In [20]:

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

```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```

In [21]:

```
#%autoreload # When utils.py is updated
from utils_unet_resunet import *
from tensorflow.keras.preprocessing.image import ImageDataGenerator
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]:

```
# 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)
```

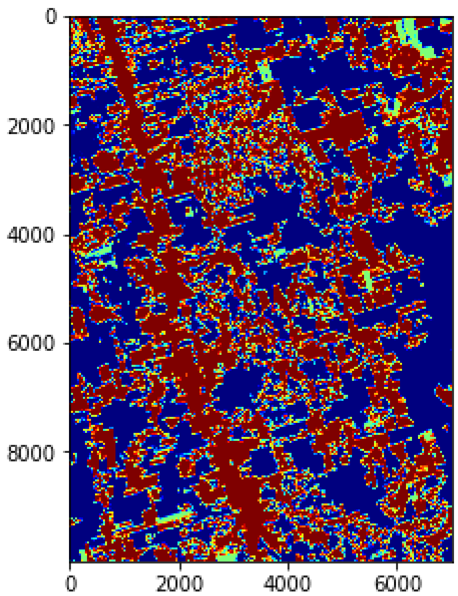
```
Tiles size:  2000 1750
Mask size:  (10000, 7000)
mask:  (10000, 7000)
image stack:  (10000, 7000, 8)
ref : (10000, 7000)
```

In [6]:

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

Out[6]:

```
<matplotlib.image.AxesImage at 0x1ec52b747f0>
```



In [7]:

```
# 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 [12]:

```
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 [22]:

```
# 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)
```

In [23]:

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

In [24]:

```
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', 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"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_3 (InputLayer)            [(None, 128, 128, 8) 0
__________________________________________________________________________________________________
conv1 (Conv2D)                  (None, 128, 128, 32) 2336        input_3[0][0]
__________________________________________________________________________________________________
max_pooling2d (MaxPooling2D)    (None, 64, 64, 32)   0           conv1[0][0]
__________________________________________________________________________________________________
conv2 (Conv2D)                  (None, 64, 64, 64)   18496       max_pooling2d[0][0]
__________________________________________________________________________________________________
max_pooling2d_1 (MaxPooling2D)  (None, 32, 32, 64)   0           conv2[0][0]
__________________________________________________________________________________________________
conv3 (Conv2D)                  (None, 32, 32, 128)  73856       max_pooling2d_1[0]
[0]
__________________________________________________________________________________________________
max_pooling2d_2 (MaxPooling2D)  (None, 16, 16, 128)  0           conv3[0][0]
__________________________________________________________________________________________________
conv4 (Conv2D)                  (None, 16, 16, 128)  147584      max_pooling2d_2[0]
[0]
__________________________________________________________________________________________________
conv5 (Conv2D)                  (None, 16, 16, 128)  147584      conv4[0][0]
__________________________________________________________________________________________________
conv6 (Conv2D)                  (None, 16, 16, 128)  147584      conv5[0][0]
__________________________________________________________________________________________________
up_sampling2d (UpSampling2D)    (None, 32, 32, 128)  0           conv6[0][0]
__________________________________________________________________________________________________
upsampling3 (Conv2D)            (None, 32, 32, 128)  147584      up_sampling2d[0][0]
__________________________________________________________________________________________________
concatenate3 (Concatenate)      (None, 32, 32, 256)  0           conv3[0][0]
                                                                 upsampling3[0][0]
__________________________________________________________________________________________________
up_sampling2d_1 (UpSampling2D)  (None, 64, 64, 256)  0           concatenate3[0][0]
__________________________________________________________________________________________________
upsampling2 (Conv2D)            (None, 64, 64, 64)   147520      up_sampling2d_1[0]
[0]
__________________________________________________________________________________________________
concatenate2 (Concatenate)      (None, 64, 64, 128)  0           conv2[0][0]
```

```
                                                                upsampling2[0][0]
__________________________________________________________________________________
__________
up_sampling2d_2 (UpSampling2D)  (None, 128, 128, 128 0          concatenate2[0][0]
__________________________________________________________________________________
__________
upsampling1 (Conv2D)            (None, 128, 128, 32) 36896      up_sampling2d_2[0]
[0]
__________________________________________________________________________________
__________
concatenate1 (Concatenate)      (None, 128, 128, 64) 0          conv1[0][0]
                                                                upsampling1[0][0]
__________________________________________________________________________________
__________
conv2d (Conv2D)                 (None, 128, 128, 3)  195        concatenate1[0][0]
==================================================================================
==========
Total params: 869,635
Trainable params: 869,635
Non-trainable params: 0
__________________________________________________________________________________
__________
Epoch 1/100
```

```
C:\Users\felferrari\AppData\Roaming\Python\Python38\site-packages\tensorflow\python
\keras\engine\training.py:1844: UserWarning: `Model.fit_generator` is deprecated and
will be removed in a future version. Please use `Model.fit`, which supports generato
rs.
  warnings.warn('`Model.fit_generator` is deprecated and '
108/108 [==============================] - 12s 56ms/step - loss: 0.0908 - accuracy:
0.7706 - val_loss: 0.1033 - val_accuracy: 0.7722

Epoch 00001: val_loss improved from inf to 0.10332, saving model to imgs/experiment
s/exp1/models\unet_0.h5
Epoch 2/100
108/108 [==============================] - 5s 46ms/step - loss: 0.0656 - accuracy:
0.8252 - val_loss: 0.1161 - val_accuracy: 0.7735

Epoch 00002: val_loss did not improve from 0.10332
Epoch 3/100
108/108 [==============================] - 5s 46ms/step - loss: 0.0599 - accuracy:
0.8358 - val_loss: 0.1093 - val_accuracy: 0.7830

Epoch 00003: val_loss did not improve from 0.10332
Epoch 4/100
108/108 [==============================] - 5s 46ms/step - loss: 0.0580 - accuracy:
0.8411 - val_loss: 0.1128 - val_accuracy: 0.7837

Epoch 00004: val_loss did not improve from 0.10332
Epoch 5/100
108/108 [==============================] - 5s 45ms/step - loss: 0.0524 - accuracy:
0.8521 - val_loss: 0.1238 - val_accuracy: 0.7878

Epoch 00005: val_loss did not improve from 0.10332
Epoch 6/100
108/108 [==============================] - 5s 46ms/step - loss: 0.0504 - accuracy:
0.8549 - val_loss: 0.1584 - val_accuracy: 0.7884

Epoch 00006: val_loss did not improve from 0.10332
Epoch 7/100
108/108 [==============================] - 5s 47ms/step - loss: 0.0450 - accuracy:
0.8620 - val_loss: 0.1584 - val_accuracy: 0.7828

Epoch 00007: val_loss did not improve from 0.10332
```

```
Epoch 8/100
108/108 [==============================] - 5s 47ms/step - loss: 0.0409 - accuracy:
0.8728 - val_loss: 0.1516 - val_accuracy: 0.7759

Epoch 00008: val_loss did not improve from 0.10332
Epoch 9/100
108/108 [==============================] - 5s 46ms/step - loss: 0.0392 - accuracy:
0.8751 - val_loss: 0.1537 - val_accuracy: 0.7970

Epoch 00009: val_loss did not improve from 0.10332
Epoch 10/100
108/108 [==============================] - 5s 47ms/step - loss: 0.0408 - accuracy:
0.8712 - val_loss: 0.1750 - val_accuracy: 0.7867

Epoch 00010: val_loss did not improve from 0.10332
Epoch 11/100
108/108 [==============================] - 5s 47ms/step - loss: 0.0334 - accuracy:
0.8866 - val_loss: 0.1722 - val_accuracy: 0.7887

Epoch 00011: val_loss did not improve from 0.10332
Epoch 00011: early stopping
time:  1
Model: "model"
____________________________________________________________________________________
______________
Layer (type)                    Output Shape         Param #     Connected to
====================================================================================
==============
input_3 (InputLayer)            [(None, 128, 128, 8) 0
____________________________________________________________________________________
______________
conv1 (Conv2D)                  (None, 128, 128, 32) 2336        input_3[0][0]
____________________________________________________________________________________
______________
max_pooling2d (MaxPooling2D)    (None, 64, 64, 32)   0           conv1[0][0]
____________________________________________________________________________________
______________
conv2 (Conv2D)                  (None, 64, 64, 64)   18496       max_pooling2d[0][0]
____________________________________________________________________________________
______________
max_pooling2d_1 (MaxPooling2D)  (None, 32, 32, 64)   0           conv2[0][0]
____________________________________________________________________________________
______________
conv3 (Conv2D)                  (None, 32, 32, 128)  73856       max_pooling2d_1[0]
[0]
____________________________________________________________________________________
______________
max_pooling2d_2 (MaxPooling2D)  (None, 16, 16, 128)  0           conv3[0][0]
____________________________________________________________________________________
______________
conv4 (Conv2D)                  (None, 16, 16, 128)  147584      max_pooling2d_2[0]
[0]
____________________________________________________________________________________
______________
conv5 (Conv2D)                  (None, 16, 16, 128)  147584      conv4[0][0]
____________________________________________________________________________________
______________
conv6 (Conv2D)                  (None, 16, 16, 128)  147584      conv5[0][0]
____________________________________________________________________________________
______________
up_sampling2d (UpSampling2D)    (None, 32, 32, 128)  0           conv6[0][0]
____________________________________________________________________________________
______________
upsampling3 (Conv2D)            (None, 32, 32, 128)  147584      up_sampling2d[0][0]
____________________________________________________________________________________
```

```
______________
concatenate3 (Concatenate)      (None, 32, 32, 256)  0           conv3[0][0]
                                                                 upsampling3[0][0]
__________________________________________________________________________________
______________
up_sampling2d_1 (UpSampling2D)  (None, 64, 64, 256)  0           concatenate3[0][0]
__________________________________________________________________________________
______________
upsampling2 (Conv2D)            (None, 64, 64, 64)   147520      up_sampling2d_1[0]
[0]
__________________________________________________________________________________
______________
concatenate2 (Concatenate)      (None, 64, 64, 128)  0           conv2[0][0]
                                                                 upsampling2[0][0]
__________________________________________________________________________________
______________
up_sampling2d_2 (UpSampling2D)  (None, 128, 128, 128 0           concatenate2[0][0]
__________________________________________________________________________________
______________
upsampling1 (Conv2D)            (None, 128, 128, 32) 36896       up_sampling2d_2[0]
[0]
__________________________________________________________________________________
______________
concatenate1 (Concatenate)      (None, 128, 128, 64) 0           conv1[0][0]
                                                                 upsampling1[0][0]
__________________________________________________________________________________
______________
conv2d (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 [==============================] - 6s 50ms/step - loss: 0.0318 - accuracy:
0.8912 - val_loss: 0.1972 - val_accuracy: 0.7895

Epoch 00001: val_loss improved from inf to 0.19716, saving model to imgs/experiment
s/exp1/models\unet_1.h5
Epoch 2/100
108/108 [==============================] - 5s 49ms/step - loss: 0.0291 - accuracy:
0.8966 - val_loss: 0.1919 - val_accuracy: 0.7952

Epoch 00002: val_loss improved from 0.19716 to 0.19192, saving model to imgs/experim
ents/exp1/models\unet_1.h5
Epoch 3/100
108/108 [==============================] - 5s 50ms/step - loss: 0.0277 - accuracy:
0.9009 - val_loss: 0.1663 - val_accuracy: 0.7914

Epoch 00003: val_loss improved from 0.19192 to 0.16634, saving model to imgs/experim
ents/exp1/models\unet_1.h5
Epoch 4/100
108/108 [==============================] - 5s 49ms/step - loss: 0.0290 - accuracy:
0.8971 - val_loss: 0.1731 - val_accuracy: 0.7854

Epoch 00004: val_loss did not improve from 0.16634
Epoch 5/100
108/108 [==============================] - 5s 50ms/step - loss: 0.0252 - accuracy:
0.9074 - val_loss: 0.1915 - val_accuracy: 0.7872

Epoch 00005: val_loss did not improve from 0.16634
Epoch 6/100
108/108 [==============================] - 6s 52ms/step - loss: 0.0238 - accuracy:
```

```
0.9121 - val_loss: 0.2250 - val_accuracy: 0.7724

Epoch 00006: val_loss did not improve from 0.16634
Epoch 7/100
108/108 [==============================] - 6s 57ms/step - loss: 0.0234 - accuracy:
0.9133 - val_loss: 0.2329 - val_accuracy: 0.7839

Epoch 00007: val_loss did not improve from 0.16634
Epoch 8/100
108/108 [==============================] - 6s 56ms/step - loss: 0.0223 - accuracy:
0.9168 - val_loss: 0.2319 - val_accuracy: 0.7839

Epoch 00008: val_loss did not improve from 0.16634
Epoch 9/100
108/108 [==============================] - 6s 58ms/step - loss: 0.0214 - accuracy:
0.9181 - val_loss: 0.2385 - val_accuracy: 0.7823

Epoch 00009: val_loss did not improve from 0.16634
Epoch 10/100
108/108 [==============================] - 6s 58ms/step - loss: 0.0207 - accuracy:
0.9217 - val_loss: 0.2355 - val_accuracy: 0.7873

Epoch 00010: val_loss did not improve from 0.16634
Epoch 11/100
108/108 [==============================] - 6s 57ms/step - loss: 0.0199 - accuracy:
0.9244 - val_loss: 0.2587 - val_accuracy: 0.7810ss:

Epoch 00011: val_loss did not improve from 0.16634
Epoch 12/100
108/108 [==============================] - 6s 57ms/step - loss: 0.0182 - accuracy:
0.9300 - val_loss: 0.2534 - val_accuracy: 0.7848

Epoch 00012: val_loss did not improve from 0.16634
Epoch 13/100
108/108 [==============================] - 6s 54ms/step - loss: 0.0186 - accuracy:
0.9291 - val_loss: 0.2600 - val_accuracy: 0.7844

Epoch 00013: val_loss did not improve from 0.16634
Epoch 00013: early stopping
time:  2
Model: "model"
____________________________________________________________________________________
______________
Layer (type)                    Output Shape         Param #     Connected to
====================================================================================
==============
input_3 (InputLayer)            [(None, 128, 128, 8) 0
____________________________________________________________________________________
______________
conv1 (Conv2D)                  (None, 128, 128, 32) 2336        input_3[0][0]
____________________________________________________________________________________
______________
max_pooling2d (MaxPooling2D)    (None, 64, 64, 32)   0           conv1[0][0]
____________________________________________________________________________________
______________
conv2 (Conv2D)                  (None, 64, 64, 64)   18496       max_pooling2d[0][0]
____________________________________________________________________________________
______________
max_pooling2d_1 (MaxPooling2D)  (None, 32, 32, 64)   0           conv2[0][0]
____________________________________________________________________________________
______________
conv3 (Conv2D)                  (None, 32, 32, 128)  73856       max_pooling2d_1[0]
[0]
____________________________________________________________________________________
______________
```

```
max_pooling2d_2 (MaxPooling2D)  (None, 16, 16, 128)  0           conv3[0][0]
__________________________________________________________________________________________________
conv4 (Conv2D)                  (None, 16, 16, 128)  147584      max_pooling2d_2[0]
[0]
__________________________________________________________________________________________________
conv5 (Conv2D)                  (None, 16, 16, 128)  147584      conv4[0][0]
__________________________________________________________________________________________________
conv6 (Conv2D)                  (None, 16, 16, 128)  147584      conv5[0][0]
__________________________________________________________________________________________________
up_sampling2d (UpSampling2D)    (None, 32, 32, 128)  0           conv6[0][0]
__________________________________________________________________________________________________
upsampling3 (Conv2D)            (None, 32, 32, 128)  147584      up_sampling2d[0][0]
__________________________________________________________________________________________________
concatenate3 (Concatenate)      (None, 32, 32, 256)  0           conv3[0][0]
                                                                 upsampling3[0][0]
__________________________________________________________________________________________________
up_sampling2d_1 (UpSampling2D)  (None, 64, 64, 256)  0           concatenate3[0][0]
__________________________________________________________________________________________________
upsampling2 (Conv2D)            (None, 64, 64, 64)   147520      up_sampling2d_1[0]
[0]
__________________________________________________________________________________________________
concatenate2 (Concatenate)      (None, 64, 64, 128)  0           conv2[0][0]
                                                                 upsampling2[0][0]
__________________________________________________________________________________________________
up_sampling2d_2 (UpSampling2D)  (None, 128, 128, 128 0           concatenate2[0][0]
__________________________________________________________________________________________________
upsampling1 (Conv2D)            (None, 128, 128, 32) 36896       up_sampling2d_2[0]
[0]
__________________________________________________________________________________________________
concatenate1 (Concatenate)      (None, 128, 128, 64) 0           conv1[0][0]
                                                                 upsampling1[0][0]
__________________________________________________________________________________________________
conv2d (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 [==============================] - 7s 55ms/step - loss: 0.0183 - accuracy:
0.9295 - val_loss: 0.2424 - val_accuracy: 0.7773

Epoch 00001: val_loss improved from inf to 0.24236, saving model to imgs/experiment
s/exp1/models\unet_2.h5
Epoch 2/100
108/108 [==============================] - 6s 55ms/step - loss: 0.0184 - accuracy:
0.9305 - val_loss: 0.2229 - val_accuracy: 0.7835

Epoch 00002: val_loss improved from 0.24236 to 0.22294, saving model to imgs/experim
ents/exp1/models\unet_2.h5
```

```
Epoch 3/100
108/108 [==============================] - 6s 56ms/step - loss: 0.0163 - accuracy:
0.9370 - val_loss: 0.2340 - val_accuracy: 0.7834

Epoch 00003: val_loss did not improve from 0.22294
Epoch 4/100
108/108 [==============================] - 6s 56ms/step - loss: 0.0165 - accuracy:
0.9362 - val_loss: 0.2691 - val_accuracy: 0.7766

Epoch 00004: val_loss did not improve from 0.22294
Epoch 5/100
108/108 [==============================] - 6s 57ms/step - loss: 0.0167 - accuracy:
0.9360 - val_loss: 0.2799 - val_accuracy: 0.7700

Epoch 00005: val_loss did not improve from 0.22294
Epoch 6/100
108/108 [==============================] - 6s 59ms/step - loss: 0.0156 - accuracy:
0.9390 - val_loss: 0.2741 - val_accuracy: 0.7736

Epoch 00006: val_loss did not improve from 0.22294
Epoch 7/100
108/108 [==============================] - 6s 59ms/step - loss: 0.0146 - accuracy:
0.9433 - val_loss: 0.2866 - val_accuracy: 0.7732

Epoch 00007: val_loss did not improve from 0.22294
Epoch 8/100
108/108 [==============================] - 6s 60ms/step - loss: 0.0146 - accuracy:
0.9430 - val_loss: 0.2920 - val_accuracy: 0.7737

Epoch 00008: val_loss did not improve from 0.22294
Epoch 9/100
108/108 [==============================] - 6s 58ms/step - loss: 0.0143 - accuracy:
0.9443 - val_loss: 0.2609 - val_accuracy: 0.7719

Epoch 00009: val_loss did not improve from 0.22294
Epoch 10/100
108/108 [==============================] - 6s 60ms/step - loss: 0.0151 - accuracy:
0.9416 - val_loss: 0.2816 - val_accuracy: 0.7757

Epoch 00010: val_loss did not improve from 0.22294
Epoch 11/100
108/108 [==============================] - 7s 61ms/step - loss: 0.0145 - accuracy:
0.9436 - val_loss: 0.2611 - val_accuracy: 0.7786

Epoch 00011: val_loss did not improve from 0.22294
Epoch 12/100
108/108 [==============================] - 6s 60ms/step - loss: 0.0183 - accuracy:
0.9342 - val_loss: 0.2428 - val_accuracy: 0.7857

Epoch 00012: val_loss did not improve from 0.22294
Epoch 00012: early stopping
time:  3
Model: "model"
____________________________________________________________________________________
______________
Layer (type)                    Output Shape         Param #     Connected to
====================================================================================
==============
input_3 (InputLayer)            [(None, 128, 128, 8) 0
____________________________________________________________________________________
______________
conv1 (Conv2D)                  (None, 128, 128, 32) 2336        input_3[0][0]
____________________________________________________________________________________
______________
max_pooling2d (MaxPooling2D)    (None, 64, 64, 32)   0           conv1[0][0]
```

```
__________________________________________________________________________________________________
conv2 (Conv2D)                  (None, 64, 64, 64)   18496       max_pooling2d[0][0]
__________________________________________________________________________________________________
max_pooling2d_1 (MaxPooling2D)  (None, 32, 32, 64)   0           conv2[0][0]
__________________________________________________________________________________________________
conv3 (Conv2D)                  (None, 32, 32, 128)  73856       max_pooling2d_1[0]
[0]
__________________________________________________________________________________________________
max_pooling2d_2 (MaxPooling2D)  (None, 16, 16, 128)  0           conv3[0][0]
__________________________________________________________________________________________________
conv4 (Conv2D)                  (None, 16, 16, 128)  147584      max_pooling2d_2[0]
[0]
__________________________________________________________________________________________________
conv5 (Conv2D)                  (None, 16, 16, 128)  147584      conv4[0][0]
__________________________________________________________________________________________________
conv6 (Conv2D)                  (None, 16, 16, 128)  147584      conv5[0][0]
__________________________________________________________________________________________________
up_sampling2d (UpSampling2D)    (None, 32, 32, 128)  0           conv6[0][0]
__________________________________________________________________________________________________
upsampling3 (Conv2D)            (None, 32, 32, 128)  147584      up_sampling2d[0][0]
__________________________________________________________________________________________________
concatenate3 (Concatenate)      (None, 32, 32, 256)  0           conv3[0][0]
                                                                 upsampling3[0][0]
__________________________________________________________________________________________________
up_sampling2d_1 (UpSampling2D)  (None, 64, 64, 256)  0           concatenate3[0][0]
__________________________________________________________________________________________________
upsampling2 (Conv2D)            (None, 64, 64, 64)   147520      up_sampling2d_1[0]
[0]
__________________________________________________________________________________________________
concatenate2 (Concatenate)      (None, 64, 64, 128)  0           conv2[0][0]
                                                                 upsampling2[0][0]
__________________________________________________________________________________________________
up_sampling2d_2 (UpSampling2D)  (None, 128, 128, 128 0           concatenate2[0][0]
__________________________________________________________________________________________________
upsampling1 (Conv2D)            (None, 128, 128, 32) 36896       up_sampling2d_2[0]
[0]
__________________________________________________________________________________________________
concatenate1 (Concatenate)      (None, 128, 128, 64) 0           conv1[0][0]
                                                                 upsampling1[0][0]
__________________________________________________________________________________________________
conv2d (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 [==============================] - 8s 66ms/step - loss: 0.0156 - accuracy:
0.9408 - val_loss: 0.2439 - val_accuracy: 0.7756

Epoch 00001: val_loss improved from inf to 0.24389, saving model to imgs/experiment
s/exp1/models\unet_3.h5
Epoch 2/100
108/108 [==============================] - 7s 64ms/step - loss: 0.0134 - accuracy:
0.9479 - val_loss: 0.2624 - val_accuracy: 0.7769

Epoch 00002: val_loss did not improve from 0.24389
Epoch 3/100
108/108 [==============================] - 7s 62ms/step - loss: 0.0130 - accuracy:
0.9493 - val_loss: 0.2969 - val_accuracy: 0.7700

Epoch 00003: val_loss did not improve from 0.24389
Epoch 4/100
108/108 [==============================] - 7s 62ms/step - loss: 0.0124 - accuracy:
0.9515 - val_loss: 0.2965 - val_accuracy: 0.7747

Epoch 00004: val_loss did not improve from 0.24389
Epoch 5/100
108/108 [==============================] - 7s 64ms/step - loss: 0.0124 - accuracy:
0.9515 - val_loss: 0.2937 - val_accuracy: 0.7750

Epoch 00005: val_loss did not improve from 0.24389
Epoch 6/100
108/108 [==============================] - 7s 66ms/step - loss: 0.0123 - accuracy:
0.9513 - val_loss: 0.2949 - val_accuracy: 0.7742

Epoch 00006: val_loss did not improve from 0.24389
Epoch 7/100
108/108 [==============================] - 7s 68ms/step - loss: 0.0122 - accuracy:
0.9530 - val_loss: 0.2989 - val_accuracy: 0.7684

Epoch 00007: val_loss did not improve from 0.24389
Epoch 8/100
108/108 [==============================] - 7s 68ms/step - loss: 0.0119 - accuracy:
0.9532 - val_loss: 0.2945 - val_accuracy: 0.7750

Epoch 00008: val_loss did not improve from 0.24389
Epoch 9/100
108/108 [==============================] - 7s 68ms/step - loss: 0.0117 - accuracy:
0.9542 - val_loss: 0.2989 - val_accuracy: 0.7741

Epoch 00009: val_loss did not improve from 0.24389
Epoch 10/100
108/108 [==============================] - 7s 68ms/step - loss: 0.0118 - accuracy:
0.9536 - val_loss: 0.2806 - val_accuracy: 0.7789

Epoch 00010: val_loss did not improve from 0.24389
Epoch 11/100
108/108 [==============================] - 7s 69ms/step - loss: 0.0127 - accuracy:
0.9502 - val_loss: 0.2836 - val_accuracy: 0.7800

Epoch 00011: val_loss did not improve from 0.24389
Epoch 00011: early stopping
time:  4
Model: "model"
____________________________________________________________________________________
______________
Layer (type)                    Output Shape         Param #     Connected to
====================================================================================
==============
input_3 (InputLayer)            [(None, 128, 128, 8) 0
```

```
__________________________________________________________________________________________________
conv1 (Conv2D)                  (None, 128, 128, 32) 2336        input_3[0][0]
__________________________________________________________________________________________________
max_pooling2d (MaxPooling2D)    (None, 64, 64, 32)   0           conv1[0][0]
__________________________________________________________________________________________________
conv2 (Conv2D)                  (None, 64, 64, 64)   18496       max_pooling2d[0][0]
__________________________________________________________________________________________________
max_pooling2d_1 (MaxPooling2D)  (None, 32, 32, 64)   0           conv2[0][0]
__________________________________________________________________________________________________
conv3 (Conv2D)                  (None, 32, 32, 128)  73856       max_pooling2d_1[0]
[0]
__________________________________________________________________________________________________
max_pooling2d_2 (MaxPooling2D)  (None, 16, 16, 128)  0           conv3[0][0]
__________________________________________________________________________________________________
conv4 (Conv2D)                  (None, 16, 16, 128)  147584      max_pooling2d_2[0]
[0]
__________________________________________________________________________________________________
conv5 (Conv2D)                  (None, 16, 16, 128)  147584      conv4[0][0]
__________________________________________________________________________________________________
conv6 (Conv2D)                  (None, 16, 16, 128)  147584      conv5[0][0]
__________________________________________________________________________________________________
up_sampling2d (UpSampling2D)    (None, 32, 32, 128)  0           conv6[0][0]
__________________________________________________________________________________________________
upsampling3 (Conv2D)            (None, 32, 32, 128)  147584      up_sampling2d[0][0]
__________________________________________________________________________________________________
concatenate3 (Concatenate)      (None, 32, 32, 256)  0           conv3[0][0]
                                                                 upsampling3[0][0]
__________________________________________________________________________________________________
up_sampling2d_1 (UpSampling2D)  (None, 64, 64, 256)  0           concatenate3[0][0]
__________________________________________________________________________________________________
upsampling2 (Conv2D)            (None, 64, 64, 64)   147520      up_sampling2d_1[0]
[0]
__________________________________________________________________________________________________
concatenate2 (Concatenate)      (None, 64, 64, 128)  0           conv2[0][0]
                                                                 upsampling2[0][0]
__________________________________________________________________________________________________
up_sampling2d_2 (UpSampling2D)  (None, 128, 128, 128 0           concatenate2[0][0]
__________________________________________________________________________________________________
upsampling1 (Conv2D)            (None, 128, 128, 32) 36896       up_sampling2d_2[0]
[0]
__________________________________________________________________________________________________
concatenate1 (Concatenate)      (None, 128, 128, 64) 0           conv1[0][0]
                                                                 upsampling1[0][0]
__________________________________________________________________________________________________
conv2d (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 [==============================] - 8s 69ms/step - loss: 0.0121 - accuracy:
0.9534 - val_loss: 0.2605 - val_accuracy: 0.7742

Epoch 00001: val_loss improved from inf to 0.26045, saving model to imgs/experiment
s/exp1/models\unet_4.h5
Epoch 2/100
108/108 [==============================] - 7s 68ms/step - loss: 0.0114 - accuracy:
0.9552 - val_loss: 0.2789 - val_accuracy: 0.77120.0114  - ETA: 1s - loss: 0.0114 - a
c

Epoch 00002: val_loss did not improve from 0.26045
Epoch 3/100
108/108 [==============================] - 7s 69ms/step - loss: 0.0114 - accuracy:
0.9556 - val_loss: 0.3120 - val_accuracy: 0.7703

Epoch 00003: val_loss did not improve from 0.26045
Epoch 4/100
108/108 [==============================] - 8s 71ms/step - loss: 0.0112 - accuracy:
0.9565 - val_loss: 0.3032 - val_accuracy: 0.7698 3s - loss: 0.0112 - accuracy - ETA:
3s - loss: 0.0112 - accuracy:  - ETA: 3s - loss: - ETA: 2s - los

Epoch 00004: val_loss did not improve from 0.26045
Epoch 5/100
108/108 [==============================] - 8s 73ms/step - loss: 0.0109 - accuracy:
0.9569 - val_loss: 0.3099 - val_accuracy: 0.7727

Epoch 00005: val_loss did not improve from 0.26045
Epoch 6/100
108/108 [==============================] - 8s 71ms/step - loss: 0.0107 - accuracy:
0.9585 - val_loss: 0.3182 - val_accuracy: 0.7738

Epoch 00006: val_loss did not improve from 0.26045
Epoch 7/100
108/108 [==============================] - 8s 74ms/step - loss: 0.0107 - accuracy:
0.9581 - val_loss: 0.3247 - val_accuracy: 0.7658

Epoch 00007: val_loss did not improve from 0.26045
Epoch 8/100
108/108 [==============================] - 8s 74ms/step - loss: 0.0105 - accuracy:
0.9586 - val_loss: 0.2955 - val_accuracy: 0.7727

Epoch 00008: val_loss did not improve from 0.26045
Epoch 9/100
108/108 [==============================] - 8s 74ms/step - loss: 0.0111 - accuracy:
0.9566 - val_loss: 0.3106 - val_accuracy: 0.7724

Epoch 00009: val_loss did not improve from 0.26045
Epoch 10/100
108/108 [==============================] - 8s 75ms/step - loss: 0.0105 - accuracy:
0.9591 - val_loss: 0.3016 - val_accuracy: 0.7765

Epoch 00010: val_loss did not improve from 0.26045
Epoch 11/100
108/108 [==============================] - 8s 78ms/step - loss: 0.0107 - accuracy:
0.9587 - val_loss: 0.2961 - val_accuracy: 0.7772

Epoch 00011: val_loss did not improve from 0.26045
Epoch 00011: early stopping
```

In [26]:

```
# 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)

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())

    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, model, patches_pred
time_ts_array = np.asarray(time_ts)
# 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 [27]:

```
# 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
```

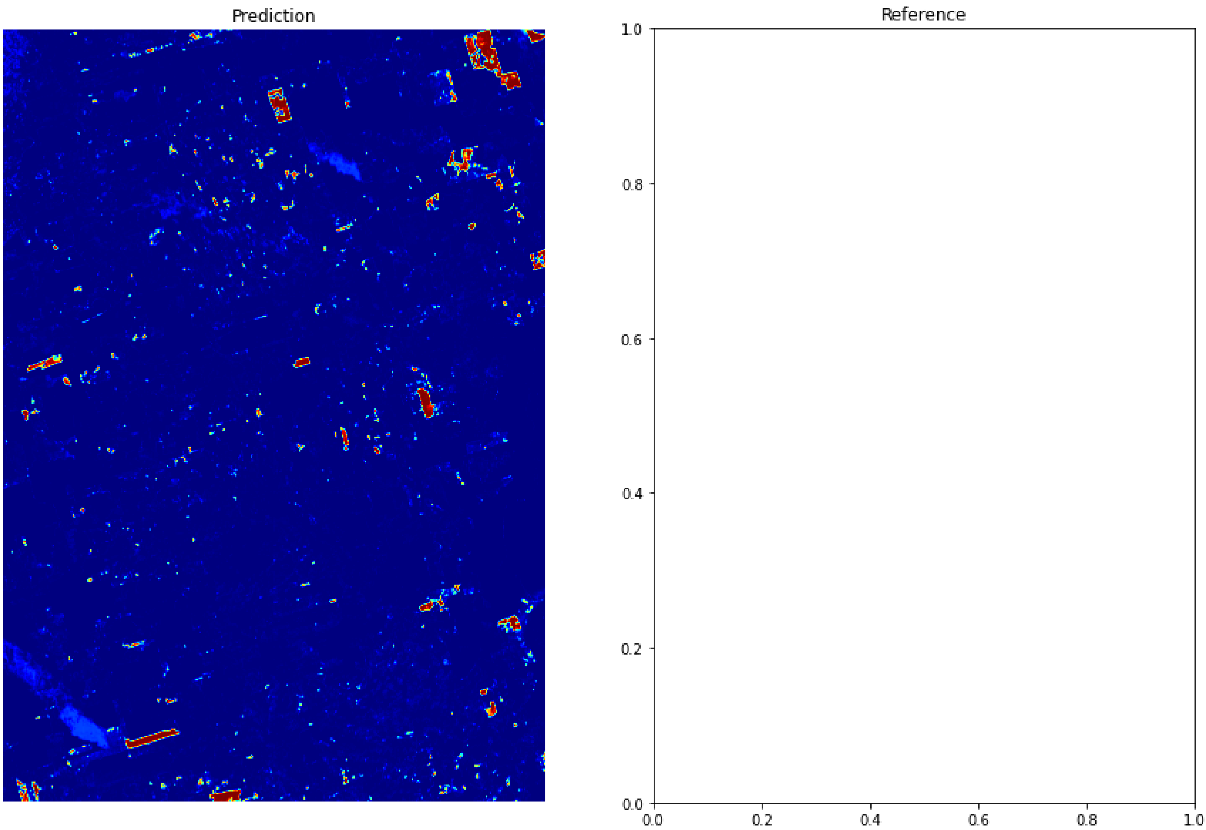
In [28]:

```
# 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_2019, cmap ='jet')
ax2.axis('off')
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
C:\Users\FELFER~1\AppData\Local\Temp/ipykernel_12872/3622215040.py in <module>
      8 ax2 = fig.add_subplot(122)
      9 plt.title('Reference')
---> 10 ax2.imshow(ref_2019, cmap ='jet')
     11 ax2.axis('off')

NameError: name 'ref_2019' is not defined
```



In [29]:
```
# 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_)
```

```
0.9987759828567505
D:\Ferrari\proj_1\Mabel_original\utils_unet_resunet.py:200: RuntimeWarning: invalid
value encountered in longlong_scalars
  precision_ = TP/(TP+FP)
0.9783927995331434
0.9580096162095362
0.9376264328859291
0.917243249562322
0.8968600662387147
0.8764768829151076
0.8560936995915005
0.835710516267893
0.815327332944286
0.7949441496206789
0.7745609662970718
0.7541777829734646
0.7337945996498575
0.7134114163262504
0.6930282330026432
0.672645049679036
0.6522618663554289
0.6318786830318217
0.6114954997082146
0.5911123163846075
0.5707291330610003
0.5503459497373931
0.529962766413786
0.5095795830901788
0.4891963997665717
0.4688132164429645
0.44843003311935736
0.42804684979575025
0.407663666472143
0.3872804831485359
0.3668972998249287
0.3465141165013216
0.32613093317771447
0.30574774985410724
0.28536456653050013
0.2649813832068929
0.2445981998832858
0.2242150165596786
0.2038318332360714
0.18344864991246435
0.16306546658885723
0.1426822832652
0.1222990999416429
0.10191591661803578
0.08153273329442856
```

```
0.06114954997082145
0.040766366647214225
0.020383183323607112
0.0
```

In [30]:

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

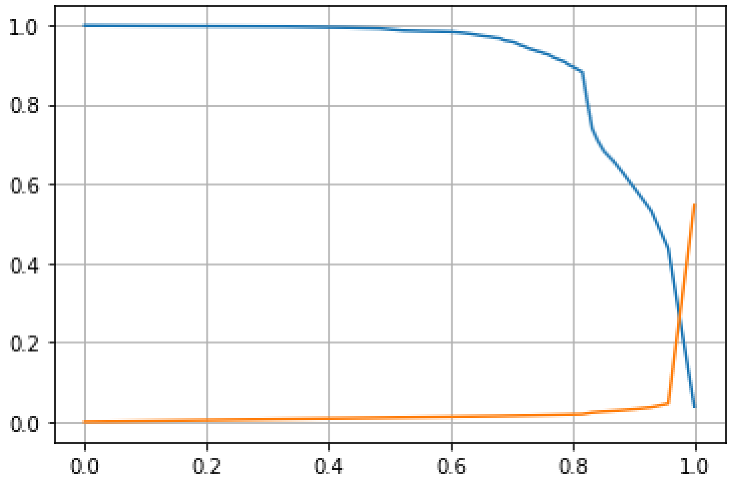
In [31]:

```
# 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 0.9124369937467522
```



In [ ]: