Report - Homework 2

Mattia Calabrese, Federico Capaccio

The first step we took towards the realization of our network was trying to fit the network seen at lesson with Professor Lattari for our problem.

The first thing we noticed was that just fixing Professor Lattari's notebook was not enough since the time needed to train the model was too much (it was taking minutes for one image used for training for each epoch).

We decided to use only one epoch to train the model and we could create a model but the score was not really satisfying.

At this point we decided to try to do something similar to what we did with the last challenge, that is splitting images into smaller frames to feed the Transfer Learning network of the notebook with smaller images to speed up the training process.

The pipeline to build the new dataset was: resize each image to a common size (2048X1536) and then split each image into a number of frames dividing each dimension by 256.

Resizing the images gave us the problem that the model gave us masks of the size of the input images and for this reason they were always wrong. For this reason we had to do an "upsampling" of the masks after the prediction to meet the requirements.

By the way, we could not give the net more than one epoch to train since the training took hours.

With this technique we saw an improvement in the score.

After this, we tried to modify a bit the layers of the network by adding a Dense layer to make the network more complex and maybe let it "learn better" but we could not find a way to improve the transfer learning network based on VGG16.

After this we thought that we could use a different model for each dataset so, with the same architecture we built four different models for the different datasets (BipBip, Weedelec, Pead and Roseau).

Thanks to this we improved our model but the last step to our current score was using Unet.

Switching to Unet for some reasons gave us many problems with Python and Keras versions, that's why in the notebook we had to delete Python and reinstall it manually to force Colab to use some versions that worked for our purposes.

We changed VGG16 with Unet with resnet34 as a base model. Thanks to this we did not need any extra layer so by using just Unet we were able to improve the meanIoU.

This meant that we could give the model more epochs to train (it still took hours but was possible), but giving more epochs didn't improve the score, possibly because the network was overfitting the training data since epochs after epochs the loss was improving but the meanIoU was not.

At this point we tried to delete the part where we resized the images and we split them because we thought that maybe Unet was able to train with such images but we could not improve the score (it was actually worse than before), so we sticked with our old process.