In [1]:

```
%load_ext autoreload
%autoreload 2
```

In [2]:

```
#%autoreload # When utils.py is updated
from utils_unet_resunet import *
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from model.models import Model_3
from model.losses import WBCE
root_path = 'imgs/'
```

In [3]:

```
# Define data type (L8-Landsat8, S2-Sentinel2, S1-Sentinel1)
img_type = 'S2'

if img_type == 'L8':
    # Load images
    ref_2019 = load_tif_image(root_path+'New_Images/References/res_10m/r10m_def_2019
    opt_2018 = load_tif_image(root_path+'New_Images/Landsat8/'+'cut_land8_2018.tif')
    opt_2019 = load_tif_image(root_path+'New_Images/Landsat8/'+'cut_land8_2019.tif')

    # Resize images
    opt_2018 = resize_image(opt_2018.copy(), ref_2019.shape[0], ref_2019.shape[1])
    opt_2019 = resize_image(opt_2019.copy(), ref_2019.shape[0], ref_2019.shape[1])

    # Filter outliers
    opt_2018 = filter_outliers(opt_2018.copy())
    opt_2019 = filter_outliers(opt_2019.copy())

    image_stack = np.concatenate((opt_2018, opt_2019), axis=-1)
    print('landsat_resize:', image_stack.shape)
    del opt_2018, opt_2019

if img_type == 'S2':
    # Load images
    sent2_2018_1 = load_tif_image(root_path+'New_Images/Sentinel2/'+'2018_10m_b2348.
    #sent2_2018_2 = load_tif_image(root_path+'New_Images/Sentinel2/'+'2018_20m_b5678

    # Resize bands of 20m
    #sent2_2018_2 = resize_image(sent2_2018_2.copy(), sent2_2018_1.shape[0], sent2_2
    #sent2_2018 = np.concatenate((sent2_2018_1, sent2_2018_2), axis=-1)
    sent2_2018 = sent2_2018_1.copy()
    del sent2_2018_1#, sent2_2018_2

    sent2_2019_1 = load_tif_image(root_path+'New_Images/Sentinel2/'+'2019_10m_b2348.
    #sent2_2019_2 = load_tif_image(root_path+'New_Images/Sentinel2/'+'2019_20m_b5678

    # Resize bands of 20m
    #sent2_2019_2 = resize_image(sent2_2019_2.copy(), sent2_2019_1.shape[0], sent2_2
    #sent2_2019 = np.concatenate((sent2_2019_1, sent2_2019_2), axis=-1)
    sent2_2019 = sent2_2019_1.copy()
    del sent2_2019_1#, sent2_2019_2

    # Filter outliers
    sent2_2018 = filter_outliers(sent2_2018.copy())
    sent2_2019 = filter_outliers(sent2_2019.copy())

    image_stack = np.concatenate((sent2_2018, sent2_2019), axis=-1)
    print('Image stack:', image_stack.shape)
    del sent2_2018, sent2_2019

if img_type == 'S1':
```

```
    # Load images
    sar_2018_vh = np.expand_dims(load_SAR_image(root_path+'New_Images/Sentinel1/'+'c
    sar_2018_vv = np.expand_dims(load_SAR_image(root_path+'New_Images/Sentinel1/'+'c
    sar_2019_vh = np.expand_dims(load_SAR_image(root_path+'New_Images/Sentinel1/'+'c
    sar_2019_vv = np.expand_dims(load_SAR_image(root_path+'New_Images/Sentinel1/'+'c

    sar_2018 = np.concatenate((sar_2018_vh, sar_2018_vv), axis=-1)
    sar_2019 = np.concatenate((sar_2019_vh, sar_2019_vv), axis=-1)
    del sar_2018_vh, sar_2018_vv, sar_2019_vh, sar_2019_vv

    # Filter outliers
    sar_2018 = filter_outliers(sar_2018.copy())
    sar_2019 = filter_outliers(sar_2019.copy())

    image_stack = np.concatenate((sar_2018, sar_2019), axis=-1)
    print('Image stack:', image_stack.shape)
    del sar_2018, sar_2019

# Load references
# Load current reference
#ref_2019 = load_tif_image(root_path+'New_Images/References/res_10m/r10m_def_2019.ti
# Load past references
#past_ref = np.load(root_path+'New_Images/References/past_ref_and_clouds.npy').astyp
#past_ref1 = load_tif_image(root_path+'New_Images/References/res_10m/r10m_def_1988_2
#past_ref2 = load_tif_image(root_path+'New_Images/References/res_10m/r10m_def_2008_2
#clouds_2018 = load_tif_image(root_path+'New_Images/References/cut_b10_2018.tif').as
#clouds_2018 = resize_image(np.expand_dims(clouds_2018.copy(), axis = -1), ref_2019.
#clouds_2018 = binary_mask_cloud(clouds_2018.copy(), 50)
#clouds_2019 = load_tif_image(root_path+'New_Images/References/cut_b10_2019.tif').as
#clouds_2019 = resize_image(np.expand_dims(clouds_2019.copy(), axis = -1), ref_2019.
#clouds_2019 = binary_mask_cloud(clouds_2019.copy(), 50)
```

```
imgs/New_Images/Sentinel2/2018_10m_b2348.tif
imgs/New_Images/Sentinel2/2019_10m_b2348.tif
Image stack: (17729, 9202, 8)
```

In [4]:

```
# Create label mask
#past_ref = past_ref1 + past_ref2 + clouds_2018 + clouds_2019
#past_ref[past_ref>=1] = 1
#buffer = 2
#final_mask1 = mask_no_considered(ref_2019, buffer, past_ref)
#del past_ref1, past_ref2, clouds_2018, clouds_2019
final_mask1 = np.load(root_path+'New_Images/ref/'+'labels.npy')

lim_x = 10000
lim_y = 7000
image_stack = image_stack[:lim_x, :lim_y, :]
final_mask1 = final_mask1[:lim_x, :lim_y]
#ref_2019 = ref_2019[:lim_x, :lim_y]

h_, w_, channels = image_stack.shape
print('image stack size: ', image_stack.shape)

# Normalization
type_norm = 1
image_array = normalization(image_stack.copy(), type_norm)
print(np.min(image_array), np.max(image_array))
del image_stack

# Print pertengate of each class (whole image)
print('Total no-deforestaion class is {}'.format(len(final_mask1[final_mask1==0])))
print('Total deforestaion class is {}'.format(len(final_mask1[final_mask1==1])))
print('Total past deforestaion class is {}'.format(len(final_mask1[final_mask1==2]))
print('Percentage of deforestaion class is {:.2f}'.format((len(final_mask1[final_mas
```

```
image stack size:  (10000, 7000, 8)
-4.987141 5.626766
Total no-deforestaion class is 36326397
Total deforestaion class is 1048775
Total past deforestaion class is 32624828
Percentage of deforestaion class is 2.89
```

In [5]:

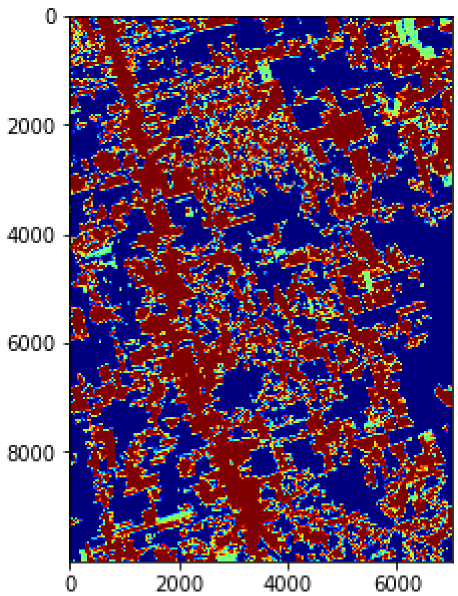
```python
# Create tile mask
mask_tiles = create_mask(final_mask1.shape[0], final_mask1.shape[1], grid_size=(5, 4
image_array = image_array[:mask_tiles.shape[0], :mask_tiles.shape[1],:]
final_mask1 = final_mask1[:mask_tiles.shape[0], :mask_tiles.shape[1]]

print('mask: ',mask_tiles.shape)
print('image stack: ', image_array.shape)
print('ref :', final_mask1.shape)
#plt.imshow(mask_tiles)
```

```
Tiles size:  2000 1750
Mask size:  (10000, 7000)
mask:  (10000, 7000)
image stack:  (10000, 7000, 8)
ref : (10000, 7000)
```

In [6]:

```python
plt.figure(figsize=(10,5))
plt.imshow(final_mask1, cmap = 'jet')
```

Out[6]: <matplotlib.image.AxesImage at 0x1a71e9c71c0>



In [7]:

```python
# Define tiles for training, validation, and test sets
tiles_tr = [1,3,5,8,11,13,14,20]
tiles_val = [6,19]
tiles_ts = (list(set(np.arange(20)+1)-set(tiles_tr)-set(tiles_val)))

mask_tr_val = np.zeros((mask_tiles.shape)).astype('float32')
# Training and validation mask
for tr_ in tiles_tr:
    mask_tr_val[mask_tiles == tr_] = 1

for val_ in tiles_val:
    mask_tr_val[mask_tiles == val_] = 2

mask_amazon_ts = np.zeros((mask_tiles.shape)).astype('float32')
```

```
for ts_ in tiles_ts:
    mask_amazon_ts[mask_tiles == ts_] = 1
```

In [8]:

```
# Create ixd image to extract patches
overlap = 0.7
patch_size = 128
batch_size = 32
im_idx = create_idx_image(final_mask1)
patches_idx = extract_patches(im_idx, patch_size=(patch_size, patch_size), overlap=o
patches_mask = extract_patches(mask_tr_val, patch_size=(patch_size, patch_size), ove
del im_idx
```

In [9]:

```
# Selecting index trn val and test patches idx
idx_trn = np.squeeze(np.where(patches_mask.sum(axis=(1, 2))==patch_size**2))
idx_val = np.squeeze(np.where(patches_mask.sum(axis=(1, 2))==2*patch_size**2))
del patches_mask

patches_idx_trn = patches_idx[idx_trn]
patches_idx_val = patches_idx[idx_val]
del idx_trn, idx_val

print('Number of training patches:  ', len(patches_idx_trn), 'Number of validation p
```

```
Number of training patches:   17110 Number of validation patches 4116
```

In [10]:

```
# Extract patches with at least 2% of deforestation class
X_train = retrieve_idx_percentage(final_mask1, patches_idx_trn, patch_size, pertenta
X_valid = retrieve_idx_percentage(final_mask1, patches_idx_val, patch_size, pertenta
print(X_train.shape, X_valid.shape)
del patches_idx_trn, patches_idx_val
```

```
(1158, 128, 128) (341, 128, 128)
```

In [11]:

```
def batch_generator(batches, image, reference, target_size, number_class):
    """Take as input a Keras ImageGen (Iterator) and generate random
    crops from the image batches generated by the original iterator.
    """
    image = image.reshape(-1, image.shape[-1])
    reference = reference.reshape(final_mask1.shape[0]*final_mask1.shape[1])
    while True:
        batch_x, batch_y = next(batches)
        batch_x = np.squeeze(batch_x.astype('int64'))
        #print(batch_x.shape)
        batch_img = np.zeros((batch_x.shape[0], target_size, target_size, image.shap
        batch_ref = np.zeros((batch_x.shape[0], target_size, target_size, number_cla

        for i in range(batch_x.shape[0]):
            if np.random.rand()>0.5:
                batch_x[i] = np.rot90(batch_x[i], 1)
            batch_img[i] = image[batch_x[i]]
            batch_ref[i] = tf.keras.utils.to_categorical(reference[batch_x[i]] , num

        yield (batch_img, batch_ref)

train_datagen = ImageDataGenerator(horizontal_flip = True,
                                   vertical_flip = True)
valid_datagen = ImageDataGenerator(horizontal_flip = True,
                                   vertical_flip = True)

y_train = np.zeros((len(X_train)))
y_valid = np.zeros((len(X_valid)))
```

```
train_gen = train_datagen.flow(np.expand_dims(X_train, axis = -1), y_train,
                               batch_size=batch_size,
                               shuffle=True)

valid_gen = valid_datagen.flow(np.expand_dims(X_valid, axis = -1), y_valid,
                               batch_size=batch_size,
                               shuffle=False)

number_class = 3
train_gen_crops = batch_generator(train_gen, image_array, final_mask1, patch_size, n
valid_gen_crops = batch_generator(valid_gen, image_array, final_mask1, patch_size, n
```

In [26]:

```
exp = 1
path_exp = root_path+'experiments/exp'+str(exp)
path_models = path_exp+'/models'
path_maps = path_exp+'/pred_maps'

if not os.path.exists(path_exp):
    os.makedirs(path_exp)
if not os.path.exists(path_models):
    os.makedirs(path_models)
if not os.path.exists(path_maps):
    os.makedirs(path_maps)
```

In [32]:

```
# Define model
input_shape = (patch_size, patch_size, channels)
nb_filters = [32, 64, 128]

method = 'unet'
if method == 'unet':
   model = build_unet(input_shape, nb_filters, number_class)

if method == 'resunet':
   model = build_resunet(input_shape, nb_filters, number_class)

#model = Model_3(nb_filters, number_class)
#model.build((None, 128,128,8))
```

In [33]:

```
# Parameters of the model
weights = [0.2, 0.8, 0]
adam = Adam(lr = 1e-3 , beta_1=0.9)

loss = weighted_categorical_crossentropy(weights)
#loss = WBCE(weights)
```

In [34]:

```
time_tr = []
times = 5
for tm in range(0,times):
    print('time: ', tm)
    model.compile(optimizer=adam, loss=loss, metrics=['accuracy'])
    model.summary()

    earlystop = EarlyStopping(monitor='val_loss', min_delta=0.0001, patience=10, ver
    #checkpoint = ModelCheckpoint(path_models+ '/' + method +'_'+str(tm)+'.h5', save
    checkpoint = ModelCheckpoint(path_models+ '/' + method +'_'+str(tm)+'.h5', monit
    lr_reduce = ReduceLROnPlateau(factor=0.9, min_delta=0.0001, patience=5, verbose=
    callbacks_list = [earlystop, checkpoint]
    # train the model
```

```
    start_training = time.time()
    history = model.fit_generator(train_gen_crops,
                                  steps_per_epoch=len(X_train)*3//train_gen.batch_size,
                                  validation_data=valid_gen_crops,
                                  validation_steps=len(X_valid)*3//valid_gen.batch_size,
                                  epochs=100,
                                  callbacks=callbacks_list)
    end_training = time.time() - start_training
    time_tr.append(end_training)
time_tr_array = np.asarray(time_tr)
# Save training time
np.save(path_exp+'/metrics_tr.npy', time_tr_array)
```

```
time:  0
Model: "model_7"
____________________________________________________________________________________________________
______________
Layer (type)                    Output Shape         Param #     Connected to                     
====================================================================================================
==============
input_8 (InputLayer)            [(None, 128, 128, 8) 0                                            
____________________________________________________________________________________________________
______________
conv1 (Conv2D)                  (None, 128, 128, 32) 2336        input_8[0][0]                    
____________________________________________________________________________________________________
______________
max_pooling2d_21 (MaxPooling2D) (None, 64, 64, 32)   0           conv1[0][0]                      
____________________________________________________________________________________________________
______________
conv2 (Conv2D)                  (None, 64, 64, 64)   18496       max_pooling2d_21[0]
[0]                                                                                                 
____________________________________________________________________________________________________
______________
max_pooling2d_22 (MaxPooling2D) (None, 32, 32, 64)   0           conv2[0][0]                      
____________________________________________________________________________________________________
______________
conv3 (Conv2D)                  (None, 32, 32, 128)  73856       max_pooling2d_22[0]
[0]                                                                                                 
____________________________________________________________________________________________________
______________
max_pooling2d_23 (MaxPooling2D) (None, 16, 16, 128)  0           conv3[0][0]                      
____________________________________________________________________________________________________
______________
conv4 (Conv2D)                  (None, 16, 16, 128)  147584      max_pooling2d_23[0]
[0]                                                                                                 
____________________________________________________________________________________________________
______________
conv5 (Conv2D)                  (None, 16, 16, 128)  147584      conv4[0][0]                      
____________________________________________________________________________________________________
______________
conv6 (Conv2D)                  (None, 16, 16, 128)  147584      conv5[0][0]                      
____________________________________________________________________________________________________
______________
up_sampling2d_21 (UpSampling2D) (None, 32, 32, 128)  0           conv6[0][0]                      
____________________________________________________________________________________________________
______________
upsampling3 (Conv2D)            (None, 32, 32, 128)  147584      up_sampling2d_21[0]
[0]                                                                                                 
____________________________________________________________________________________________________
______________
concatenate3 (Concatenate)      (None, 32, 32, 256)  0           conv3[0][0]                      
                                                                 upsampling3[0][0]                
____________________________________________________________________________________________________
______________
up_sampling2d_22 (UpSampling2D) (None, 64, 64, 256)  0           concatenate3[0][0]               
```

```
________________________________________________________________________________
______________
upsampling2 (Conv2D)            (None, 64, 64, 64)   147520      up_sampling2d_22[0]
[0]
________________________________________________________________________________
______________
concatenate2 (Concatenate)      (None, 64, 64, 128)  0           conv2[0][0]
                                                                 upsampling2[0][0]
________________________________________________________________________________
______________
up_sampling2d_23 (UpSampling2D) (None, 128, 128, 128 0           concatenate2[0][0]
________________________________________________________________________________
______________
upsampling1 (Conv2D)            (None, 128, 128, 32) 36896       up_sampling2d_23[0]
[0]
________________________________________________________________________________
______________
concatenate1 (Concatenate)      (None, 128, 128, 64) 0           conv1[0][0]
                                                                 upsampling1[0][0]
________________________________________________________________________________
______________
conv2d_8 (Conv2D)               (None, 128, 128, 3)  195         concatenate1[0][0]
================================================================================
==============
Total params: 869,635
Trainable params: 869,635
Non-trainable params: 0
________________________________________________________________________________
______________
Epoch 1/100
108/108 [==============================] - 18s 166ms/step - loss: 0.1003 - accuracy:
0.7463 - val_loss: 0.0939 - val_accuracy: 0.7857

Epoch 00001: val_loss improved from inf to 0.09390, saving model to imgs/experiment
s/exp1/models\unet_0.h5
Epoch 2/100
108/108 [==============================] - 17s 161ms/step - loss: 0.0666 - accuracy:
0.8225 - val_loss: 0.1026 - val_accuracy: 0.7789

Epoch 00002: val_loss did not improve from 0.09390
Epoch 3/100
108/108 [==============================] - 17s 161ms/step - loss: 0.0621 - accuracy:
0.8334 - val_loss: 0.1251 - val_accuracy: 0.7941

Epoch 00003: val_loss did not improve from 0.09390
Epoch 4/100
108/108 [==============================] - 17s 161ms/step - loss: 0.0590 - accuracy:
0.8367 - val_loss: 0.1607 - val_accuracy: 0.7731

Epoch 00004: val_loss did not improve from 0.09390
Epoch 5/100
108/108 [==============================] - 17s 160ms/step - loss: 0.0539 - accuracy:
0.8485 - val_loss: 0.1381 - val_accuracy: 0.7863

Epoch 00005: val_loss did not improve from 0.09390
Epoch 6/100
108/108 [==============================] - 17s 160ms/step - loss: 0.0506 - accuracy:
0.8556 - val_loss: 0.1500 - val_accuracy: 0.7970

Epoch 00006: val_loss did not improve from 0.09390
Epoch 7/100
108/108 [==============================] - 17s 160ms/step - loss: 0.0466 - accuracy:
0.8623 - val_loss: 0.1455 - val_accuracy: 0.7963

Epoch 00007: val_loss did not improve from 0.09390
```

```
Epoch 8/100
108/108 [==============================] - 17s 162ms/step - loss: 0.0423 - accuracy:
0.8707 - val_loss: 0.1424 - val_accuracy: 0.7970

Epoch 00008: val_loss did not improve from 0.09390
Epoch 9/100
108/108 [==============================] - 17s 159ms/step - loss: 0.0376 - accuracy:
0.8798 - val_loss: 0.1559 - val_accuracy: 0.7967

Epoch 00009: val_loss did not improve from 0.09390
Epoch 10/100
108/108 [==============================] - 18s 164ms/step - loss: 0.0374 - accuracy:
0.8777 - val_loss: 0.1772 - val_accuracy: 0.7991

Epoch 00010: val_loss did not improve from 0.09390
Epoch 11/100
108/108 [==============================] - 18s 168ms/step - loss: 0.0370 - accuracy:
0.8796 - val_loss: 0.1705 - val_accuracy: 0.8036

Epoch 00011: val_loss did not improve from 0.09390
Epoch 00011: early stopping
time:  1
Model: "model_7"
____________________________________________________________________________________
______________
Layer (type)                    Output Shape         Param #     Connected to
====================================================================================
==============
input_8 (InputLayer)            [(None, 128, 128, 8) 0
____________________________________________________________________________________
______________
conv1 (Conv2D)                  (None, 128, 128, 32) 2336        input_8[0][0]
____________________________________________________________________________________
______________
max_pooling2d_21 (MaxPooling2D) (None, 64, 64, 32)   0           conv1[0][0]
____________________________________________________________________________________
______________
conv2 (Conv2D)                  (None, 64, 64, 64)   18496       max_pooling2d_21[0]
[0]
____________________________________________________________________________________
______________
max_pooling2d_22 (MaxPooling2D) (None, 32, 32, 64)   0           conv2[0][0]
____________________________________________________________________________________
______________
conv3 (Conv2D)                  (None, 32, 32, 128)  73856       max_pooling2d_22[0]
[0]
____________________________________________________________________________________
______________
max_pooling2d_23 (MaxPooling2D) (None, 16, 16, 128)  0           conv3[0][0]
____________________________________________________________________________________
______________
conv4 (Conv2D)                  (None, 16, 16, 128)  147584      max_pooling2d_23[0]
[0]
____________________________________________________________________________________
______________
conv5 (Conv2D)                  (None, 16, 16, 128)  147584      conv4[0][0]
____________________________________________________________________________________
______________
conv6 (Conv2D)                  (None, 16, 16, 128)  147584      conv5[0][0]
____________________________________________________________________________________
______________
up_sampling2d_21 (UpSampling2D) (None, 32, 32, 128)  0           conv6[0][0]
____________________________________________________________________________________
______________
upsampling3 (Conv2D)            (None, 32, 32, 128)  147584      up_sampling2d_21[0]
```

```
[0]
__________________________________________________________________________________________________
concatenate3 (Concatenate)      (None, 32, 32, 256)  0           conv3[0][0]
                                                                 upsampling3[0][0]
__________________________________________________________________________________________________
up_sampling2d_22 (UpSampling2D) (None, 64, 64, 256)  0           concatenate3[0][0]
__________________________________________________________________________________________________
upsampling2 (Conv2D)            (None, 64, 64, 64)   147520      up_sampling2d_22[0]
[0]
__________________________________________________________________________________________________
concatenate2 (Concatenate)      (None, 64, 64, 128)  0           conv2[0][0]
                                                                 upsampling2[0][0]
__________________________________________________________________________________________________
up_sampling2d_23 (UpSampling2D) (None, 128, 128, 128 0           concatenate2[0][0]
__________________________________________________________________________________________________
upsampling1 (Conv2D)            (None, 128, 128, 32) 36896       up_sampling2d_23[0]
[0]
__________________________________________________________________________________________________
concatenate1 (Concatenate)      (None, 128, 128, 64) 0           conv1[0][0]
                                                                 upsampling1[0][0]
__________________________________________________________________________________________________
conv2d_8 (Conv2D)               (None, 128, 128, 3)  195         concatenate1[0][0]
==================================================================================================
Total params: 869,635
Trainable params: 869,635
Non-trainable params: 0
__________________________________________________________________________________________________
Epoch 1/100
108/108 [==============================] - 18s 162ms/step - loss: 0.0318 - accuracy:
0.8902 - val_loss: 0.1494 - val_accuracy: 0.7999

Epoch 00001: val_loss improved from inf to 0.14944, saving model to imgs/experiment
s/exp1/models\unet_1.h5
Epoch 2/100
108/108 [==============================] - 17s 157ms/step - loss: 0.0296 - accuracy:
0.8946 - val_loss: 0.1651 - val_accuracy: 0.7938

Epoch 00002: val_loss did not improve from 0.14944
Epoch 3/100
108/108 [==============================] - 17s 159ms/step - loss: 0.0286 - accuracy:
0.8973 - val_loss: 0.1804 - val_accuracy: 0.7880

Epoch 00003: val_loss did not improve from 0.14944
Epoch 4/100
108/108 [==============================] - 17s 161ms/step - loss: 0.0270 - accuracy:
0.9011 - val_loss: 0.2099 - val_accuracy: 0.7895

Epoch 00004: val_loss did not improve from 0.14944
Epoch 5/100
108/108 [==============================] - 17s 157ms/step - loss: 0.0255 - accuracy:
0.9059 - val_loss: 0.2023 - val_accuracy: 0.7971

Epoch 00005: val_loss did not improve from 0.14944
Epoch 6/100
108/108 [==============================] - 17s 160ms/step - loss: 0.0252 - accuracy:
```

```
0.9061 - val_loss: 0.2033 - val_accuracy: 0.7890

Epoch 00006: val_loss did not improve from 0.14944
Epoch 7/100
108/108 [==============================] - 18s 171ms/step - loss: 0.0239 - accuracy:
0.9103 - val_loss: 0.2182 - val_accuracy: 0.7835

Epoch 00007: val_loss did not improve from 0.14944
Epoch 8/100
108/108 [==============================] - 17s 161ms/step - loss: 0.0228 - accuracy:
0.9139 - val_loss: 0.2200 - val_accuracy: 0.7898

Epoch 00008: val_loss did not improve from 0.14944
Epoch 9/100
108/108 [==============================] - 17s 160ms/step - loss: 0.0228 - accuracy:
0.9143 - val_loss: 0.1793 - val_accuracy: 0.7855

Epoch 00009: val_loss did not improve from 0.14944
Epoch 10/100
108/108 [==============================] - 17s 162ms/step - loss: 0.0227 - accuracy:
0.9142 - val_loss: 0.1980 - val_accuracy: 0.7855

Epoch 00010: val_loss did not improve from 0.14944
Epoch 11/100
108/108 [==============================] - 18s 164ms/step - loss: 0.0210 - accuracy:
0.9200 - val_loss: 0.1976 - val_accuracy: 0.7954

Epoch 00011: val_loss did not improve from 0.14944
Epoch 00011: early stopping
time:  2
Model: "model_7"
____________________________________________________________________________________
______________
Layer (type)                    Output Shape         Param #     Connected to
====================================================================================
==============
input_8 (InputLayer)            [(None, 128, 128, 8) 0
____________________________________________________________________________________
______________
conv1 (Conv2D)                  (None, 128, 128, 32) 2336        input_8[0][0]
____________________________________________________________________________________
______________
max_pooling2d_21 (MaxPooling2D) (None, 64, 64, 32)   0           conv1[0][0]
____________________________________________________________________________________
______________
conv2 (Conv2D)                  (None, 64, 64, 64)   18496       max_pooling2d_21[0]
[0]
____________________________________________________________________________________
______________
max_pooling2d_22 (MaxPooling2D) (None, 32, 32, 64)   0           conv2[0][0]
____________________________________________________________________________________
______________
conv3 (Conv2D)                  (None, 32, 32, 128)  73856       max_pooling2d_22[0]
[0]
____________________________________________________________________________________
______________
max_pooling2d_23 (MaxPooling2D) (None, 16, 16, 128)  0           conv3[0][0]
____________________________________________________________________________________
______________
conv4 (Conv2D)                  (None, 16, 16, 128)  147584      max_pooling2d_23[0]
[0]
____________________________________________________________________________________
______________
conv5 (Conv2D)                  (None, 16, 16, 128)  147584      conv4[0][0]
____________________________________________________________________________________
```

```
______________
conv6 (Conv2D)                 (None, 16, 16, 128)  147584      conv5[0][0]
__________________________________________________________________________________________________
up_sampling2d_21 (UpSampling2D) (None, 32, 32, 128)  0          conv6[0][0]
__________________________________________________________________________________________________
upsampling3 (Conv2D)           (None, 32, 32, 128)  147584      up_sampling2d_21[0]
[0]
__________________________________________________________________________________________________
concatenate3 (Concatenate)     (None, 32, 32, 256)  0           conv3[0][0]
                                                                upsampling3[0][0]
__________________________________________________________________________________________________
up_sampling2d_22 (UpSampling2D) (None, 64, 64, 256)  0          concatenate3[0][0]
__________________________________________________________________________________________________
upsampling2 (Conv2D)           (None, 64, 64, 64)   147520      up_sampling2d_22[0]
[0]
__________________________________________________________________________________________________
concatenate2 (Concatenate)     (None, 64, 64, 128)  0           conv2[0][0]
                                                                upsampling2[0][0]
__________________________________________________________________________________________________
up_sampling2d_23 (UpSampling2D) (None, 128, 128, 128 0          concatenate2[0][0]
__________________________________________________________________________________________________
upsampling1 (Conv2D)           (None, 128, 128, 32) 36896       up_sampling2d_23[0]
[0]
__________________________________________________________________________________________________
concatenate1 (Concatenate)     (None, 128, 128, 64) 0           conv1[0][0]
                                                                upsampling1[0][0]
__________________________________________________________________________________________________
conv2d_8 (Conv2D)              (None, 128, 128, 3)  195         concatenate1[0][0]
==================================================================================================
==============
Total params: 869,635
Trainable params: 869,635
Non-trainable params: 0
__________________________________________________________________________________________________
______________
Epoch 1/100
108/108 [==============================] - 18s 165ms/step - loss: 0.0197 - accuracy:
0.9231 - val_loss: 0.1974 - val_accuracy: 0.7882

Epoch 00001: val_loss improved from inf to 0.19741, saving model to imgs/experiment
s/exp1/models\unet_2.h5
Epoch 2/100
108/108 [==============================] - 17s 159ms/step - loss: 0.0186 - accuracy:
0.9272 - val_loss: 0.2131 - val_accuracy: 0.7873

Epoch 00002: val_loss did not improve from 0.19741
Epoch 3/100
108/108 [==============================] - 17s 158ms/step - loss: 0.0181 - accuracy:
0.9288 - val_loss: 0.2495 - val_accuracy: 0.7782

Epoch 00003: val_loss did not improve from 0.19741
Epoch 4/100
108/108 [==============================] - 17s 159ms/step - loss: 0.0176 - accuracy:
0.9312 - val_loss: 0.2474 - val_accuracy: 0.7803
```

```
Epoch 00004: val_loss did not improve from 0.19741
Epoch 5/100
108/108 [==============================] - 17s 161ms/step - loss: 0.0175 - accuracy:
0.9311 - val_loss: 0.2371 - val_accuracy: 0.7872

Epoch 00005: val_loss did not improve from 0.19741
Epoch 6/100
108/108 [==============================] - 17s 158ms/step - loss: 0.0168 - accuracy:
0.9337 - val_loss: 0.2448 - val_accuracy: 0.7804

Epoch 00006: val_loss did not improve from 0.19741
Epoch 7/100
108/108 [==============================] - 17s 160ms/step - loss: 0.0160 - accuracy:
0.9374 - val_loss: 0.2632 - val_accuracy: 0.7825

Epoch 00007: val_loss did not improve from 0.19741
Epoch 8/100
108/108 [==============================] - 17s 161ms/step - loss: 0.0157 - accuracy:
0.9382 - val_loss: 0.2664 - val_accuracy: 0.7788

Epoch 00008: val_loss did not improve from 0.19741
Epoch 9/100
108/108 [==============================] - 17s 160ms/step - loss: 0.0153 - accuracy:
0.9391 - val_loss: 0.2506 - val_accuracy: 0.7847

Epoch 00009: val_loss did not improve from 0.19741
Epoch 10/100
108/108 [==============================] - 17s 163ms/step - loss: 0.0207 - accuracy:
0.9262 - val_loss: 0.2336 - val_accuracy: 0.7871

Epoch 00010: val_loss did not improve from 0.19741
Epoch 11/100
108/108 [==============================] - 17s 160ms/step - loss: 0.0170 - accuracy:
0.9339 - val_loss: 0.2414 - val_accuracy: 0.7804

Epoch 00011: val_loss did not improve from 0.19741
Epoch 00011: early stopping
time:  3
Model: "model_7"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_8 (InputLayer)            [(None, 128, 128, 8) 0
__________________________________________________________________________________________________
conv1 (Conv2D)                  (None, 128, 128, 32) 2336        input_8[0][0]
__________________________________________________________________________________________________
max_pooling2d_21 (MaxPooling2D) (None, 64, 64, 32)   0           conv1[0][0]
__________________________________________________________________________________________________
conv2 (Conv2D)                  (None, 64, 64, 64)   18496       max_pooling2d_21[0]
[0]
__________________________________________________________________________________________________
max_pooling2d_22 (MaxPooling2D) (None, 32, 32, 64)   0           conv2[0][0]
__________________________________________________________________________________________________
conv3 (Conv2D)                  (None, 32, 32, 128)  73856       max_pooling2d_22[0]
[0]
__________________________________________________________________________________________________
max_pooling2d_23 (MaxPooling2D) (None, 16, 16, 128)  0           conv3[0][0]
```

```
__________________________________________________________________________________
conv4 (Conv2D)                  (None, 16, 16, 128)  147584      max_pooling2d_23[0]
[0]
__________________________________________________________________________________
conv5 (Conv2D)                  (None, 16, 16, 128)  147584      conv4[0][0]
__________________________________________________________________________________
conv6 (Conv2D)                  (None, 16, 16, 128)  147584      conv5[0][0]
__________________________________________________________________________________
up_sampling2d_21 (UpSampling2D) (None, 32, 32, 128)  0           conv6[0][0]
__________________________________________________________________________________
upsampling3 (Conv2D)            (None, 32, 32, 128)  147584      up_sampling2d_21[0]
[0]
__________________________________________________________________________________
concatenate3 (Concatenate)      (None, 32, 32, 256)  0           conv3[0][0]
                                                                 upsampling3[0][0]
__________________________________________________________________________________
up_sampling2d_22 (UpSampling2D) (None, 64, 64, 256)  0           concatenate3[0][0]
__________________________________________________________________________________
upsampling2 (Conv2D)            (None, 64, 64, 64)   147520      up_sampling2d_22[0]
[0]
__________________________________________________________________________________
concatenate2 (Concatenate)      (None, 64, 64, 128)  0           conv2[0][0]
                                                                 upsampling2[0][0]
__________________________________________________________________________________
up_sampling2d_23 (UpSampling2D) (None, 128, 128, 128 0           concatenate2[0][0]
__________________________________________________________________________________
upsampling1 (Conv2D)            (None, 128, 128, 32) 36896       up_sampling2d_23[0]
[0]
__________________________________________________________________________________
concatenate1 (Concatenate)      (None, 128, 128, 64) 0           conv1[0][0]
                                                                 upsampling1[0][0]
__________________________________________________________________________________
conv2d_8 (Conv2D)               (None, 128, 128, 3)  195         concatenate1[0][0]
==================================================================================================
Total params: 869,635
Trainable params: 869,635
Non-trainable params: 0
__________________________________________________________________________________
Epoch 1/100
108/108 [==============================] - 18s 159ms/step - loss: 0.0145 - accuracy:
0.9427 - val_loss: 0.2176 - val_accuracy: 0.7829

Epoch 00001: val_loss improved from inf to 0.21760, saving model to imgs/experiment
s/exp1/models\unet_3.h5
Epoch 2/100
108/108 [==============================] - 17s 159ms/step - loss: 0.0141 - accuracy:
0.9447 - val_loss: 0.2358 - val_accuracy: 0.7821

Epoch 00002: val_loss did not improve from 0.21760
Epoch 3/100
```

```
108/108 [==============================] - 17s 157ms/step - loss: 0.0139 - accuracy:
0.9444 - val_loss: 0.2713 - val_accuracy: 0.7740

Epoch 00003: val_loss did not improve from 0.21760
Epoch 4/100
108/108 [==============================] - 17s 160ms/step - loss: 0.0137 - accuracy:
0.9452 - val_loss: 0.2757 - val_accuracy: 0.7692

Epoch 00004: val_loss did not improve from 0.21760
Epoch 5/100
108/108 [==============================] - 17s 157ms/step - loss: 0.0137 - accuracy:
0.9451 - val_loss: 0.2651 - val_accuracy: 0.7829

Epoch 00005: val_loss did not improve from 0.21760
Epoch 6/100
108/108 [==============================] - 18s 164ms/step - loss: 0.0147 - accuracy:
0.9418 - val_loss: 0.2751 - val_accuracy: 0.7823

Epoch 00006: val_loss did not improve from 0.21760
Epoch 7/100
108/108 [==============================] - 17s 163ms/step - loss: 0.0128 - accuracy:
0.9494 - val_loss: 0.2878 - val_accuracy: 0.7739

Epoch 00007: val_loss did not improve from 0.21760
Epoch 8/100
108/108 [==============================] - 17s 159ms/step - loss: 0.0131 - accuracy:
0.9475 - val_loss: 0.2755 - val_accuracy: 0.7828

Epoch 00008: val_loss did not improve from 0.21760
Epoch 9/100
108/108 [==============================] - 17s 160ms/step - loss: 0.0128 - accuracy:
0.9492 - val_loss: 0.2735 - val_accuracy: 0.7805

Epoch 00009: val_loss did not improve from 0.21760
Epoch 10/100
108/108 [==============================] - 17s 160ms/step - loss: 0.0126 - accuracy:
0.9501 - val_loss: 0.2779 - val_accuracy: 0.7847

Epoch 00010: val_loss did not improve from 0.21760
Epoch 11/100
108/108 [==============================] - 18s 165ms/step - loss: 0.0127 - accuracy:
0.9493 - val_loss: 0.2624 - val_accuracy: 0.7824

Epoch 00011: val_loss did not improve from 0.21760
Epoch 00011: early stopping
time:  4
Model: "model_7"
____________________________________________________________________________________
______________
Layer (type)                    Output Shape         Param #     Connected to
====================================================================================
==============
input_8 (InputLayer)            [(None, 128, 128, 8) 0
____________________________________________________________________________________
______________
conv1 (Conv2D)                  (None, 128, 128, 32) 2336        input_8[0][0]
____________________________________________________________________________________
______________
max_pooling2d_21 (MaxPooling2D) (None, 64, 64, 32)   0           conv1[0][0]
____________________________________________________________________________________
______________
conv2 (Conv2D)                  (None, 64, 64, 64)   18496       max_pooling2d_21[0]
[0]
____________________________________________________________________________________
______________
```

```
max_pooling2d_22 (MaxPooling2D) (None, 32, 32, 64)   0           conv2[0][0]
__________________________________________________________________________________________________
conv3 (Conv2D)                  (None, 32, 32, 128)  73856       max_pooling2d_22[0][0]
__________________________________________________________________________________________________
max_pooling2d_23 (MaxPooling2D) (None, 16, 16, 128)  0           conv3[0][0]
__________________________________________________________________________________________________
conv4 (Conv2D)                  (None, 16, 16, 128)  147584      max_pooling2d_23[0][0]
__________________________________________________________________________________________________
conv5 (Conv2D)                  (None, 16, 16, 128)  147584      conv4[0][0]
__________________________________________________________________________________________________
conv6 (Conv2D)                  (None, 16, 16, 128)  147584      conv5[0][0]
__________________________________________________________________________________________________
up_sampling2d_21 (UpSampling2D) (None, 32, 32, 128)  0           conv6[0][0]
__________________________________________________________________________________________________
upsampling3 (Conv2D)            (None, 32, 32, 128)  147584      up_sampling2d_21[0][0]
__________________________________________________________________________________________________
concatenate3 (Concatenate)      (None, 32, 32, 256)  0           conv3[0][0]
                                                                 upsampling3[0][0]
__________________________________________________________________________________________________
up_sampling2d_22 (UpSampling2D) (None, 64, 64, 256)  0           concatenate3[0][0]
__________________________________________________________________________________________________
upsampling2 (Conv2D)            (None, 64, 64, 64)   147520      up_sampling2d_22[0][0]
__________________________________________________________________________________________________
concatenate2 (Concatenate)      (None, 64, 64, 128)  0           conv2[0][0]
                                                                 upsampling2[0][0]
__________________________________________________________________________________________________
up_sampling2d_23 (UpSampling2D) (None, 128, 128, 128 0           concatenate2[0][0]
__________________________________________________________________________________________________
upsampling1 (Conv2D)            (None, 128, 128, 32) 36896       up_sampling2d_23[0][0]
__________________________________________________________________________________________________
concatenate1 (Concatenate)      (None, 128, 128, 64) 0           conv1[0][0]
                                                                 upsampling1[0][0]
__________________________________________________________________________________________________
conv2d_8 (Conv2D)               (None, 128, 128, 3)  195         concatenate1[0][0]
==================================================================================================
Total params: 869,635
Trainable params: 869,635
Non-trainable params: 0
__________________________________________________________________________________________________
Epoch 1/100
108/108 [==============================] - 18s 163ms/step - loss: 0.0122 - accuracy: 0.9512 - val_loss: 0.2368 - val_accuracy: 0.7839
```

```
Epoch 00001: val_loss improved from inf to 0.23680, saving model to imgs/experiment
s/exp1/models\unet_4.h5
Epoch 2/100
108/108 [==============================] - 17s 159ms/step - loss: 0.0131 - accuracy:
0.9487 - val_loss: 0.2457 - val_accuracy: 0.7797

Epoch 00002: val_loss did not improve from 0.23680
Epoch 3/100
108/108 [==============================] - 17s 161ms/step - loss: 0.0124 - accuracy:
0.9509 - val_loss: 0.2908 - val_accuracy: 0.7739

Epoch 00003: val_loss did not improve from 0.23680
Epoch 4/100
108/108 [==============================] - 17s 158ms/step - loss: 0.0116 - accuracy:
0.9538 - val_loss: 0.2862 - val_accuracy: 0.7789

Epoch 00004: val_loss did not improve from 0.23680
Epoch 5/100
108/108 [==============================] - 17s 158ms/step - loss: 0.0128 - accuracy:
0.9503 - val_loss: 0.2470 - val_accuracy: 0.7675

Epoch 00005: val_loss did not improve from 0.23680
Epoch 6/100
108/108 [==============================] - 17s 161ms/step - loss: 0.0194 - accuracy:
0.9300 - val_loss: 0.2517 - val_accuracy: 0.7813

Epoch 00006: val_loss did not improve from 0.23680
Epoch 7/100
108/108 [==============================] - 17s 159ms/step - loss: 0.0138 - accuracy:
0.9470 - val_loss: 0.2723 - val_accuracy: 0.7813

Epoch 00007: val_loss did not improve from 0.23680
Epoch 8/100
108/108 [==============================] - 17s 161ms/step - loss: 0.0118 - accuracy:
0.9530 - val_loss: 0.2801 - val_accuracy: 0.7727

Epoch 00008: val_loss did not improve from 0.23680
Epoch 9/100
108/108 [==============================] - 17s 160ms/step - loss: 0.0111 - accuracy:
0.9556 - val_loss: 0.2817 - val_accuracy: 0.7762

Epoch 00009: val_loss did not improve from 0.23680
Epoch 10/100
108/108 [==============================] - 17s 160ms/step - loss: 0.0109 - accuracy:
0.9565 - val_loss: 0.2831 - val_accuracy: 0.7786

Epoch 00010: val_loss did not improve from 0.23680
Epoch 11/100
108/108 [==============================] - 17s 160ms/step - loss: 0.0110 - accuracy:
0.9564 - val_loss: 0.2730 - val_accuracy: 0.7794

Epoch 00011: val_loss did not improve from 0.23680
Epoch 00011: early stopping
```

In [35]:

```python
# Test Loop
time_ts = []
n_pool = 3
n_rows = 5
n_cols = 4
rows, cols = image_array.shape[:2]
pad_rows = rows - np.ceil(rows/(n_rows*2**n_pool))*n_rows*2**n_pool
pad_cols = cols - np.ceil(cols/(n_cols*2**n_pool))*n_cols*2**n_pool
print(pad_rows, pad_cols)
```

```
npad = ((0, int(abs(pad_rows))), (0, int(abs(pad_cols))), (0, 0))
image1_pad = np.pad(image_array, pad_width=npad, mode='reflect')

h, w, c = image1_pad.shape
patch_size_rows = h//n_rows
patch_size_cols = w//n_cols
num_patches_x = int(h/patch_size_rows)
num_patches_y = int(w/patch_size_cols)

input_shape=(patch_size_rows,patch_size_cols, c)

if method == 'unet':
   new_model = build_unet(input_shape, nb_filters, number_class)

if method == 'resunet':
   new_model = build_resunet(input_shape, nb_filters, number_class)

#new_model = Model_3(nb_filters, number_class)
#new_model.build((None, 128,128,8))
#new_model.compile(optimizer=adam, loss=loss, metrics=['accuracy'])

for tm in range(0,times):
    print('time: ', tm)
    model = load_model(path_models+ '/' + method +'_'+str(tm)+'.h5', compile=False)

    for l in range(1, len(model.layers)):
        new_model.layers[l].set_weights(model.layers[l].get_weights())
    #new_model.load_weights(path_models+ '/' + method +'_'+str(tm)+'.h5')

    start_test = time.time()
    patch_t = []

    for i in range(0,num_patches_y):
        for j in range(0,num_patches_x):
            patch = image1_pad[patch_size_rows*j:patch_size_rows*(j+1), patch_size_c
            predictions_ = new_model.predict(np.expand_dims(patch, axis=0))
            del patch
            patch_t.append(predictions_[:,:,:,1])
            del predictions_
    end_test =  time.time() - start_test
    patches_pred = np.asarray(patch_t).astype(np.float32)

    prob_recontructed = pred_reconctruct(h, w, num_patches_x, num_patches_y, patch_s
    np.save(path_maps+'/'+'prob_'+str(tm)+'.npy',prob_recontructed)

    time_ts.append(end_test)
    del prob_recontructed, patches_pred
time_ts_array = np.asarray(time_ts)
del new_model
# Save test time
np.save(path_exp+'/metrics_ts.npy', time_ts_array)
```

```
0.0 -8.0
time:  0
time:  1
time:  2
time:  3
time:  4
```

In [36]:

```
# Compute mean of the tm predictions maps
prob_rec = np.zeros((image1_pad.shape[0],image1_pad.shape[1], times))

for tm in range (0, times):
```

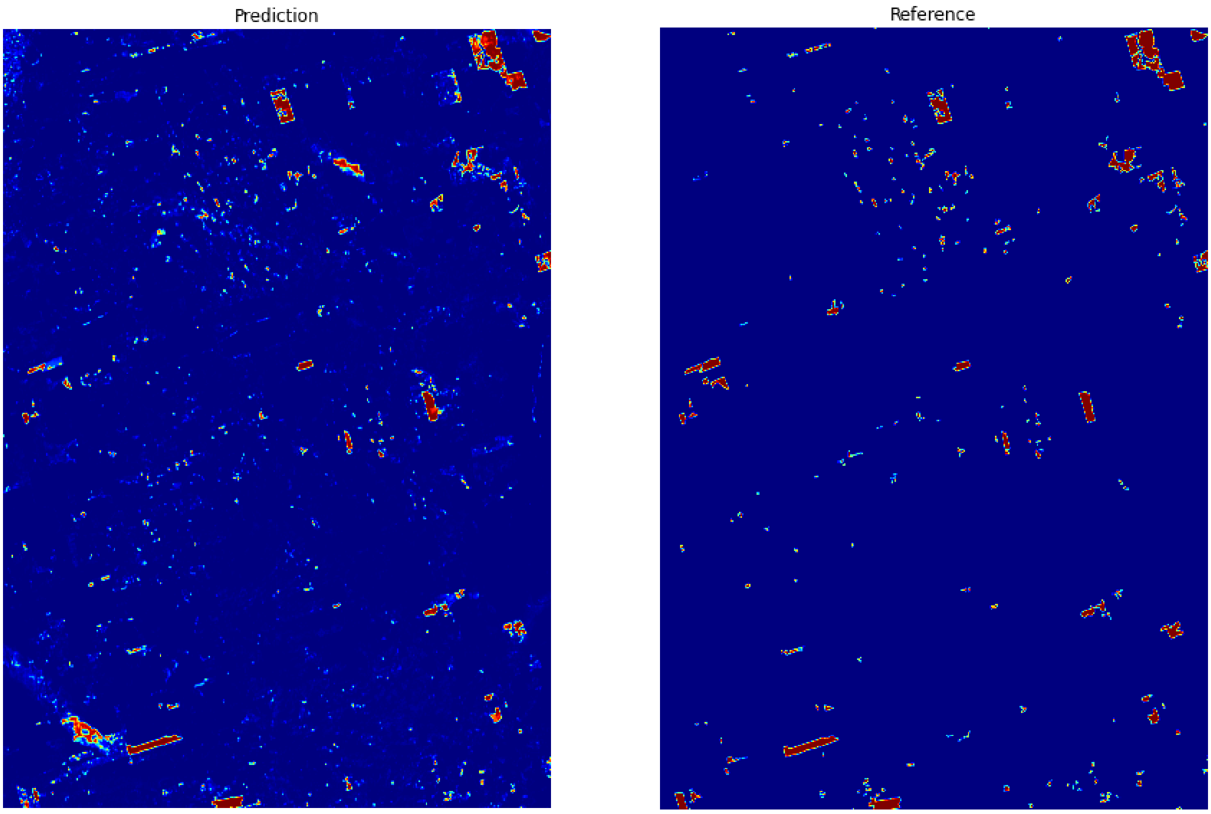
```
    print(tm)
    prob_rec[:,:,tm] = np.load(path_maps+'/'+'prob_'+str(tm)+'.npy').astype(np.float

mean_prob = np.mean(prob_rec, axis = -1)
np.save(path_maps+'/prob_mean.npy', mean_prob)
```

```
0
1
2
3
4
```

In [37]:

```
ref = final_mask1
ref[ref==0]=0
ref[ref==2]=0
# Plot mean map and reference
fig = plt.figure(figsize=(15,10))
ax1 = fig.add_subplot(121)
plt.title('Prediction')
ax1.imshow(mean_prob, cmap ='jet')
ax1.axis('off')

ax2 = fig.add_subplot(122)
plt.title('Reference')
ax2.imshow(ref, cmap ='jet')
ax2.axis('off')
```

Out[37]: (-0.5, 6999.5, 9999.5, -0.5)



In [38]:

```
# Computing metrics
mean_prob = mean_prob[:final_mask1.shape[0], :final_mask1.shape[1]]
ref1 = np.ones_like(final_mask1).astype(np.float32)

ref1 [final_mask1 == 2] = 0
TileMask = mask_amazon_ts * ref1
GTTruePositives = final_mask1==1
```

```
Npoints = 50
Pmax = np.max(mean_prob[GTTruePositives * TileMask ==1])
ProbList = np.linspace(Pmax,0,Npoints)

metrics_ = matrics_AA_recall(ProbList, mean_prob, final_mask1, mask_amazon_ts, 625)
np.save(path_exp+'/acc_metrics.npy',metrics_)
```

```
0.9997152328491211
D:\Ferrari\proj_1\projeto\utils_unet_resunet.py:200: RuntimeWarning: invalid value e
ncountered in longlong_scalars
  precision_ = TP/(TP+FP)
0.9793128811583227
0.9589105294675243
0.9385081777767259
0.9181058260859276
0.8977034743951292
0.8773011227043308
0.8568987710135324
0.836496419322734
0.8160940676319356
0.7956917159411372
0.7752893642503389
0.7548870125595404
0.7344846608687421
0.7140823091779437
0.6936799574871453
0.673277605796346
0.6528752541055485
0.6324729024147501
0.6120705507239517
0.5916681990331534
0.5712658473423549
0.5508634956515566
0.5304611439607582
0.5100587922699598
0.48965644057916136
0.46925408888836295
0.44885173719756455
0.42844938550676626
0.40804703381596785
0.38764468212516945
0.36724233043437104
0.34683997874357264
0.32643762705277424
0.30603527536197583
0.28563292367117743
0.265230571980379
0.24482822028958073
0.22442586859878233
0.20402351690798393
0.18362116521718552
0.16321881352638712
0.14281646183558871
0.12241411014479031
0.10201175845399202
0.08160940676319361
0.06120705507239521
0.04080470338159681
0.020402351690798404
0.0
```

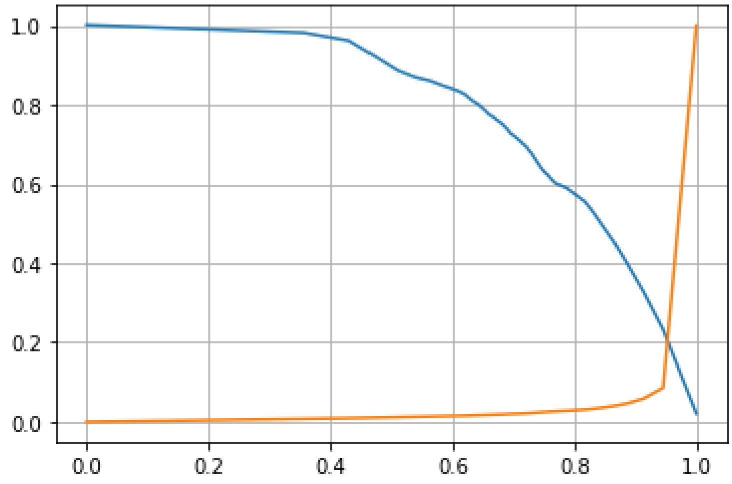
In [39]:

```
# Complete NaN values
metrics_copy = metrics_.copy()
metrics_copy = complete_nan_values(metrics_copy)
```

In [40]:

```
# Comput Mean Average Precision (mAP) score
Recall = metrics_copy[:,0]
Precision = metrics_copy[:,1]
AA = metrics_copy[:,2]

DeltaR = Recall[1:]-Recall[:-1]
AP = np.sum(Precision[:-1]*DeltaR)
print('mAP', AP)

# Plot Recall vs. Precision curve
plt.close('all')
plt.plot(metrics_copy[:,0],metrics_copy[:,1])
plt.plot(metrics_copy[:,0],metrics_copy[:,2])
plt.grid()
```

```
mAP 0.8055881170596597
```



In [ ]: