



**POLITECNICO**  
**MILANO 1863**

Artificial Neural Networks and Deep Learning  
Homework 2 - Image Segmentation

Frantuma Elia - 10567359 - 945729,  
Fucci Tiziano - 10524029 - 946638

A.Y. 2020/2021

# Table of contents

# Chapter 1

## Introduction

### 1.1 Description of the task

The homework consists in an image segmentation problem on the proposed dataset. In particular, it is required to segment RGB images to distinguish between crop, weeds, and background, as done in the following example.



(1) dataset image

(2) segmented image

The competition is public and organized by the ACRE (Agri-food Competition for Robot Evaluation).

### 1.2 Dataset

The dataset is composed of images captured by different sensors in different moments and are about two kinds of crops: haricot and maize. Data comes from the 2019 ROSE Challenge where four teams have competed with agricultural robots. Each team has collected images of the same two crops, but in different moments and with different sensors (RGB cameras).

Images in the dataset are divided into different folders based on the team that acquired the image, i.e., Bipbip, Pead, Roseau, Weedelec. For each team, we have two different sub-folders named as the type of crop present in the images, i.e., Haricot and Mais. Finally, for each crop, the captured RGB images (in the Images folder) and the corresponding ground-truth segmentations (in the Masks folder) are provided.

We decided to apply for maize from the Bipbip dataset.

### 1.2.1 Images

Teams' images share most of the properties but differ from image size and file format. The common properties are:

- color space: RGB;
- classes:
  - crop;
  - weed;
  - background;
- number of Training images (per team per crop): 90;
- number of Test\_Dev images (per team per crop): 15;
- number of Test images (per team per crop): 20.

### 1.2.2 Masks

Masks folders contain the ground-truth segmentation for each corresponding (having the same name) image in the Images folder. They have the same exact properties of the Images set apart from the fact that they all have the same file format: PNG.

In each mask, classes are represented by different colors. The dictionary which allows assigning a label to each corresponding color is provided in the `starting_kit`, as well as the example script in which we show how to transform RGB masks into target masks. In the following, the provided dictionary (`'RGBtoTarget.txt'` in the `starting_kit`):

- RGB: 0 0 0 - Target 0 (background);
- RGB: 254 124 18 - Target 0 (background);

- RGB: 255 255 255 - Target 1 (crop);
- RGB: 216 67 82 - Target 2 (weed).

### 1.2.3 Data augmentation

We have performed data augmentation in order to increase the dataset dimension. Some of the parameters used to perform the transformations are: rotation, zoom, horizontal/vertical shift and flip.

### 1.2.4 Tiling

During the training, we have experienced many issues with the RAM and VRAM. This happened both with Colab Pro and local tests. We decided not to downsize the dataset images, since the quality loss would have been too much, so we performed image tiling. In the final version, we splitted the original images into 8 tiles per image.

## 1.3 Validation set

No automatic validation set is provided. This means that a subset of the training set must be used to perform validation.

In our case, we parametrized the number of training images to be moved into the validation set, with a 10% probability.

## 1.4 Test set

Test\_Dev images are provided without any ground-truth mask. Participants are required to provide the segmentations for the Test\_Dev images by submitting the solution with the correct submission format.

## 1.5 Evaluation

Submissions are evaluated on the mean Intersection over Union (IoU) obtained on the two classes, crop and weed. IoU is computed for each target class (crop and weed) separately, by considering prediction and ground truth as binary masks. Then, the final IoU is computed by averaging the two.

## Chapter 2

# Neural network architecture

For all the networks, we have modified the function `read_mask_example`. The modified function can be found into the `starting_kit` folder. The notebook expects that folder to be in `\content\drive\MyDrive`.

### 2.1 U-Net

Our first try was with U-Net, but the results turned out to be very poor. In particular, the high number of parameters made the tuning very difficult to perform. The training time was simply too much to experience some progress, since it took up to one hour and a half for one epoch. After some epoch, the result was the one shown below.

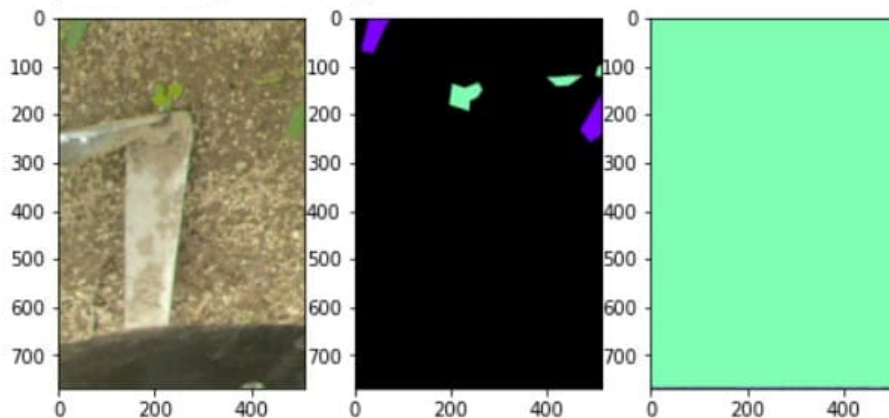


Figure 2.1: dataset image, target segmentation, actual segmentation

We tried to change the learning rate, the weights initialization and the

loss function, using a weighted one, but the situation remained the same.

## 2.2 Transfer learning with MobileNetV2

Then, we tried to use MobileNetV2, as it was suggested by the TensorFlow tutorial. After a while, we noticed that the trained model could not distinguish the maize plants from weeds. The result was a well defined foreground, but with a faded violet, since the two colours were mixed in correspondence of the plant. This was probably caused by the incompatibility of the weights with non-squared images, as shown by Jupyter in a warning.

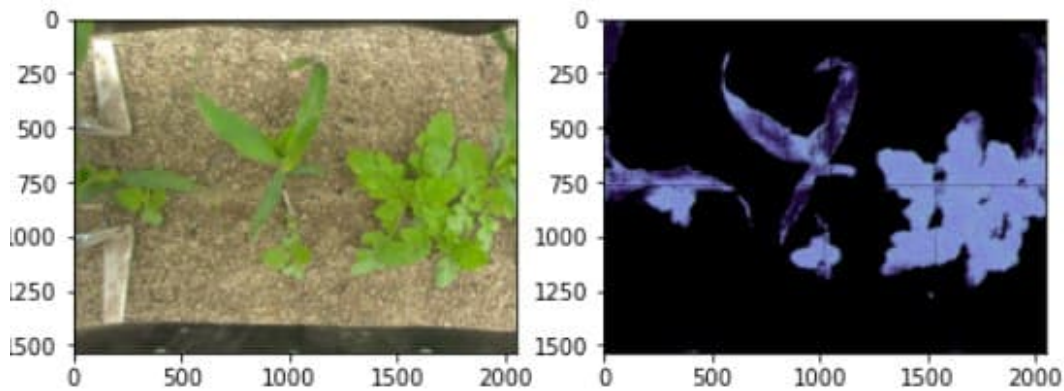


Figure 2.2: image segmentation

## 2.3 U-Net, with VGG as a backbone

### 2.3.1 Without skip connections

This architecture is taken from the notebooks seen during the exercise sessions of the course, adapted to this competition. After some epochs, the results were similar to the one shown below.

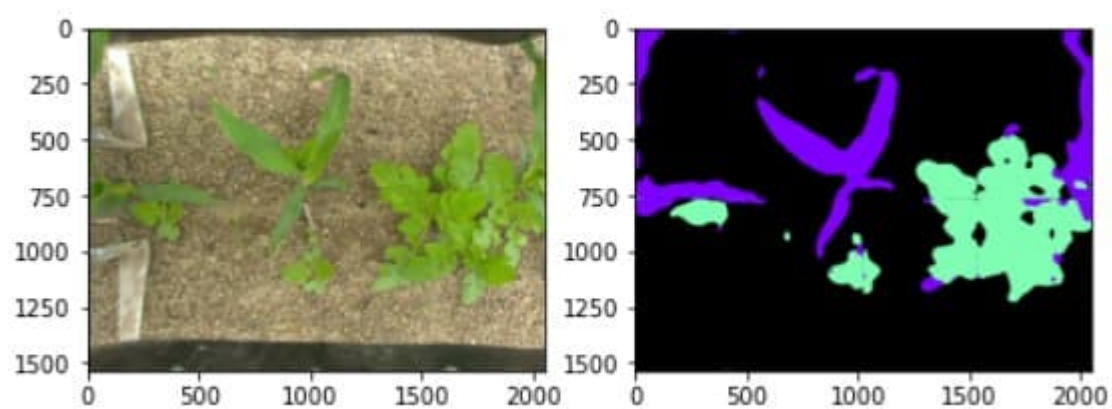


Figure 2.3: image segmentation



### 2.3.2 With skip connections

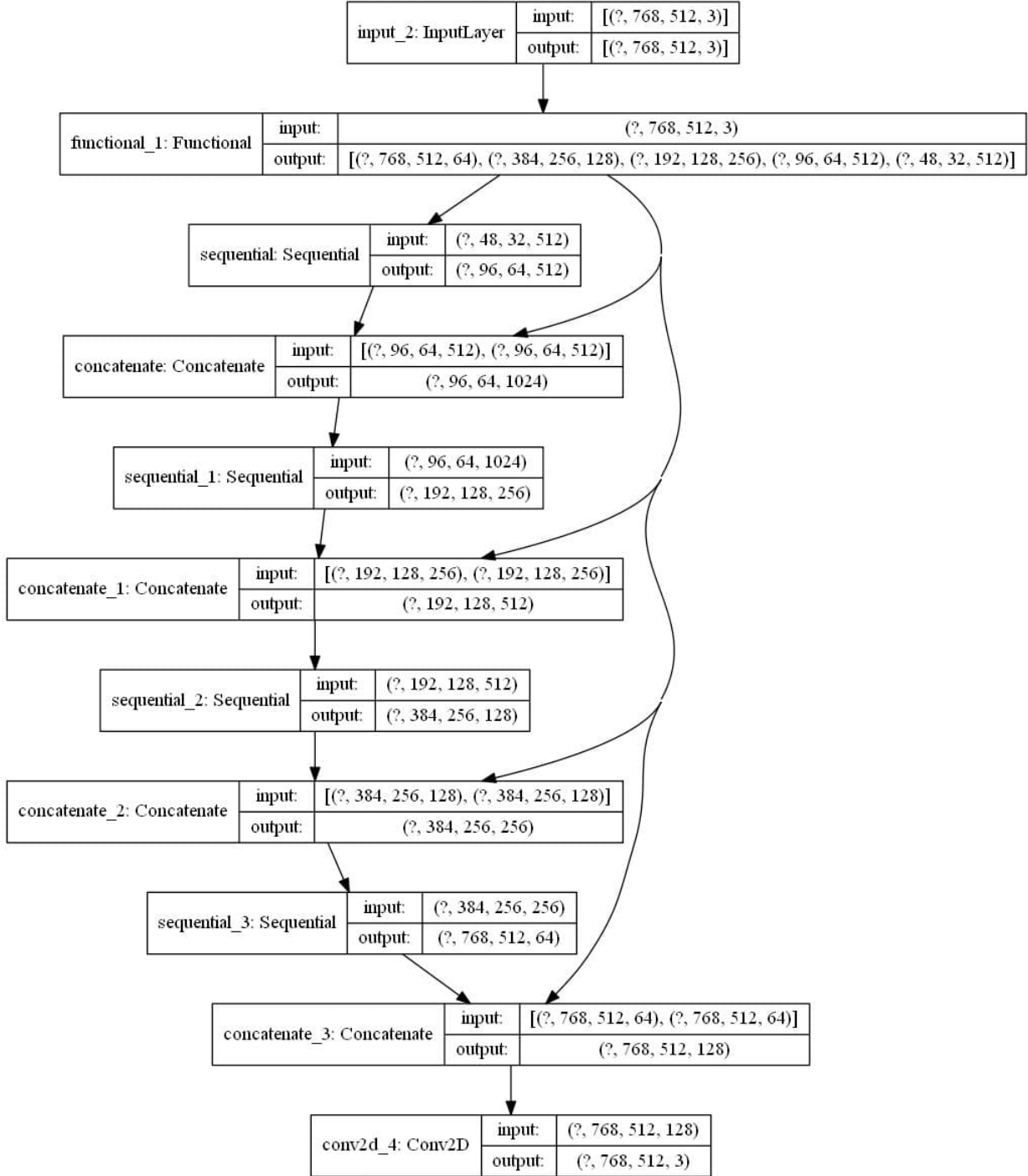


Figure 2.4: network obtained with VGG and skipped connections

After considering the former architecture, we decided to add skip connections in order to improve the performance of the upsampling layers. This led to a better definition of shapes and leaves and thus to a higher score.

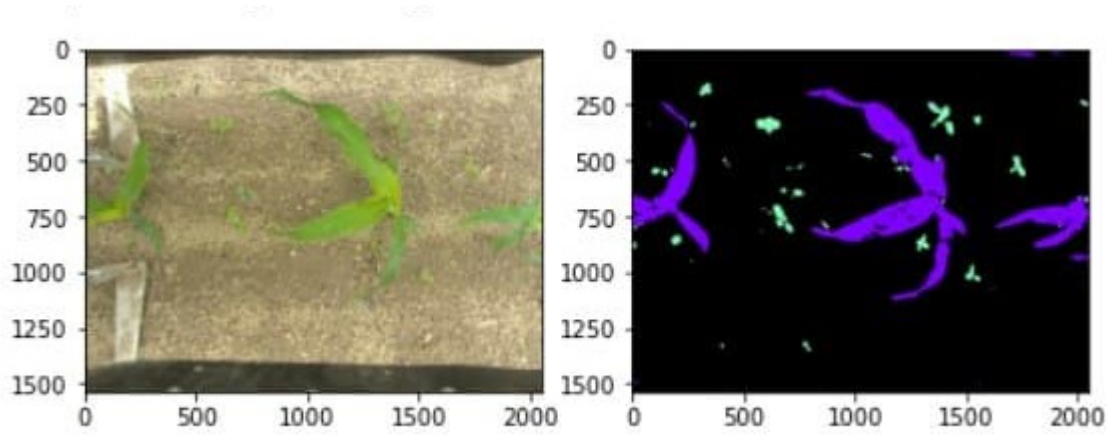


Figure 2.5: image segmentation

### 2.3.3 Score

The best score of the network was 0.7207, obtained with the skip connections.

# Chapter 3

## Main issues

During this competition we had to face problems that didn't occur in the first one: despite we could solve memory capacity issues by tiling the images, the main problem remained time. The classic network tuning procedure has been hardened by the great training time, up to one hour per epoch (obtained with U-Net). Even with the other network architectures, the training time was so long that it took hours, or days, to understand if the network was properly learning the model. This was the main reason because we used the Maize dataset only.

The tiling procedure took so long (8-10 hours) that we could not make many attempts on how many tiles were better. Furthermore, we had to manually fix the folder structure in the .zip file, since Colab didn't allow us to recreate the original dataset structure.

Other factors that limited our score were the absence of the last softmax layer and the TensorFlow preprocessing of predictions for VGG, that drastically lowered the performance of the network. We managed to correct these two mistakes just in the last days.

# Chapter 4

## References

### 4.1 Links

- GitHub repository of the project: <https://github.com/tizianofucci/A2NDLSegmentation>
- Competition web page: <https://competitions.codalab.org/competitions/27176>