**Group Tis**

**Manuel Bressan, Federica Maddaloni**

First of all, once we have downloaded the data, we noticed that the validation set was not given. This time, we used the built-in method train\_test\_spli*t* from sklearn.model\_selection to split the training set’s filenames into a training (80%) and a validation one (20%).

Firstly, we trained only on the *Bipbip* dataset, where we merged the *Haricot* and the *Mais* subsets in order to have a bigger number of images to train.

We started by adapting the notebook file created during the fourth lab session as a framework, then we tried changing the number of layers, the batch size, the size of the input image, the learning rate and the other hyperparameters without getting satisfactory results.

Among the different segmentation techniques, we chose to implement a Unet architecture from scratch, following the blueprint presented in the lectures. Then we manipulated it by adding and removing layers arriving to a final model that received in input a 512x512 image. This led to a significant improvement in the val\_mean\_IoU score that reached roughly the 70% always in the *Bipbip* dataset.

We found out that the crop/weed problem is a well-known segmentation task and there were many papers regarding this subject. In many of them, it was explained how the use of additional layers such as ExG, ExGR, VEG can be useful. The first two are indexes that say how much green there is in each pixel, while the last is a derivation of the first ones and refers to vegetation. We added these layers to our images and trained with these, but the results were not as good as expected. Moreover, in all the following models we got better results without these additional layers, so we eventually removed them.

At that point, in order to reach a higher score, we looked for a more efficient implementation of Unet on GitHub and, after trying some of the proposed architectures, we registered an improvement with a Unet with a VGG backbone for the encoding part that received as input a 256x256 image. With this model we eventually reached a score of 0.7217. All the procedures presented up to this point are included in the Bipbip notebook.

We then trained all the four datasets on this architecture, but the other three did not go equivalently well so we turned our attention to another GitHub repository (<https://github.com/qubvel/segmentation_models>) that provided an extensive set of backbones to use as encoder for the Unet, for instance: VGG, ResNet, DenseNet, MobileNet, EfficientNet. There were also many different choices for the losses and we decided to use a total\_loss that took into account both the dice\_loss (i.e. a version of the IoU score designed to be differentiable and therefore fit to be used as a loss) and the focal\_loss (an improved version of the Cross-Entropy loss that weighs the contribution of each sample to the loss based on the classification error).

Among those, the one with the best result was efficientnetb4 and so we employed this model (using pretrained ImageNet weights) on *Roseau* and *Weedelec* obtaining good results (*Roseau*: 0.5906 *Weedelec:* 0.7612; for *Bipbip* it did not outdo the previous model, so we stuck with that).

For the *Pead* dataset the situation was a little different since all the photographs were taken from an angle that included a lot of external grass. So, in order to tackle this perspective issue we had to preprocess all the images and their respective masks: we cut out the grass on the edges and through a cv2 method we squared the resulting trapezoid. With this improvement the final mean\_IoU score of the *Pead* dataset increased from roughly 0.33 to 0.5411.

The model for the *Pead* dataset, the Test prediction and the preparation for the submission are all included in the Pead\_and\_FinalSubmission notebook.