Plant detection : README

Misrachi Laura

I. Dependencies

The code, which contains three files: *PlantExtractor.py*, *DataLoader.py* and *main.py* should be run with the following library dependencies:

- OpenCV : 2.4.11
- Python : 2.7
- Keras : 2.1.5

II. Algorithm overview

The code is fairly well commented but here are the main steps of the plant detection algorithm that was set up:

1. From the initial 3-channel image, we compute the associated ExG-ExR grayscale image, based on the vegetative index ExG-ExR described in the article that was mentioned in the statement.
2. The grayscale image is then binarized according to the zero threshold highlighted in the article.
3. Erosion and dilatation are then applied to further improve the markers of plant-region by removing noise (erosion) and amplifying the region of interest (dilatation).
4. The associated bounding boxes are found using the *findContours* function of OpenCV.

III. Results and comments

The results for the first image are quite good. Without surprise, the 4 plant-regions were detected, with a relatively correct localization of the bounding boxes. Notably, the shadow of the plant on the soil is not fully included in the bounding box, which is satisfactory. Also, as one could verify, the erosion step enables the removal of some noisy regions in the ExG-ExR binary image.

The results of the second image, which illustrates a more complex case of plant detection, are less impressive. Even though most of the regions containing plants were detected in some way, (with bounding boxes being plotted), one can easily spot the lower accuracy in the bounding boxes localization. Also, it is quite frequent that a plant that should have been detected as one, was detected with more than one separated bounding boxes. I tried to avoid this issue by using the cv2.RETR_EXTERNAL parameter in the *findContours* function but, it calls, I guess, for a better process to link neighboring markers of plant-region in the binary image. It is also important to note that some plant-regions were not highlighted in the ExG-ExR grayscale image, which shows some limitation of the ExG-ExR vegetation index.

The third image highlighted the following issues with our algorithm: bounding box precision in localization, unwanted region detected due to shadows in the image, tiny plant-region detection issues. But still, the ExG-ExR vegetation index seems fairly robust to brightness, since a significant plant-region localized in the shadow region was correctly detected by our algorithm.

Finally, I would like to state that, given only the three images we were provided with, it is particularly difficult not to specify too much our algorithm to this very small dataset, in order to be able to create an algorithm that would generalize well on unseen data. Notably, I tried several settings for the *dilatation_kernel_size* parameter (15, 20 and 30) and kept 20, that gave an eyeballed acceptable balance between false positives (detecting regions that are not plant) and false negatives (missing plant regions). Indeed, in this application our final aim is to minimize/reduce the proportion of herbicide spray to the very necessary regions. We therefore want to detect all the regions that contain plants (and thus achieve a low level of false negatives at first) and then ultimately minimize the false positive rate. Of course, with a larger dataset, the knowledge of ground truth labels and with the use of appropriately defined metrics, parameters would be more adequately selected.