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
img_type = 'FUSION'

if img_type == 'FUSION':
    image_array = np.load(root_path+'New_Images/fus_stack.npy')


if img_type == 'OPT':
    image_array = np.load(root_path+'New_Images/opt_stack.npy')


if img_type == 'SAR':
    image_array = np.load(root_path+'New_Images/sar_stack.npy')
print('Image stack:', image_array.shape)

final_mask1 = np.load(root_path+'New_Images/'+'final_mask1.npy')
print('Labels stack:', final_mask1.shape)

h_, w_, channels = image_array.shape
n_opt_layers = 20
```

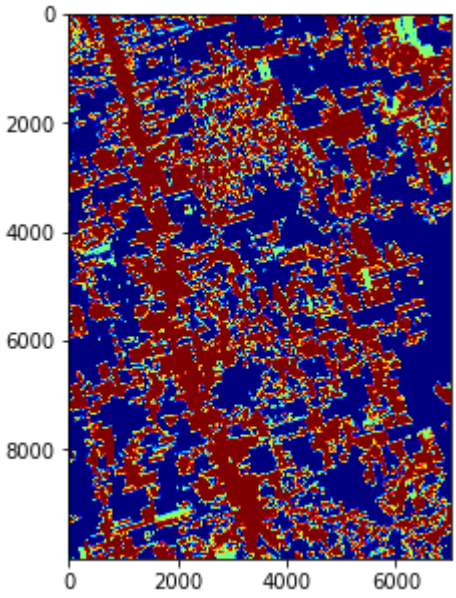
```
Image stack: (10000, 7000, 24)
Labels stack: (10000, 7000)
```

In [4]:
```
# 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, 24)
ref : (10000, 7000)
```

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

Out[5]:
```
<matplotlib.image.AxesImage at 0x1595b947160>
```



In [6]:
```
# 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 [7]:
```
# 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 [8]:
```
# 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 [9]:
```
# 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 [10]:

```
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 [11]:

```
exp = 3
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 [12]:

```
# 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, n_opt_layers)
```

In [13]:

```
# 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 = weights)
#loss = WBCE(weights = weights, class_indexes = [0, 1])
```

In [14]:

```
metrics_all = []
times=5
for tm in range(0,times):
    print('time: ', tm)

    rows = patch_size
    cols = patch_size
    adam = Adam(lr = 1e-4 , beta_1=0.9)

    loss = weighted_categorical_crossentropy(weights)
    #loss = WBCE(weights = weights)
    #loss = WBCE(weights = weights, class_indexes = [0, 1])

    #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, n_opt_layers)
    model.build((None,)+input_shape)

    model.compile(optimizer=adam, loss=loss, metrics=['accuracy'])
    model.summary()

    earlystop = EarlyStopping(monitor='val_loss', min_delta=0.0001, patience=10, ver
    #earlystop = EarlyStopping(monitor='val_loss', min_delta=0.0001, patience=10, ve
    #checkpoint = ModelCheckpoint(path_models+ '/' + method +'_'+str(tm)+'.h5', moni
    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(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
    metrics_all.append(end_training)
    del model, history
```

```
time:  0
```

```
Model: "model_3_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
opt_encoder (UNET_Encoder)   multiple                  540896
_________________________________________________________________
sar_encoder (UNET_Encoder)   multiple                  536288
_________________________________________________________________
decoder (UNET_Decoder)       multiple                  332000
_________________________________________________________________
opt_classifier (Classifier)  multiple                  195
_________________________________________________________________
sar_classifier (Classifier)  multiple                  195
_________________________________________________________________
fus_classifier (Classifier)  multiple                  195
_________________________________________________________________
combination (CombinationLaye multiple                  3
=================================================================
Total params: 1,409,786
Trainable params: 1,409,769
Non-trainable params: 17
_________________________________________________________________
Epoch 1/100
108/108 [==============================] - 24s 163ms/step - opt_accuracy: 0.8050 - s
ar_accuracy: 0.7008 - fus_accuracy: 0.7571 - opt_loss: 0.1094 - sar_loss: 0.1505 - f
usion_loss: 0.1412 - loss: 0.4011 - val_opt_accuracy: 0.8468 - val_sar_accuracy: 0.7
649 - val_fus_accuracy: 0.8525 - val_opt_loss: 0.1036 - val_sar_loss: 0.1185 - val_f
usion_loss: 0.1033 - val_loss: 0.3253

Epoch 00001: val_loss improved from inf to 0.32530, saving model to imgs/experiment
s/exp3/models\unet_0.h5
Epoch 2/100
108/108 [==============================] - 15s 141ms/step - opt_accuracy: 0.8462 - s
ar_accuracy: 0.7313 - fus_accuracy: 0.8147 - opt_loss: 0.0883 - sar_loss: 0.1365 - f
usion_loss: 0.1028 - loss: 0.3276 - val_opt_accuracy: 0.8531 - val_sar_accuracy: 0.7
839 - val_fus_accuracy: 0.8534 - val_opt_loss: 0.1258 - val_sar_loss: 0.1194 - val_f
usion_loss: 0.1329 - val_loss: 0.3781

Epoch 00002: val_loss did not improve from 0.32530
Epoch 3/100
108/108 [==============================] - 15s 141ms/step - opt_accuracy: 0.8570 - s
ar_accuracy: 0.7482 - fus_accuracy: 0.8281 - opt_loss: 0.0830 - sar_loss: 0.1312 - f
usion_loss: 0.0967 - loss: 0.3109 - val_opt_accuracy: 0.8518 - val_sar_accuracy: 0.7
898 - val_fus_accuracy: 0.8500 - val_opt_loss: 0.1049 - val_sar_loss: 0.1159 - val_f
usion_loss: 0.1039 - val_loss: 0.3247

Epoch 00003: val_loss improved from 0.32530 to 0.32467, saving model to imgs/experim
ents/exp3/models\unet_0.h5
Epoch 4/100
108/108 [==============================] - 15s 142ms/step - opt_accuracy: 0.8667 - s
ar_accuracy: 0.7553 - fus_accuracy: 0.8393 - opt_loss: 0.0769 - sar_loss: 0.1276 - f
usion_loss: 0.0904 - loss: 0.2948 - val_opt_accuracy: 0.8650 - val_sar_accuracy: 0.7
569 - val_fus_accuracy: 0.8647 - val_opt_loss: 0.0921 - val_sar_loss: 0.1276 - val_f
usion_loss: 0.0878 - val_loss: 0.3075

Epoch 00004: val_loss improved from 0.32467 to 0.30752, saving model to imgs/experim
ents/exp3/models\unet_0.h5
Epoch 5/100
108/108 [==============================] - 15s 142ms/step - opt_accuracy: 0.8727 - s
ar_accuracy: 0.7578 - fus_accuracy: 0.8467 - opt_loss: 0.0734 - sar_loss: 0.1265 - f
usion_loss: 0.0856 - loss: 0.2856 - val_opt_accuracy: 0.8668 - val_sar_accuracy: 0.7
951 - val_fus_accuracy: 0.8662 - val_opt_loss: 0.0913 - val_sar_loss: 0.1106 - val_f
usion_loss: 0.0911 - val_loss: 0.2930

Epoch 00005: val_loss improved from 0.30752 to 0.29297, saving model to imgs/experim
```

```
ents/exp3/models\unet_0.h5
Epoch 6/100
108/108 [==============================] - 15s 142ms/step - opt_accuracy: 0.8785 - s
ar_accuracy: 0.7624 - fus_accuracy: 0.8571 - opt_loss: 0.0684 - sar_loss: 0.1237 - f
usion_loss: 0.0794 - loss: 0.2715 - val_opt_accuracy: 0.8632 - val_sar_accuracy: 0.8
026 - val_fus_accuracy: 0.8613 - val_opt_loss: 0.1060 - val_sar_loss: 0.1142 - val_f
usion_loss: 0.1094 - val_loss: 0.3295

Epoch 00006: val_loss did not improve from 0.29297
Epoch 7/100
108/108 [==============================] - 15s 143ms/step - opt_accuracy: 0.8822 - s
ar_accuracy: 0.7663 - fus_accuracy: 0.8640 - opt_loss: 0.0659 - sar_loss: 0.1232 - f
usion_loss: 0.0757 - loss: 0.2648 - val_opt_accuracy: 0.8577 - val_sar_accuracy: 0.7
992 - val_fus_accuracy: 0.8545 - val_opt_loss: 0.1212 - val_sar_loss: 0.1137 - val_f
usion_loss: 0.1204 - val_loss: 0.3552

Epoch 00007: val_loss did not improve from 0.29297
Epoch 8/100
108/108 [==============================] - 15s 142ms/step - opt_accuracy: 0.8855 - s
ar_accuracy: 0.7702 - fus_accuracy: 0.8694 - opt_loss: 0.0626 - sar_loss: 0.1216 - f
usion_loss: 0.0715 - loss: 0.2558 - val_opt_accuracy: 0.8655 - val_sar_accuracy: 0.7
920 - val_fus_accuracy: 0.8629 - val_opt_loss: 0.1206 - val_sar_loss: 0.1147 - val_f
usion_loss: 0.1220 - val_loss: 0.3573

Epoch 00008: val_loss did not improve from 0.29297
Epoch 9/100
108/108 [==============================] - 15s 142ms/step - opt_accuracy: 0.8910 - s
ar_accuracy: 0.7749 - fus_accuracy: 0.8762 - opt_loss: 0.0586 - sar_loss: 0.1213 - f
usion_loss: 0.0665 - loss: 0.2463 - val_opt_accuracy: 0.8634 - val_sar_accuracy: 0.8
045 - val_fus_accuracy: 0.8614 - val_opt_loss: 0.1177 - val_sar_loss: 0.1106 - val_f
usion_loss: 0.1176 - val_loss: 0.3459

Epoch 00009: val_loss did not improve from 0.29297
Epoch 10/100
108/108 [==============================] - 15s 143ms/step - opt_accuracy: 0.8942 - s
ar_accuracy: 0.7760 - fus_accuracy: 0.8795 - opt_loss: 0.0561 - sar_loss: 0.1193 - f
usion_loss: 0.0633 - loss: 0.2386 - val_opt_accuracy: 0.8642 - val_sar_accuracy: 0.8
111 - val_fus_accuracy: 0.8642 - val_opt_loss: 0.1196 - val_sar_loss: 0.1136 - val_f
usion_loss: 0.1196 - val_loss: 0.3527

Epoch 00010: val_loss did not improve from 0.29297
Epoch 11/100
108/108 [==============================] - 15s 143ms/step - opt_accuracy: 0.8986 - s
ar_accuracy: 0.7706 - fus_accuracy: 0.8844 - opt_loss: 0.0529 - sar_loss: 0.1228 - f
usion_loss: 0.0595 - loss: 0.2353 - val_opt_accuracy: 0.8622 - val_sar_accuracy: 0.8
023 - val_fus_accuracy: 0.8600 - val_opt_loss: 0.1202 - val_sar_loss: 0.1154 - val_f
usion_loss: 0.1209 - val_loss: 0.3565

Epoch 00011: val_loss did not improve from 0.29297
Epoch 12/100
108/108 [==============================] - 16s 144ms/step - opt_accuracy: 0.9011 - s
ar_accuracy: 0.7401 - fus_accuracy: 0.8882 - opt_loss: 0.0508 - sar_loss: 0.1315 - f
usion_loss: 0.0569 - loss: 0.2391 - val_opt_accuracy: 0.8626 - val_sar_accuracy: 0.8
006 - val_fus_accuracy: 0.8627 - val_opt_loss: 0.1372 - val_sar_loss: 0.1142 - val_f
usion_loss: 0.1407 - val_loss: 0.3921

Epoch 00012: val_loss did not improve from 0.29297
Epoch 13/100
108/108 [==============================] - 16s 145ms/step - opt_accuracy: 0.9046 - s
ar_accuracy: 0.7387 - fus_accuracy: 0.8926 - opt_loss: 0.0483 - sar_loss: 0.1358 - f
usion_loss: 0.0538 - loss: 0.2379 - val_opt_accuracy: 0.8646 - val_sar_accuracy: 0.7
815 - val_fus_accuracy: 0.8627 - val_opt_loss: 0.1510 - val_sar_loss: 0.1262 - val_f
usion_loss: 0.1514 - val_loss: 0.4286

Epoch 00013: val_loss did not improve from 0.29297
```

```
Epoch 14/100
108/108 [==============================] - 16s 145ms/step - opt_accuracy: 0.9070 - s
ar_accuracy: 0.7096 - fus_accuracy: 0.8982 - opt_loss: 0.0462 - sar_loss: 0.1479 - f
usion_loss: 0.0502 - loss: 0.2443 - val_opt_accuracy: 0.8696 - val_sar_accuracy: 0.7
338 - val_fus_accuracy: 0.8685 - val_opt_loss: 0.1446 - val_sar_loss: 0.1298 - val_f
usion_loss: 0.1385 - val_loss: 0.4129

Epoch 00014: val_loss did not improve from 0.29297
Epoch 15/100
108/108 [==============================] - 16s 144ms/step - opt_accuracy: 0.9110 - s
ar_accuracy: 0.7121 - fus_accuracy: 0.9028 - opt_loss: 0.0436 - sar_loss: 0.1460 - f
usion_loss: 0.0471 - loss: 0.2366 - val_opt_accuracy: 0.8665 - val_sar_accuracy: 0.7
446 - val_fus_accuracy: 0.8654 - val_opt_loss: 0.1416 - val_sar_loss: 0.1202 - val_f
usion_loss: 0.1365 - val_loss: 0.3983

Epoch 00015: val_loss did not improve from 0.29297
Epoch 00015: early stopping
time:  1
Model: "model_3_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
opt_encoder (UNET_Encoder)   multiple                  540896
_________________________________________________________________
sar_encoder (UNET_Encoder)   multiple                  536288
_________________________________________________________________
decoder (UNET_Decoder)       multiple                  332000
_________________________________________________________________
opt_classifier (Classifier)  multiple                  195
_________________________________________________________________
sar_classifier (Classifier)  multiple                  195
_________________________________________________________________
fus_classifier (Classifier)  multiple                  195
_________________________________________________________________
combination (CombinationLaye multiple                  3
=================================================================
Total params: 1,409,786
Trainable params: 1,409,769
Non-trainable params: 17
_________________________________________________________________
Epoch 1/100
108/108 [==============================] - 19s 156ms/step - opt_accuracy: 0.7924 - s
ar_accuracy: 0.7168 - fus_accuracy: 0.7039 - opt_loss: 0.1164 - sar_loss: 0.1450 - f
usion_loss: 0.1638 - loss: 0.4253 - val_opt_accuracy: 0.8596 - val_sar_accuracy: 0.7
758 - val_fus_accuracy: 0.8536 - val_opt_loss: 0.0869 - val_sar_loss: 0.1211 - val_f
usion_loss: 0.0936 - val_loss: 0.3016

Epoch 00001: val_loss improved from inf to 0.30161, saving model to imgs/experiment
s/exp3/models\unet_1.h5
Epoch 2/100
108/108 [==============================] - 16s 149ms/step - opt_accuracy: 0.8543 - s
ar_accuracy: 0.7491 - fus_accuracy: 0.8044 - opt_loss: 0.0867 - sar_loss: 0.1346 - f
usion_loss: 0.1103 - loss: 0.3316 - val_opt_accuracy: 0.8592 - val_sar_accuracy: 0.7
831 - val_fus_accuracy: 0.8569 - val_opt_loss: 0.0941 - val_sar_loss: 0.1208 - val_f
usion_loss: 0.0992 - val_loss: 0.3141

Epoch 00002: val_loss did not improve from 0.30161
Epoch 3/100
108/108 [==============================] - 16s 146ms/step - opt_accuracy: 0.8639 - s
ar_accuracy: 0.7587 - fus_accuracy: 0.8262 - opt_loss: 0.0788 - sar_loss: 0.1290 - f
usion_loss: 0.0978 - loss: 0.3056 - val_opt_accuracy: 0.8635 - val_sar_accuracy: 0.7
919 - val_fus_accuracy: 0.8634 - val_opt_loss: 0.1095 - val_sar_loss: 0.1190 - val_f
usion_loss: 0.1103 - val_loss: 0.3388

Epoch 00003: val_loss did not improve from 0.30161
```

```
Epoch 4/100
108/108 [==============================] - 16s 147ms/step - opt_accuracy: 0.8694 - s
ar_accuracy: 0.7595 - fus_accuracy: 0.8419 - opt_loss: 0.0747 - sar_loss: 0.1285 - f
usion_loss: 0.0898 - loss: 0.2930 - val_opt_accuracy: 0.8554 - val_sar_accuracy: 0.7
814 - val_fus_accuracy: 0.8565 - val_opt_loss: 0.1253 - val_sar_loss: 0.1275 - val_f
usion_loss: 0.1256 - val_loss: 0.3784

Epoch 00004: val_loss did not improve from 0.30161
Epoch 5/100
108/108 [==============================] - 16s 147ms/step - opt_accuracy: 0.8752 - s
ar_accuracy: 0.7598 - fus_accuracy: 0.8511 - opt_loss: 0.0707 - sar_loss: 0.1281 - f
usion_loss: 0.0845 - loss: 0.2833 - val_opt_accuracy: 0.8675 - val_sar_accuracy: 0.7
929 - val_fus_accuracy: 0.8659 - val_opt_loss: 0.1062 - val_sar_loss: 0.1196 - val_f
usion_loss: 0.1153 - val_loss: 0.3411

Epoch 00005: val_loss did not improve from 0.30161
Epoch 6/100
108/108 [==============================] - 16s 147ms/step - opt_accuracy: 0.8801 - s
ar_accuracy: 0.7661 - fus_accuracy: 0.8585 - opt_loss: 0.0677 - sar_loss: 0.1253 - f
usion_loss: 0.0797 - loss: 0.2727 - val_opt_accuracy: 0.8538 - val_sar_accuracy: 0.7
759 - val_fus_accuracy: 0.8519 - val_opt_loss: 0.1364 - val_sar_loss: 0.1272 - val_f
usion_loss: 0.1278 - val_loss: 0.3914

Epoch 00006: val_loss did not improve from 0.30161
Epoch 7/100
108/108 [==============================] - 16s 148ms/step - opt_accuracy: 0.8836 - s
ar_accuracy: 0.7685 - fus_accuracy: 0.8637 - opt_loss: 0.0646 - sar_loss: 0.1245 - f
usion_loss: 0.0761 - loss: 0.2652 - val_opt_accuracy: 0.8648 - val_sar_accuracy: 0.7
949 - val_fus_accuracy: 0.8649 - val_opt_loss: 0.1313 - val_sar_loss: 0.1163 - val_f
usion_loss: 0.1342 - val_loss: 0.3818

Epoch 00007: val_loss did not improve from 0.30161
Epoch 8/100
108/108 [==============================] - 16s 148ms/step - opt_accuracy: 0.8874 - s
ar_accuracy: 0.7692 - fus_accuracy: 0.8684 - opt_loss: 0.0620 - sar_loss: 0.1234 - f
usion_loss: 0.0726 - loss: 0.2580 - val_opt_accuracy: 0.8639 - val_sar_accuracy: 0.7
911 - val_fus_accuracy: 0.8651 - val_opt_loss: 0.1309 - val_sar_loss: 0.1188 - val_f
usion_loss: 0.1295 - val_loss: 0.3792

Epoch 00008: val_loss did not improve from 0.30161
Epoch 9/100
108/108 [==============================] - 16s 149ms/step - opt_accuracy: 0.8920 - s
ar_accuracy: 0.7683 - fus_accuracy: 0.8749 - opt_loss: 0.0584 - sar_loss: 0.1231 - f
usion_loss: 0.0678 - loss: 0.2492 - val_opt_accuracy: 0.8666 - val_sar_accuracy: 0.8
021 - val_fus_accuracy: 0.8675 - val_opt_loss: 0.1115 - val_sar_loss: 0.1099 - val_f
usion_loss: 0.1172 - val_loss: 0.3386

Epoch 00009: val_loss did not improve from 0.30161
Epoch 10/100
108/108 [==============================] - 16s 149ms/step - opt_accuracy: 0.8952 - s
ar_accuracy: 0.7712 - fus_accuracy: 0.8798 - opt_loss: 0.0561 - sar_loss: 0.1214 - f
usion_loss: 0.0645 - loss: 0.2420 - val_opt_accuracy: 0.8698 - val_sar_accuracy: 0.7
876 - val_fus_accuracy: 0.8678 - val_opt_loss: 0.1157 - val_sar_loss: 0.1198 - val_f
usion_loss: 0.1258 - val_loss: 0.3613

Epoch 00010: val_loss did not improve from 0.30161
Epoch 11/100
108/108 [==============================] - 16s 150ms/step - opt_accuracy: 0.8994 - s
ar_accuracy: 0.7743 - fus_accuracy: 0.8848 - opt_loss: 0.0529 - sar_loss: 0.1201 - f
usion_loss: 0.0604 - loss: 0.2333 - val_opt_accuracy: 0.8686 - val_sar_accuracy: 0.7
809 - val_fus_accuracy: 0.8675 - val_opt_loss: 0.1193 - val_sar_loss: 0.1157 - val_f
usion_loss: 0.1224 - val_loss: 0.3573

Epoch 00011: val_loss did not improve from 0.30161
Epoch 00011: early stopping
```

```
time:  2
Model: "model_3_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
opt_encoder (UNET_Encoder)   multiple                  540896
_________________________________________________________________
sar_encoder (UNET_Encoder)   multiple                  536288
_________________________________________________________________
decoder (UNET_Decoder)       multiple                  332000
_________________________________________________________________
opt_classifier (Classifier)  multiple                  195
_________________________________________________________________
sar_classifier (Classifier)  multiple                  195
_________________________________________________________________
fus_classifier (Classifier)  multiple                  195
_________________________________________________________________
combination (CombinationLaye multiple                  3
=================================================================
Total params: 1,409,786
Trainable params: 1,409,769
Non-trainable params: 17
_________________________________________________________________
Epoch 1/100
108/108 [==============================] - 18s 152ms/step - opt_accuracy: 0.7504 - s
ar_accuracy: 0.7017 - fus_accuracy: 0.7360 - opt_loss: 0.1283 - sar_loss: 0.1467 - f
usion_loss: 0.1529 - loss: 0.4279 - val_opt_accuracy: 0.8617 - val_sar_accuracy: 0.7
790 - val_fus_accuracy: 0.8603 - val_opt_loss: 0.0918 - val_sar_loss: 0.1178 - val_f
usion_loss: 0.0913 - val_loss: 0.3009

Epoch 00001: val_loss improved from inf to 0.30092, saving model to imgs/experiment
s/exp3/models\unet_2.h5
Epoch 2/100
108/108 [==============================] - 16s 150ms/step - opt_accuracy: 0.8307 - s
ar_accuracy: 0.7445 - fus_accuracy: 0.8145 - opt_loss: 0.0942 - sar_loss: 0.1351 - f
usion_loss: 0.1083 - loss: 0.3376 - val_opt_accuracy: 0.8573 - val_sar_accuracy: 0.7
837 - val_fus_accuracy: 0.8604 - val_opt_loss: 0.1022 - val_sar_loss: 0.1200 - val_f
usion_loss: 0.0982 - val_loss: 0.3204

Epoch 00002: val_loss did not improve from 0.30092
Epoch 3/100
108/108 [==============================] - 16s 151ms/step - opt_accuracy: 0.8508 - s
ar_accuracy: 0.7598 - fus_accuracy: 0.8393 - opt_loss: 0.0863 - sar_loss: 0.1270 - f
usion_loss: 0.0956 - loss: 0.3090 - val_opt_accuracy: 0.8544 - val_sar_accuracy: 0.7
937 - val_fus_accuracy: 0.8610 - val_opt_loss: 0.1111 - val_sar_loss: 0.1158 - val_f
usion_loss: 0.1067 - val_loss: 0.3336

Epoch 00003: val_loss did not improve from 0.30092
Epoch 4/100
108/108 [==============================] - 16s 151ms/step - opt_accuracy: 0.8618 - s
ar_accuracy: 0.7693 - fus_accuracy: 0.8517 - opt_loss: 0.0796 - sar_loss: 0.1225 - f
usion_loss: 0.0877 - loss: 0.2898 - val_opt_accuracy: 0.8535 - val_sar_accuracy: 0.8
027 - val_fus_accuracy: 0.8610 - val_opt_loss: 0.1144 - val_sar_loss: 0.1156 - val_f
usion_loss: 0.1081 - val_loss: 0.3382

Epoch 00004: val_loss did not improve from 0.30092
Epoch 5/100
108/108 [==============================] - 16s 153ms/step - opt_accuracy: 0.8683 - s
ar_accuracy: 0.7617 - fus_accuracy: 0.8583 - opt_loss: 0.0765 - sar_loss: 0.1233 - f
usion_loss: 0.0831 - loss: 0.2829 - val_opt_accuracy: 0.8543 - val_sar_accuracy: 0.7
874 - val_fus_accuracy: 0.8610 - val_opt_loss: 0.1080 - val_sar_loss: 0.1147 - val_f
usion_loss: 0.1016 - val_loss: 0.3244

Epoch 00005: val_loss did not improve from 0.30092
Epoch 6/100
```

```
108/108 [==============================] - 17s 153ms/step - opt_accuracy: 0.8719 - s
ar_accuracy: 0.7766 - fus_accuracy: 0.8629 - opt_loss: 0.0737 - sar_loss: 0.1172 - f
usion_loss: 0.0789 - loss: 0.2698 - val_opt_accuracy: 0.8542 - val_sar_accuracy: 0.8
023 - val_fus_accuracy: 0.8654 - val_opt_loss: 0.1056 - val_sar_loss: 0.1122 - val_f
usion_loss: 0.1017 - val_loss: 0.3195

Epoch 00006: val_loss did not improve from 0.30092
Epoch 7/100
108/108 [==============================] - 17s 154ms/step - opt_accuracy: 0.8778 - s
ar_accuracy: 0.7759 - fus_accuracy: 0.8689 - opt_loss: 0.0699 - sar_loss: 0.1161 - f
usion_loss: 0.0746 - loss: 0.2606 - val_opt_accuracy: 0.8639 - val_sar_accuracy: 0.8
075 - val_fus_accuracy: 0.8664 - val_opt_loss: 0.1055 - val_sar_loss: 0.1116 - val_f
usion_loss: 0.0993 - val_loss: 0.3164

Epoch 00007: val_loss did not improve from 0.30092
Epoch 8/100
108/108 [==============================] - 17s 155ms/step - opt_accuracy: 0.8798 - s
ar_accuracy: 0.7782 - fus_accuracy: 0.8713 - opt_loss: 0.0682 - sar_loss: 0.1153 - f
usion_loss: 0.0722 - loss: 0.2557 - val_opt_accuracy: 0.8551 - val_sar_accuracy: 0.8
103 - val_fus_accuracy: 0.8620 - val_opt_loss: 0.1279 - val_sar_loss: 0.1130 - val_f
usion_loss: 0.1201 - val_loss: 0.3610

Epoch 00008: val_loss did not improve from 0.30092
Epoch 9/100
108/108 [==============================] - 17s 155ms/step - opt_accuracy: 0.8852 - s
ar_accuracy: 0.7879 - fus_accuracy: 0.8759 - opt_loss: 0.0643 - sar_loss: 0.1123 - f
usion_loss: 0.0688 - loss: 0.2454 - val_opt_accuracy: 0.8672 - val_sar_accuracy: 0.8
115 - val_fus_accuracy: 0.8654 - val_opt_loss: 0.1075 - val_sar_loss: 0.1123 - val_f
usion_loss: 0.1042 - val_loss: 0.3240

Epoch 00009: val_loss did not improve from 0.30092
Epoch 10/100
108/108 [==============================] - 17s 155ms/step - opt_accuracy: 0.8875 - s
ar_accuracy: 0.7900 - fus_accuracy: 0.8785 - opt_loss: 0.0620 - sar_loss: 0.1108 - f
usion_loss: 0.0665 - loss: 0.2393 - val_opt_accuracy: 0.8673 - val_sar_accuracy: 0.7
991 - val_fus_accuracy: 0.8655 - val_opt_loss: 0.1200 - val_sar_loss: 0.1107 - val_f
usion_loss: 0.1097 - val_loss: 0.3404

Epoch 00010: val_loss did not improve from 0.30092
Epoch 11/100
108/108 [==============================] - 17s 155ms/step - opt_accuracy: 0.8897 - s
ar_accuracy: 0.7917 - fus_accuracy: 0.8816 - opt_loss: 0.0603 - sar_loss: 0.1120 - f
usion_loss: 0.0648 - loss: 0.2371 - val_opt_accuracy: 0.8691 - val_sar_accuracy: 0.8
088 - val_fus_accuracy: 0.8680 - val_opt_loss: 0.1095 - val_sar_loss: 0.1217 - val_f
usion_loss: 0.1063 - val_loss: 0.3375

Epoch 00011: val_loss did not improve from 0.30092
Epoch 00011: early stopping
time:  3
Model: "model_3_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
opt_encoder (UNET_Encoder)   multiple                  540896
_________________________________________________________________
sar_encoder (UNET_Encoder)   multiple                  536288
_________________________________________________________________
decoder (UNET_Decoder)       multiple                  332000
_________________________________________________________________
opt_classifier (Classifier)  multiple                  195
_________________________________________________________________
sar_classifier (Classifier)  multiple                  195
_________________________________________________________________
fus_classifier (Classifier)  multiple                  195
_________________________________________________________________
```

```
combination (CombinationLaye multiple                  3
=================================================================
Total params: 1,409,786
Trainable params: 1,409,769
Non-trainable params: 17

_________________________________________________________________
Epoch 1/100
108/108 [==============================] - 20s 162ms/step - opt_accuracy: 0.8165 - s
ar_accuracy: 0.6907 - fus_accuracy: 0.7310 - opt_loss: 0.1094 - sar_loss: 0.1591 - f
usion_loss: 0.1474 - loss: 0.4160 - val_opt_accuracy: 0.8640 - val_sar_accuracy: 0.7
733 - val_fus_accuracy: 0.8569 - val_opt_loss: 0.0828 - val_sar_loss: 0.1210 - val_f
usion_loss: 0.0844 - val_loss: 0.2882

Epoch 00001: val_loss improved from inf to 0.28816, saving model to imgs/experiment
s/exp3/models\unet_3.h5
Epoch 2/100
108/108 [==============================] - 17s 158ms/step - opt_accuracy: 0.8588 - s
ar_accuracy: 0.7315 - fus_accuracy: 0.8251 - opt_loss: 0.0849 - sar_loss: 0.1427 - f
usion_loss: 0.0993 - loss: 0.3270 - val_opt_accuracy: 0.8614 - val_sar_accuracy: 0.7
778 - val_fus_accuracy: 0.8548 - val_opt_loss: 0.0976 - val_sar_loss: 0.1216 - val_f
usion_loss: 0.0900 - val_loss: 0.3093

Epoch 00002: val_loss did not improve from 0.28816
Epoch 3/100
108/108 [==============================] - 17s 160ms/step - opt_accuracy: 0.8640 - s
ar_accuracy: 0.7367 - fus_accuracy: 0.8394 - opt_loss: 0.0802 - sar_loss: 0.1397 - f
usion_loss: 0.0916 - loss: 0.3115 - val_opt_accuracy: 0.8616 - val_sar_accuracy: 0.7
794 - val_fus_accuracy: 0.8581 - val_opt_loss: 0.1137 - val_sar_loss: 0.1206 - val_f
usion_loss: 0.0990 - val_loss: 0.3333

Epoch 00003: val_loss did not improve from 0.28816
Epoch 4/100
108/108 [==============================] - 17s 160ms/step - opt_accuracy: 0.8701 - s
ar_accuracy: 0.7416 - fus_accuracy: 0.8495 - opt_loss: 0.0761 - sar_loss: 0.1368 - f
usion_loss: 0.0857 - loss: 0.2985 - val_opt_accuracy: 0.8615 - val_sar_accuracy: 0.7
865 - val_fus_accuracy: 0.8588 - val_opt_loss: 0.1052 - val_sar_loss: 0.1178 - val_f
usion_loss: 0.0948 - val_loss: 0.3178

Epoch 00004: val_loss did not improve from 0.28816
Epoch 5/100
108/108 [==============================] - 17s 161ms/step - opt_accuracy: 0.8749 - s
ar_accuracy: 0.7509 - fus_accuracy: 0.8571 - opt_loss: 0.0722 - sar_loss: 0.1321 - f
usion_loss: 0.0808 - loss: 0.2851 - val_opt_accuracy: 0.8581 - val_sar_accuracy: 0.7
954 - val_fus_accuracy: 0.8564 - val_opt_loss: 0.1116 - val_sar_loss: 0.1146 - val_f
usion_loss: 0.1007 - val_loss: 0.3269

Epoch 00005: val_loss did not improve from 0.28816
Epoch 6/100
108/108 [==============================] - 17s 161ms/step - opt_accuracy: 0.8790 - s
ar_accuracy: 0.7547 - fus_accuracy: 0.8637 - opt_loss: 0.0687 - sar_loss: 0.1282 - f
usion_loss: 0.0766 - loss: 0.2735 - val_opt_accuracy: 0.8571 - val_sar_accuracy: 0.7
958 - val_fus_accuracy: 0.8534 - val_opt_loss: 0.1058 - val_sar_loss: 0.1156 - val_f
usion_loss: 0.0973 - val_loss: 0.3186

Epoch 00006: val_loss did not improve from 0.28816
Epoch 7/100
108/108 [==============================] - 17s 162ms/step - opt_accuracy: 0.8834 - s
ar_accuracy: 0.7543 - fus_accuracy: 0.8698 - opt_loss: 0.0657 - sar_loss: 0.1282 - f
usion_loss: 0.0727 - loss: 0.2666 - val_opt_accuracy: 0.8648 - val_sar_accuracy: 0.7
968 - val_fus_accuracy: 0.8625 - val_opt_loss: 0.1053 - val_sar_loss: 0.1149 - val_f
usion_loss: 0.0998 - val_loss: 0.3200

Epoch 00007: val_loss did not improve from 0.28816
Epoch 8/100
108/108 [==============================] - 18s 164ms/step - opt_accuracy: 0.8860 - s
```

```
ar_accuracy: 0.7563 - fus_accuracy: 0.8732 - opt_loss: 0.0627 - sar_loss: 0.1261 - f
usion_loss: 0.0690 - loss: 0.2578 - val_opt_accuracy: 0.8638 - val_sar_accuracy: 0.7
880 - val_fus_accuracy: 0.8628 - val_opt_loss: 0.1083 - val_sar_loss: 0.1156 - val_f
usion_loss: 0.0997 - val_loss: 0.3236

Epoch 00008: val_loss did not improve from 0.28816
Epoch 9/100
108/108 [==============================] - 18s 163ms/step - opt_accuracy: 0.8897 - s
ar_accuracy: 0.7436 - fus_accuracy: 0.8786 - opt_loss: 0.0602 - sar_loss: 0.1350 - f
usion_loss: 0.0662 - loss: 0.2613 - val_opt_accuracy: 0.8660 - val_sar_accuracy: 0.7
944 - val_fus_accuracy: 0.8632 - val_opt_loss: 0.1120 - val_sar_loss: 0.1155 - val_f
usion_loss: 0.1054 - val_loss: 0.3329

Epoch 00009: val_loss did not improve from 0.28816
Epoch 10/100
108/108 [==============================] - 18s 164ms/step - opt_accuracy: 0.8908 - s
ar_accuracy: 0.7509 - fus_accuracy: 0.8797 - opt_loss: 0.0592 - sar_loss: 0.1300 - f
usion_loss: 0.0649 - loss: 0.2540 - val_opt_accuracy: 0.8676 - val_sar_accuracy: 0.7
951 - val_fus_accuracy: 0.8666 - val_opt_loss: 0.1069 - val_sar_loss: 0.1145 - val_f
usion_loss: 0.0996 - val_loss: 0.3210

Epoch 00010: val_loss did not improve from 0.28816
Epoch 11/100
108/108 [==============================] - 18s 166ms/step - opt_accuracy: 0.8962 - s
ar_accuracy: 0.7515 - fus_accuracy: 0.8859 - opt_loss: 0.0553 - sar_loss: 0.1312 - f
usion_loss: 0.0605 - loss: 0.2469 - val_opt_accuracy: 0.8644 - val_sar_accuracy: 0.7
830 - val_fus_accuracy: 0.8622 - val_opt_loss: 0.0957 - val_sar_loss: 0.1129 - val_f
usion_loss: 0.0933 - val_loss: 0.3019

Epoch 00011: val_loss did not improve from 0.28816
Epoch 00011: early stopping
time:  4
Model: "model_3_5"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
opt_encoder (UNET_Encoder)   multiple                  540896
_________________________________________________________________
sar_encoder (UNET_Encoder)   multiple                  536288
_________________________________________________________________
decoder (UNET_Decoder)       multiple                  332000
_________________________________________________________________
opt_classifier (Classifier)  multiple                  195
_________________________________________________________________
sar_classifier (Classifier)  multiple                  195
_________________________________________________________________
fus_classifier (Classifier)  multiple                  195
_________________________________________________________________
combination (CombinationLaye multiple                  3
=================================================================
Total params: 1,409,786
Trainable params: 1,409,769
Non-trainable params: 17
_________________________________________________________________
Epoch 1/100
108/108 [==============================] - 20s 167ms/step - opt_accuracy: 0.7478 - s
ar_accuracy: 0.7002 - fus_accuracy: 0.7693 - opt_loss: 0.1240 - sar_loss: 0.1486 - f
usion_loss: 0.1294 - loss: 0.4020 - val_opt_accuracy: 0.8533 - val_sar_accuracy: 0.7
777 - val_fus_accuracy: 0.8618 - val_opt_loss: 0.0876 - val_sar_loss: 0.1192 - val_f
usion_loss: 0.0886 - val_loss: 0.2955

Epoch 00001: val_loss improved from inf to 0.29546, saving model to imgs/experiment
s/exp3/models\unet_4.h5
Epoch 2/100
108/108 [==============================] - 18s 166ms/step - opt_accuracy: 0.8371 - s
```

```
ar_accuracy: 0.7489 - fus_accuracy: 0.8304 - opt_loss: 0.0871 - sar_loss: 0.1345 - f
usion_loss: 0.0964 - loss: 0.3179 - val_opt_accuracy: 0.8444 - val_sar_accuracy: 0.7
872 - val_fus_accuracy: 0.8559 - val_opt_loss: 0.1083 - val_sar_loss: 0.1177 - val_f
usion_loss: 0.1063 - val_loss: 0.3322

Epoch 00002: val_loss did not improve from 0.29546
Epoch 3/100
108/108 [==============================] - 18s 167ms/step - opt_accuracy: 0.8558 - s
ar_accuracy: 0.7588 - fus_accuracy: 0.8429 - opt_loss: 0.0794 - sar_loss: 0.1292 - f
usion_loss: 0.0883 - loss: 0.2968 - val_opt_accuracy: 0.8553 - val_sar_accuracy: 0.7
947 - val_fus_accuracy: 0.8624 - val_opt_loss: 0.1165 - val_sar_loss: 0.1176 - val_f
usion_loss: 0.1166 - val_loss: 0.3507

Epoch 00003: val_loss did not improve from 0.29546
Epoch 4/100
108/108 [==============================] - 18s 168ms/step - opt_accuracy: 0.8622 - s
ar_accuracy: 0.7672 - fus_accuracy: 0.8518 - opt_loss: 0.0750 - sar_loss: 0.1257 - f
usion_loss: 0.0828 - loss: 0.2836 - val_opt_accuracy: 0.8615 - val_sar_accuracy: 0.7
977 - val_fus_accuracy: 0.8652 - val_opt_loss: 0.1047 - val_sar_loss: 0.1182 - val_f
usion_loss: 0.1096 - val_loss: 0.3325

Epoch 00004: val_loss did not improve from 0.29546
Epoch 5/100
108/108 [==============================] - 18s 168ms/step - opt_accuracy: 0.8700 - s
ar_accuracy: 0.7683 - fus_accuracy: 0.8611 - opt_loss: 0.0707 - sar_loss: 0.1268 - f
usion_loss: 0.0771 - loss: 0.2745 - val_opt_accuracy: 0.8588 - val_sar_accuracy: 0.8
005 - val_fus_accuracy: 0.8633 - val_opt_loss: 0.1195 - val_sar_loss: 0.1142 - val_f
usion_loss: 0.1205 - val_loss: 0.3542

Epoch 00005: val_loss did not improve from 0.29546
Epoch 6/100
108/108 [==============================] - 18s 168ms/step - opt_accuracy: 0.8761 - s
ar_accuracy: 0.7726 - fus_accuracy: 0.8668 - opt_loss: 0.0675 - sar_loss: 0.1228 - f
usion_loss: 0.0733 - loss: 0.2636 - val_opt_accuracy: 0.8621 - val_sar_accuracy: 0.7
996 - val_fus_accuracy: 0.8629 - val_opt_loss: 0.1249 - val_sar_loss: 0.1160 - val_f
usion_loss: 0.1232 - val_loss: 0.3641

Epoch 00006: val_loss did not improve from 0.29546
Epoch 7/100
108/108 [==============================] - 18s 169ms/step - opt_accuracy: 0.8814 - s
ar_accuracy: 0.7815 - fus_accuracy: 0.8717 - opt_loss: 0.0649 - sar_loss: 0.1178 - f
usion_loss: 0.0698 - loss: 0.2524 - val_opt_accuracy: 0.8612 - val_sar_accuracy: 0.8
068 - val_fus_accuracy: 0.8624 - val_opt_loss: 0.1191 - val_sar_loss: 0.1151 - val_f
usion_loss: 0.1197 - val_loss: 0.3539

Epoch 00007: val_loss did not improve from 0.29546
Epoch 8/100
108/108 [==============================] - 18s 171ms/step - opt_accuracy: 0.8855 - s
ar_accuracy: 0.7775 - fus_accuracy: 0.8768 - opt_loss: 0.0612 - sar_loss: 0.1184 - f
usion_loss: 0.0658 - loss: 0.2454 - val_opt_accuracy: 0.8564 - val_sar_accuracy: 0.8
036 - val_fus_accuracy: 0.8580 - val_opt_loss: 0.1244 - val_sar_loss: 0.1138 - val_f
usion_loss: 0.1240 - val_loss: 0.3621

Epoch 00008: val_loss did not improve from 0.29546
Epoch 9/100
108/108 [==============================] - 19s 172ms/step - opt_accuracy: 0.8888 - s
ar_accuracy: 0.7810 - fus_accuracy: 0.8810 - opt_loss: 0.0585 - sar_loss: 0.1162 - f
usion_loss: 0.0627 - loss: 0.2374 - val_opt_accuracy: 0.8621 - val_sar_accuracy: 0.7
977 - val_fus_accuracy: 0.8645 - val_opt_loss: 0.1220 - val_sar_loss: 0.1147 - val_f
usion_loss: 0.1215 - val_loss: 0.3582

Epoch 00009: val_loss did not improve from 0.29546
Epoch 10/100
108/108 [==============================] - 19s 173ms/step - opt_accuracy: 0.8912 - s
ar_accuracy: 0.7757 - fus_accuracy: 0.8843 - opt_loss: 0.0567 - sar_loss: 0.1194 - f
```

```
usion_loss: 0.0601 - loss: 0.2361 - val_opt_accuracy: 0.8645 - val_sar_accuracy: 0.8
024 - val_fus_accuracy: 0.8656 - val_opt_loss: 0.1351 - val_sar_loss: 0.1278 - val_f
usion_loss: 0.1378 - val_loss: 0.4008

Epoch 00010: val_loss did not improve from 0.29546
Epoch 11/100
108/108 [==============================] - 19s 172ms/step - opt_accuracy: 0.8971 - s
ar_accuracy: 0.7885 - fus_accuracy: 0.8914 - opt_loss: 0.0519 - sar_loss: 0.1129 - f
usion_loss: 0.0552 - loss: 0.2200 - val_opt_accuracy: 0.8600 - val_sar_accuracy: 0.8
047 - val_fus_accuracy: 0.8623 - val_opt_loss: 0.1168 - val_sar_loss: 0.1069 - val_f
usion_loss: 0.1154 - val_loss: 0.3391

Epoch 00011: val_loss did not improve from 0.29546
Epoch 00011: early stopping
```

In [15]:

```
# 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, n_opt_layers)
new_model.build((None,)+input_shape)
adam = Adam(lr = 1e-3 , beta_1=0.9)
loss = weighted_categorical_crossentropy(weights)
new_model.compile(optimizer=adam, loss=loss, metrics=['accuracy'], run_eagerly=True)

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_opt = []
    patch_sar = []
    patch_fus = []
    patch_comb = []

    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
            pred_opt, pred_sar, pred_fus, pred_comb = new_model.predict(np.expand_di
            del patch
            patch_opt.append(pred_opt[:,:,:,1])
            patch_sar.append(pred_sar[:,:,:,1])
            patch_fus.append(pred_fus[:,:,:,1])
            patch_comb.append(pred_comb[:,:,:,1])
            del pred_opt, pred_sar, pred_fus, pred_comb
    end_test =  time.time() - start_test

    patches_pred_opt = np.asarray(patch_opt).astype(np.float32)
    patches_pred_sar = np.asarray(patch_sar).astype(np.float32)
    patches_pred_fus = np.asarray(patch_fus).astype(np.float32)
    patches_pred_comb = np.asarray(patch_comb).astype(np.float32)

    prob_recontructed_opt = pred_reconctruct(h, w, num_patches_x, num_patches_y, pat
    prob_recontructed_sar = pred_reconctruct(h, w, num_patches_x, num_patches_y, pat
    prob_recontructed_fus = pred_reconctruct(h, w, num_patches_x, num_patches_y, pat
    prob_recontructed_comb = pred_reconctruct(h, w, num_patches_x, num_patches_y, pa

    del patches_pred_opt, patches_pred_sar, patches_pred_fus, patches_pred_comb
    np.save(path_maps+'/'+'prob_opt_'+str(tm)+'.npy',prob_recontructed_opt)
    np.save(path_maps+'/'+'prob_sar_'+str(tm)+'.npy',prob_recontructed_sar)
    np.save(path_maps+'/'+'prob_fus_'+str(tm)+'.npy',prob_recontructed_fus)
    np.save(path_maps+'/'+'prob_comb_'+str(tm)+'.npy',prob_recontructed_comb)

    time_ts.append(end_test)
    del prob_recontructed_opt, prob_recontructed_sar, prob_recontructed_fus, prob_re
    #del model
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 [16]:

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

for tm in range (0, times):
    print(tm)
    prob_rec_opt[:,:,tm] = np.load(path_maps+'/'+'prob_opt_'+str(tm)+'.npy').astype(
    prob_rec_sar[:,:,tm] = np.load(path_maps+'/'+'prob_sar_'+str(tm)+'.npy').astype(
    prob_rec_fus[:,:,tm] = np.load(path_maps+'/'+'prob_fus_'+str(tm)+'.npy').astype(
    prob_rec_comb[:,:,tm] = np.load(path_maps+'/'+'prob_comb_'+str(tm)+'.npy').astyp

mean_prob_opt = np.mean(prob_rec_opt, axis = -1)
mean_prob_sar = np.mean(prob_rec_sar, axis = -1)
mean_prob_fus = np.mean(prob_rec_fus, axis = -1)
mean_prob_comb = np.mean(prob_rec_comb, axis = -1)

np.save(path_maps+'/prob_mean_opt.npy', mean_prob_opt)
np.save(path_maps+'/prob_mean_sar.npy', mean_prob_sar)
np.save(path_maps+'/prob_mean_fus.npy', mean_prob_fus)
np.save(path_maps+'/prob_mean_comb.npy', mean_prob_comb)
```

```
0
```

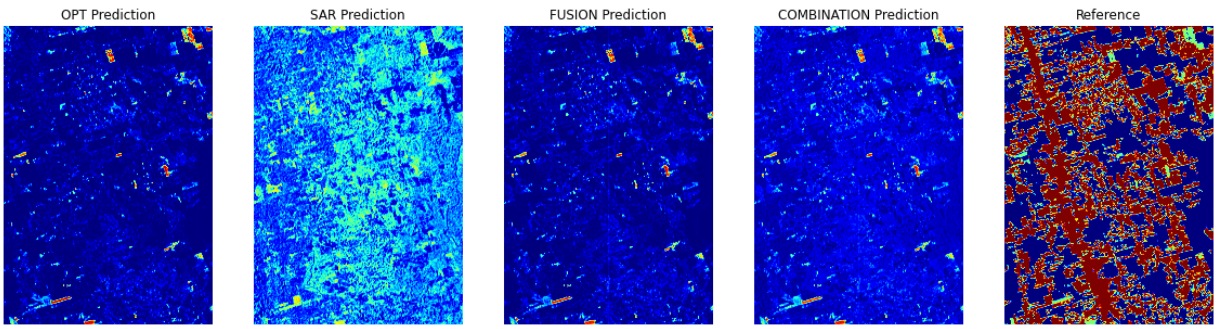
```
1
2
3
4
```

In [17]:

```
# Plot mean map and reference
fig = plt.figure(figsize=(20,10))
ax1 = fig.add_subplot(151)
plt.title('OPT Prediction')
ax1.imshow(mean_prob_opt, cmap ='jet')
ax1.axis('off')

ax1 = fig.add_subplot(152)
plt.title('SAR Prediction')
ax1.imshow(mean_prob_sar, cmap ='jet')
ax1.axis('off')

ax1 = fig.add_subplot(153)
plt.title('FUSION Prediction')
ax1.imshow(mean_prob_fus, cmap ='jet')
ax1.axis('off')

ax1 = fig.add_subplot(154)
plt.title('COMBINATION Prediction')
ax1.imshow(mean_prob_comb, cmap ='jet')
ax1.axis('off')

ax2 = fig.add_subplot(155)
plt.title('Reference')
ax2.imshow(final_mask1, cmap ='jet')
ax2.axis('off')
```

Out[17]:

```
(-0.5, 6999.5, 9999.5, -0.5)
```



In [18]:

```
# Computing metrics
mean_prob_opt = mean_prob_opt[:final_mask1.shape[0], :final_mask1.shape[1]]
mean_prob_sar = mean_prob_sar[:final_mask1.shape[0], :final_mask1.shape[1]]
mean_prob_fus = mean_prob_fus[:final_mask1.shape[0], :final_mask1.shape[1]]
mean_prob_comb = mean_prob_comb[: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 = 10

Pmax_opt = np.max(mean_prob_opt[GTTruePositives * TileMask ==1])
ProbList_opt = np.linspace(Pmax_opt,0,Npoints)

Pmax_sar = np.max(mean_prob_sar[GTTruePositives * TileMask ==1])
```

```
ProbList_sar = np.linspace(Pmax_sar,0,Npoints)

Pmax_fus = np.max(mean_prob_fus[GTTruePositives * TileMask ==1])
ProbList_fus = np.linspace(Pmax_fus,0,Npoints)

Pmax_comb = np.max(mean_prob_comb[GTTruePositives * TileMask ==1])
ProbList_comb = np.linspace(Pmax_comb,0,Npoints)

metrics_opt = matrics_AA_recall(ProbList_opt, mean_prob_opt, final_mask1, mask_amazo
metrics_sar = matrics_AA_recall(ProbList_sar, mean_prob_sar, final_mask1, mask_amazo
metrics_fus = matrics_AA_recall(ProbList_fus, mean_prob_fus, final_mask1, mask_amazo
metrics_comb = matrics_AA_recall(ProbList_comb, mean_prob_comb, final_mask1, mask_am

np.save(path_exp+'/acc_metrics_opt.npy',metrics_opt)
np.save(path_exp+'/acc_metrics_sar.npy',metrics_sar)
np.save(path_exp+'/acc_metrics_fus.npy',metrics_fus)
np.save(path_exp+'/acc_metrics_comb.npy',metrics_comb)
```

```
0.9669155836105346
D:\Ferrari\proj_1\projeto\utils_unet_resunet.py:200: RuntimeWarning: invalid value e
ncountered in longlong_scalars
  precision_ = TP/(TP+FP)
0.8594805187649197
0.7520454539193047
0.6446103890736897
0.5371753242280748
0.42974025938245985
0.3223051945368448
0.21487012969122987
0.10743506484561494
0.0
0.4337931662797928
D:\Ferrari\proj_1\projeto\utils_unet_resunet.py:200: RuntimeWarning: invalid value e
ncountered in longlong_scalars
  precision_ = TP/(TP+FP)
0.38559392558203803
D:\Ferrari\proj_1\projeto\utils_unet_resunet.py:200: RuntimeWarning: invalid value e
ncountered in longlong_scalars
  precision_ = TP/(TP+FP)
0.3373946848842833
0.28919544418652854
0.2409962034887738
0.19279696279101904
0.14459772209326427
0.09639848139550955
0.048199240697754775
0.0
0.9594096422195435
D:\Ferrari\proj_1\projeto\utils_unet_resunet.py:200: RuntimeWarning: invalid value e
ncountered in longlong_scalars
  precision_ = TP/(TP+FP)
0.8528085708618165
0.7462074995040894
0.6396064281463623
0.5330053567886353
0.42640428543090825
0.31980321407318113
0.213202142715454
0.106601071357727
0.0
0.7607574343681336
D:\Ferrari\proj_1\projeto\utils_unet_resunet.py:200: RuntimeWarning: invalid value e
ncountered in longlong_scalars
```

```
  precision_ = TP/(TP+FP)
0.676228830549452
0.5917002267307706
0.5071716229120891
0.42264301909340757
0.3381144152747261
0.25358581145604453
0.1690572076373631
0.08452860381868155
0.0
```

In [19]:

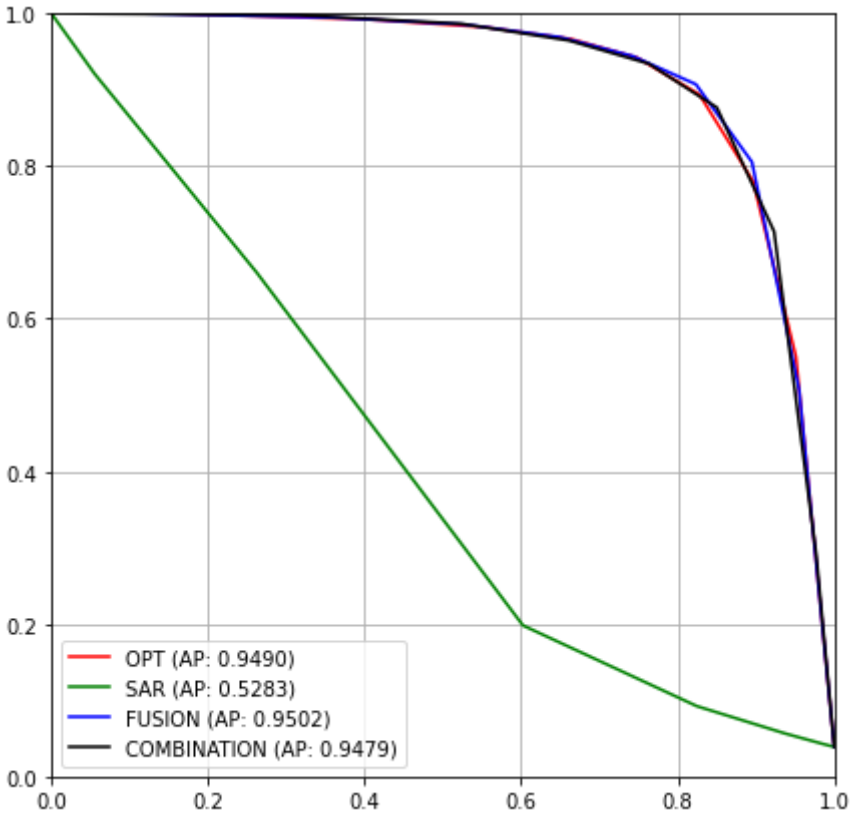
```
# Complete NaN values
metrics_copy_opt = metrics_opt.copy()
metrics_copy_opt = complete_nan_values(metrics_copy_opt)

metrics_copy_sar = metrics_sar.copy()
metrics_copy_sar = complete_nan_values(metrics_copy_sar)

metrics_copy_fus = metrics_fus.copy()
metrics_copy_fus = complete_nan_values(metrics_copy_fus)

metrics_copy_comb = metrics_comb.copy()
metrics_copy_comb = complete_nan_values(metrics_copy_comb)
```

In [20]:

```
# Comput Mean Average Precision (mAP) score
Recall_opt = metrics_copy_opt[:,0]
Precision_opt = metrics_copy_opt[:,1]
AA_opt = metrics_copy_opt[:,2]

Recall_sar = metrics_copy_sar[:,0]
Precision_sar = metrics_copy_sar[:,1]
AA_sar = metrics_copy_sar[:,2]

Recall_fus = metrics_copy_fus[:,0]
Precision_fus = metrics_copy_fus[:,1]
AA_fus = metrics_copy_fus[:,2]

Recall_comb = metrics_copy_comb[:,0]
Precision_comb = metrics_copy_comb[:,1]
AA_comb = metrics_copy_comb[:,2]

DeltaR_opt = Recall_opt[1:]-Recall_opt[:-1]
AP_opt = np.sum(Precision_opt[:-1]*DeltaR_opt)
print('OPT mAP', AP_opt)

DeltaR_sar = Recall_sar[1:]-Recall_sar[:-1]
AP_sar = np.sum(Precision_sar[:-1]*DeltaR_sar)
print('SAR mAP', AP_sar)

DeltaR_fus = Recall_fus[1:]-Recall_fus[:-1]
AP_fus = np.sum(Precision_fus[:-1]*DeltaR_fus)
print('FUSION mAP', AP_fus)

DeltaR_comb = Recall_comb[1:]-Recall_comb[:-1]
AP_comb = np.sum(Precision_comb[:-1]*DeltaR_comb)
print('COMBINATION mAP', AP_comb)

# Plot Recall vs. Precision curve
plt.figure(figsize=(7,7))
plt.plot(metrics_copy_opt[:,0],metrics_copy_opt[:,1], 'r-', label = f'OPT (AP: {AP_o
plt.plot(metrics_copy_sar[:,0],metrics_copy_sar[:,1], 'g-', label = f'SAR (AP: {AP_s
plt.plot(metrics_copy_fus[:,0],metrics_copy_fus[:,1], 'b-', label = f'FUSION (AP: {A
plt.plot(metrics_copy_comb[:,0],metrics_copy_comb[:,1], 'k-', label = f'COMBINATION
```

```
plt.legend(loc="lower left")
ax = plt.gca()
ax.set_ylim([0,1])
ax.set_xlim([0,1])
#plt.plot(metrics_copy[:,0],metrics_copy[:,2])
plt.grid()
```

```
OPT mAP 0.9489764776438415
SAR mAP 0.5283187261232288
FUSION mAP 0.9501819178651769
COMBINATION mAP 0.9478558440097608
```



In [ ]: