**Report Homework2**

For this second challenge, firstly we focused on importing the dataset. We started from the Multiclass Segmentation Notebook of the exercise session, but we had several problems in the image generation and in the application of the read\_rgb\_mask function (we noticed this trying to visualize the images and they were not correct); so we decided to try an approach similar to the one of the 1st challenge: we finally succeed creating "manually" a Numpy array for each set (training, valid and test) of images and masks resized as 512x512 pixels and generating the datasets with the "*from\_tensor\_slices*" function.

Regarding the NN model, we started with a network very similar to the "trivial model" of the exercise class. We experienced difficulties trying to create our first submission, because the predictions were all black (background).

This problem was due to a wrong use of the resize function to reshape the prediction masks to the original dimensions. We finally found a function (*skimage.transform.resize*) from the skimage library that led us to a correct reshape, but only after we noticed that this function was swapping the height and the width of images. Anyway the results with this trivial model were very low, so we implemented a U-Net similar to the one seen in the theory classes, which made the real difference.

Initially, we trained with singular classes of images (e.g. Bipbip Haricot) and then we tried to merge different classes and different combinations of crops (e.g. Bipbip with Weedelec, since they were the most similar ones), but we encountered "out of memory" errors and anyway the training time increased a lot. So, for our last tests, we used singular classes since the results were pretty good (only those regarding the trained class).

For what concerns parameter tuning, we tried to experiment different learning rates, batch sizes (with more than bs = 8 we had problems of memory allocation) and image dimensions. The best dimension we found is the initial one, because a more compact one was not enough and a larger one was too heavy to handle in the model training.

In addition we noticed that changing the data augmentation parameters was not so useful, since it didn’t change in an appreciable way the results.

During our tests, we tried also to use tiling techniques in order to have more precise predictions, but after some attempts we found out that there was something wrong, since the results were very low. So we focused only on the “classic way” of reshaping images.

We experiment the model using all kinds of teams and crops, but we achieved our best result (0,71 of meanIoU on CodaLab) training on Bipbip mais crop dataset.