Assignment 1: Multi-Object Detection with CNNs

Course held by: Prof. Giuseppe Serra, Dr. Ali Abdari,
Dr. Beatrice Portelli, Dr. Alessandro Locaputo

November 29, 2023

DEADLINE
Wednesday December 13th, 2023, 23:59

1 Instructions

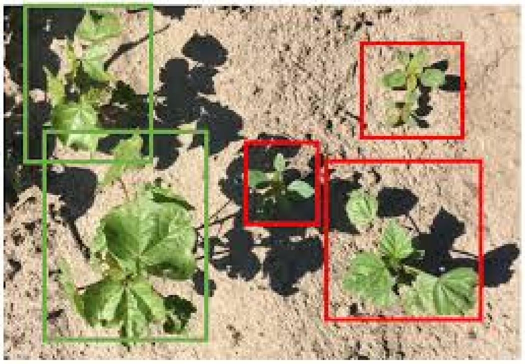
1.1 Submission instructions

- you can do this assignment either alone or in group (at most 3 students)
- please submit no more than *one submission* per group
- alongside your code, please submit a *short report* (in .pdf), containing a written description of what you have done, including used methods, results, discussion, conclusions and eventual references
- indicatively, the report should not exceed 3 pages of text (images and tables don't count towards this limit)
- in case of multiple source code files please summarize the content of each .py/.ipynb file in the report
- put both the source code and the report in a .zip file named after your student id: "idstudent.zip" for individual works and "idstudent1_idstudent2_idstudent3.zip" for group works (e.g. "012345_678901.zip" or "012345_678901_124311.zip")
- the .zip file must be uploaded to the appropriate folder of the e-Learning page of this course no later than the specified deadline
- any submission that will be submitted after the deadline won't be evaluated.

1.2 Source code instructions

- be sure to write your code in Python 3
- you can use the libraries seen in class (pandas, sklearn, numpy, matplotlib...). If you choose to use additional libraries, please mention them in the report and explain why you chose them. While it is highly recommended to use the PyTorch framework, you are free to use other machine learning frameworks. If you encounter any issues with PyTorch, feel free to ask for help; we can assist you in resolving any problems that may arise.
- please comment your code. The code can be presented as either a .py (python source code) or a .ipynb (Jupyter Notebook) file
- if you have more than one source code file please name them so that it is easy to understand what's inside each file and which file contains the code to run your work
- alongside your code, please submit a *requirements.txt* file which specifies the versions of the used libraries (you can use pipreqs https://github.com/bndr/pipreqs to automatically create a minimal clean requirements file). Alternatively, make sure that the code can be run on Google Colaboratory.

2 Assignment



In this assignment, you will become familiarized with CNNs, and you will develop a solution for a **multi-object detection problem**.

Given a dataset of images containing plants, you will need to implement a model based on CNNs, to predict the **position** and the **type** of the plants. This involves binary classification, distinguishing between crops and weeds.

You will analyze the role and effect of different architectures and parameters of your CNN model (e.g. number of layers, filters, pooling, normalization, etc.), and report the correct evaluation metrics and observations in a report.

In a nutshell, you have to:

- you should use the dataset present in this repository: `https://github.com/cropandweed/cropandweed-dataset/tree/main`. Also there is a `materials.zip` file in the e-learning platform. First, you should clone the repository and download the mentioned file. Then replace the `setup.py` file in `.../cropandweed-dataset/cnw/setup.py` with `setup.py` file existing in the `materials.zip`. Now run the new `setup.py` to download the dataset.
- Once the process finished you should have a `.../data` directory in which you can see a couple of directories. But you are supposed to use the directory of `.../data/images` as source images and `.../data/bboxes/CropOrWeed2` as their related bounding boxes information. Furthermore, you should consider the two other files in materials.zip, including `train_split.txt` and `test_split.txt` containing the images and their related annotations for the train and test stages. Use the training samples to train your model (part of it can be used as a validation set), and the test set to test it and report its performance. The information regarding the annotations has been explained in the Github repository.
- Remember that transformations can be applied to your images to improve their representation (e.g. cropping, contrast, saturation changes)
- You have to design the architecture of your model. Building your CNN model is one of the most important issues for this task, and you should try different architectures and configurations, including a different number of layers, and filters (e.g. check the Conv2D function from `torch.nn`, which provides many options for configuring CNN parameters). You can use a Sequential model, for which you can specify the types of desired operations after each convolutional layer, including batch normalization and the activation function (e.g. ReLU)
- You need to select a loss function, and an optimization module, along with a learning rate. For the task of object detection, you can find some basic losses online[1]. Don't stress yourself by spending too much time looking for fancy loss functions, it isn't the main goal of the assignment.

[1] `https://ambolt.io/how-object-detectors-learn/`

- Train and test your CNN model using the usual pipeline we have seen during the practical lessons. Please try different architectures and parameters and then, based on the performed experiments, select a final model. **The decisions made and the rationale behind them will be explained in the report** (e.g. why did you use specific filter sizes and how did it help at improving the accuracy?). Please add diagrams of the architectures you designed for this task (we suggest using tools like draw.io for this)
- to measure the performance of the model on the test data, please use methods based on Intersection-over-Union to calculate the precision of the bounding boxes (see `https://blog.mturk.com/tutorial-measuring-the-accuracy-of-bounding-box-image-annotations-from-mturk-ad3dfcdf8aa0`). Use accuracy, precision, and recall to measure the performance of the task of crop / weed classification
- in any case, display some relevant results of correct and/or incorrect predictions (the image with the bounding boxes, like in the figure above)
- **formulate your critical reflections about the performed experiment and the lessons learnt** (e.g. what is important when constructing a CNN model and how to optimize it?).

Remember that you can optionally compare your model with a pre-trained one like VGG. However, the goal is *not* to achieve state-of-the-art results but to learn how to configure a good CNN model.

Good luck!

2.1 Contacts

For any problem or doubt about the assignment, you can contact us. Send your question to one of these addresses, but be sure to **always CC all of us** to ensure a response. Our emails are:

- abdari.ali@spes.uniud.it
- portelli.beatrice@spes.uniud.it
- locaputo.alessandro@spes.uniud.it