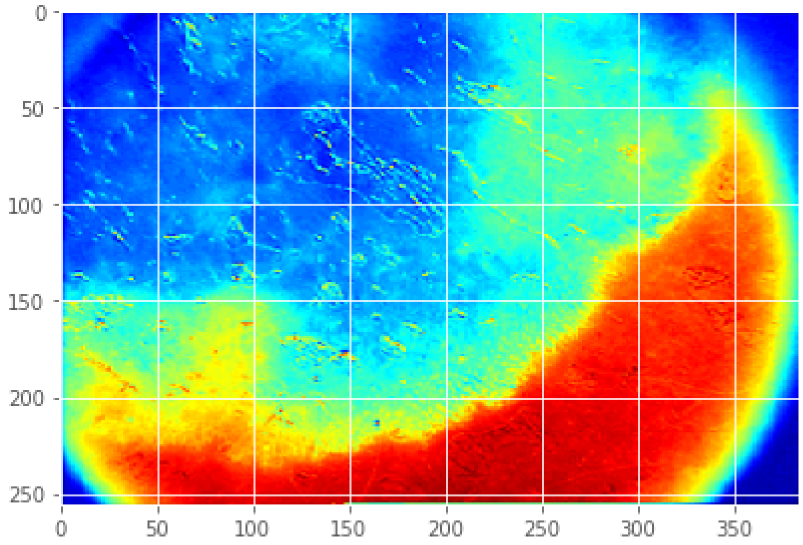
```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:1: TqdmDeprecationWarning: This function will be removed in tqdm==5.
0.0
Please use `tqdm.notebook.tqdm` instead of `tqdm.tqdm_notebook`
  """Entry point for launching an IPython kernel.
```

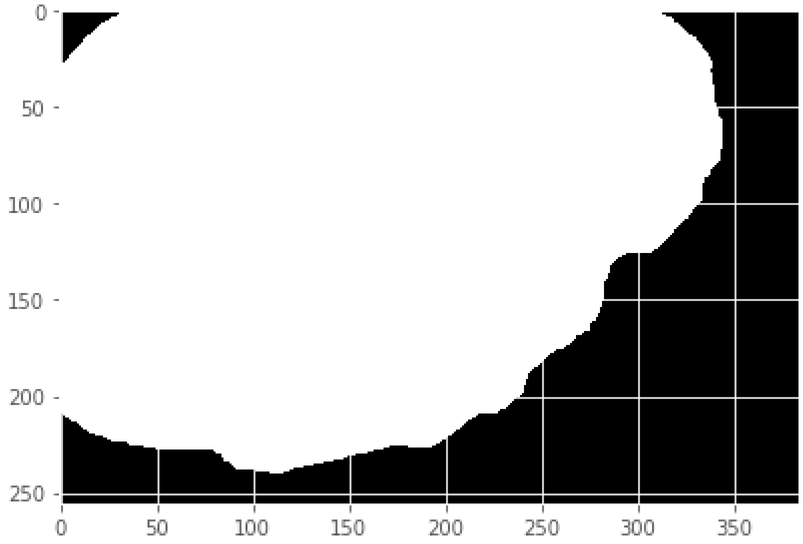
```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:8: TqdmDeprecationWarning: This function will be removed in tqdm==5.
0.0
Please use `tqdm.notebook.tqdm` instead of `tqdm.tqdm_notebook`
```

```
/usr/local/lib/python3.6/dist-packages/keras_preprocessing/image/utils.py:107: UserWarning: grayscale is deprecated. Please use co
lor_mode = "grayscale"
  warnings.warn('grayscale is deprecated. Please use '
```

In [ ]:

```
image_x = random.randint(0, len(Test_ids))
imshow(X[image_x])
plt.show()
imshow(np.squeeze(Y[image_x]))
plt.show()
```





In [ ]:

```
# Evaluate on validation set (this must be equals to the best log_loss)
model.evaluate(X, Y, verbose=1)
```

```
7/7 [==============================] - 25s 4s/step - loss: 0.1875 - accuracy: 0.9400
```

Out[ ]:

```
[0.187486469745636, 0.9400370121002197]
```

In [ ]:

```
# Predict on test dataset
predicted = model.predict(X, verbose=1)
predicted = (predicted > 0.5).astype(np.bool)
```

```
7/7 [==============================] - 25s 4s/step
```

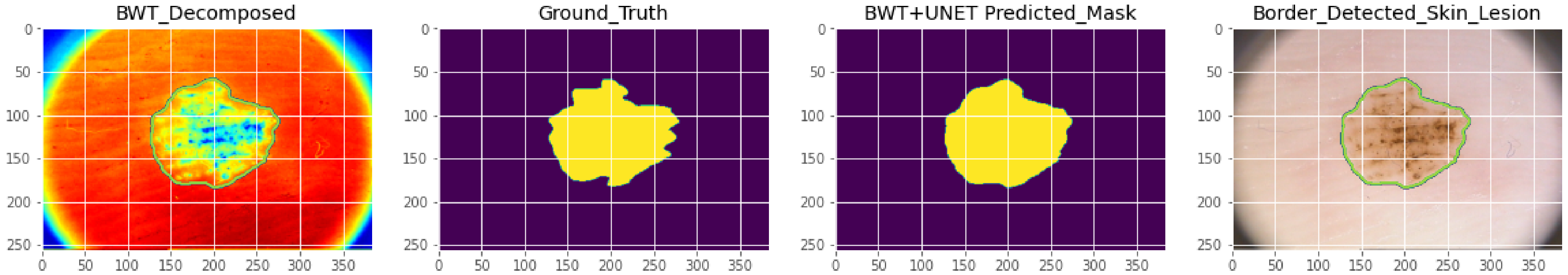
In [ ]:

```
def plot_sample(X, y, preds, X1, ix=None):
    """Function to plot the results"""
    if ix is None:
        ix = random.randint(0, len(X))

    has_mask = y[ix].max() > 0

    fig, ax = plt.subplots(1, 4, figsize=(20, 10))
    ax[0].imshow(X[ix])
    if has_mask:
        ax[0].contour(np.squeeze(preds[ix]))
    ax[0].set_title('BWT_Decomposed')

    ax[1].imshow(y[ix].squeeze())
    ax[1].set_title('Ground_Truth')

    ax[2].imshow(preds[ix].squeeze())
   # if has_mask:
    #    ax[2].contour(y[ix].squeeze())
    ax[2].set_title('BWT+UNET Predicted_Mask')

    ax[3].imshow(img)
    ax[3].contour(np.squeeze(preds[ix]))
    ax[3].set_title('Border_Detected_Skin_Lesion')
```

Segmentation results

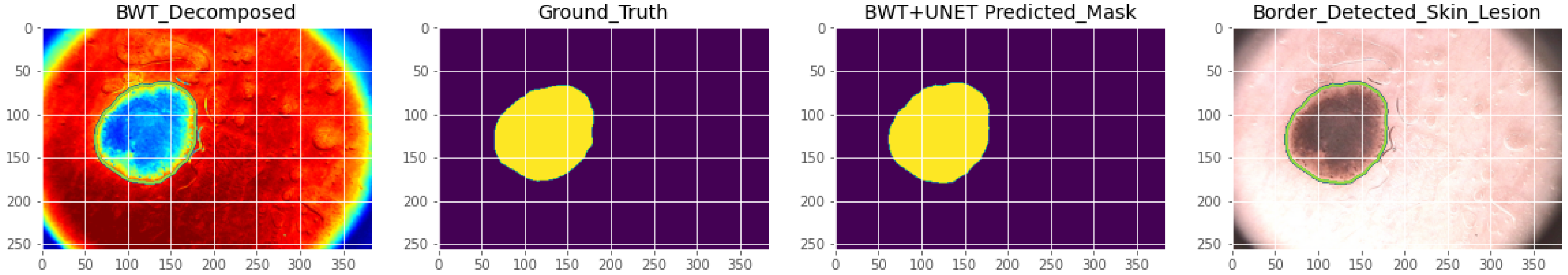
In [ ]:

```
# Results on
plot_sample(X, Y, predicted, X1, ix= 4)
```



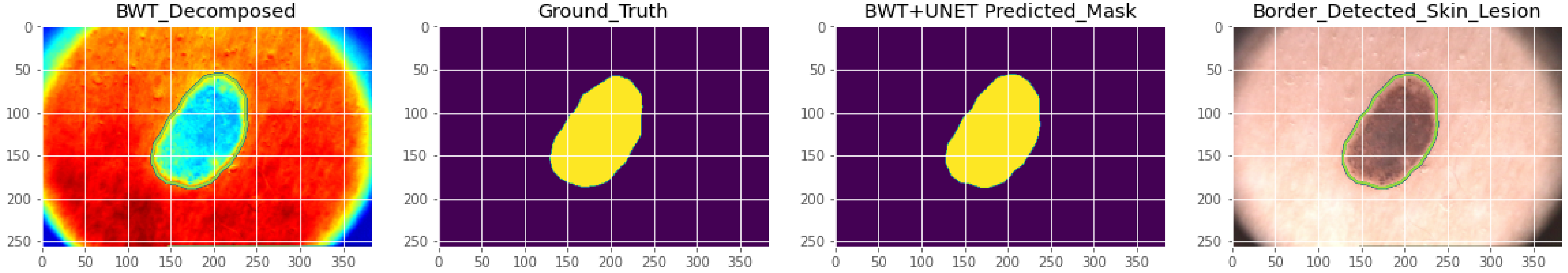
In [ ]:

```
plot_sample(X, Y, predicted, X1, ix= 13)
```



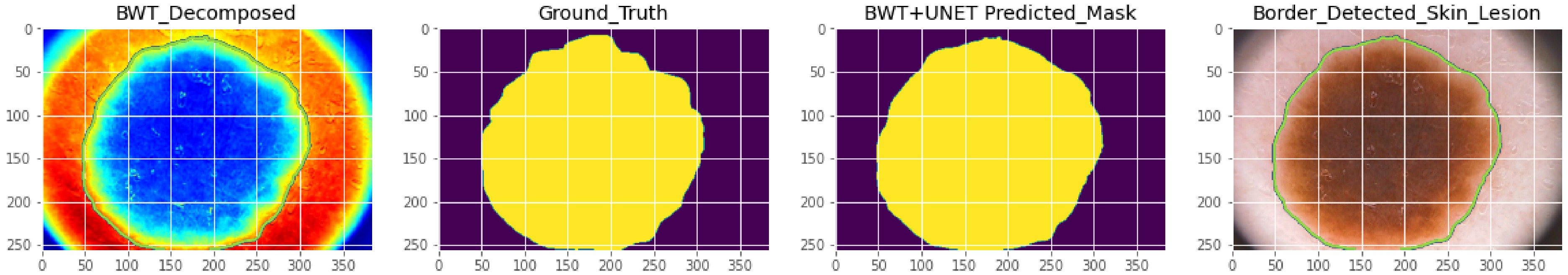
In [ ]:

```
plot_sample(X, Y, predicted, X1, ix= 12)
```



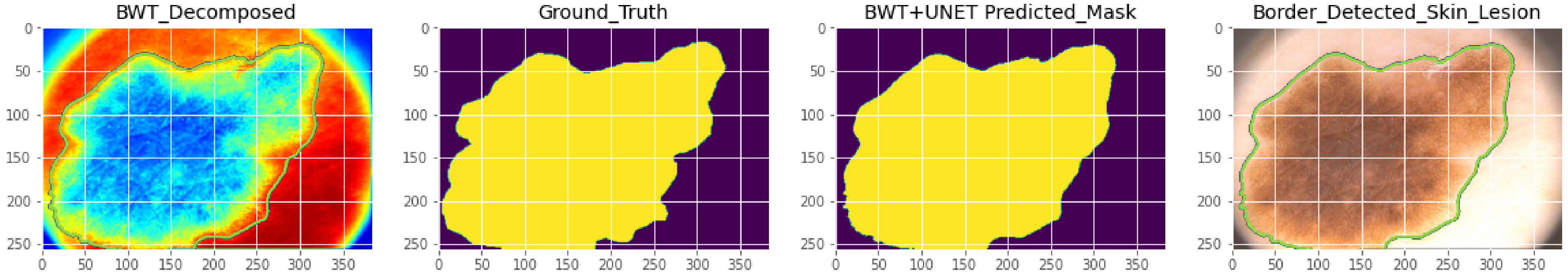
In [ ]:

```
plot_sample(X, Y, predicted, X1, ix= 11)
```



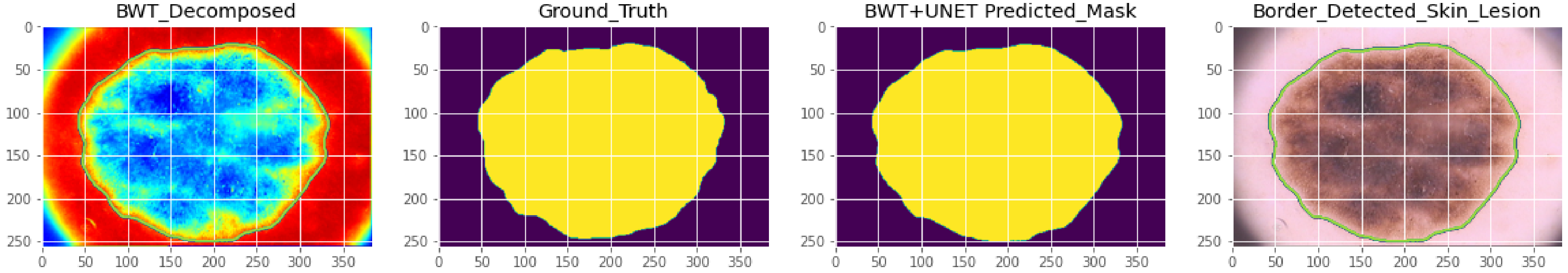
In [ ]:

```
plot_sample(X, Y, predicted, X1, ix= 198)
```



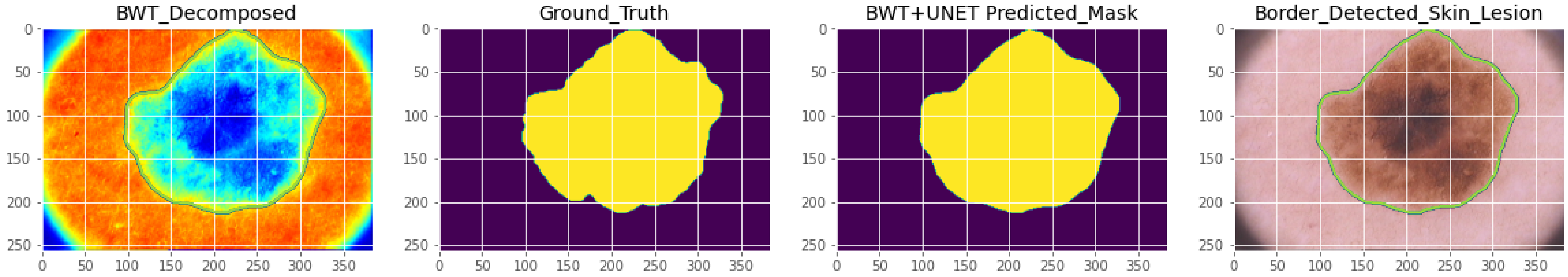
In [ ]:

```
plot_sample(X, Y, predicted, X1, ix= 199)
```



In [ ]:

```
plot_sample(X, Y, predicted, X1, ix= 190)
```


BWT_Decomposed
Ground_Truth
BWT+UNET Predicted_Mask
Border_Detected_Skin_Lesion
0
50
100
150
200
250
0
50
100
150
200
250
300
350