COMPUTER VISION – June & July project

Automatic tree detection and recognition

The world of agriculture and farming is increasingly adopting high-end technological solutions to automatize several tasks – see this link if you are interested in some insights:

https://www.engineering.com/DesignerEdge/DesignerEdgeArticles/ArticleID/16653/Smart-FarmingAutomated-and-Connected-Agriculture.aspx

The project is divided into two parts. Students willing to work alone should develop part 1 only (they are not allowed to develop part 2 only). Students who work in a 2-person group should develop parts 1 and 2. You are free to design your solution as you prefer, using the tools and algorithms discussed during the course and/or introducing new ones (but you are allowed to use only the OpenCV library + Keras environment with Colab). If you use Machine Learning (ML), you are not allowed to re-use systems or network already trained by others – you need to train them and report about the training phase.

Part 1

You are asked to develop a system that is capable of automatically detecting trees in an image by creating a bounding box around each one. In order to be recognized as a tree, it should be clearly visible and evident – small trees in the background, grass etc. do not need to be detected.

There is an additional constraint in part 1: you cannot base your solution on an end-to-end learning – you need to perform some image manipulation using non-ML based techniques. You are allowed to couple such techniques with a ML-based component. If you employ ML, it is your duty to find a suitable dataset and organize the training, and to detail the classifier that you employed, the dataset you used, the details of the training history and the performance level.

You should comment the performance of your algorithm on the benchmark dataset that is available on moodle (not all the images include a tree). If you use ML, **you are not allowed use the images in such dataset in the training set**. To make your approach more robust, you are free to add further images to the benchmark dataset and to comment the performance of your approach on the additional images – this is appreciated.

Part 2

For this second part you should develop a tree and herb species classification system. The starting point is the Plant Identification 2013 contest, that you can find at this link:

https://www.imageclef.org/2013/plant

At the beginning of the page you can find the links to download the training and test set. The whole dataset is huge and contains several different species. However, you are asked to work on a reduced version of the dataset, choosing only 5 classes. The dataset is composed of images and corresponding XML files with the description. Such descriptions are very rich, and you may neglect most of the data. You should only take care of the ClassId, that is the ID to be used as a ground truth ("the class label that must be used as ground-truth. It is a non-official short name of the taxon used for easily designating a taxon", according to the documentation).

The documentation explains that there are five categories of images, namely leaf, flower, fruit, stem, entire. Leaves are available both with a natural background and with an artificial background (a sheet). You should choose one of them and, within such category, choose five species (five ClassIds) that your classifier should learn to distinguish.

In this second step, you are free to choose to exploit ML or not. If you do, you should report:

- The choices you did about category and species
- The topology of the network
- The details of the training process (including the training accuracy history)
- The final results of your network

Submission

You should submit:

- Your code (both C++ and Python)
- A report describing the details of the approach you developed (no page limit, but the evaluation will depend on the content, not on the number of pages!) – the report should include the details outlined above
- The performance level
- Part 1: the results on all the samples in the benchmark dataset (and, in case, its extension)
- Part 2: selected results on the images in the test set (at least 10)
- The analysis of critical cases and of the images in which you get wrong results (if any)

The C++ project should be provided with the CMakeLists.txt file. No executables nor object files are needed. You should also submit the trained network – if it is too large to be uploaded on moodle, please share on some online services (e.g. dropbox, google drive, …). You can also attach your Colab notebook to the submission.

If you worked in a 2-student group, you should clearly specify in the report how you distributed the workload. Also, you should specify the author of each source file. If both of you need to work on the same source, you should specify this at the beginning of the file and separate your code into two groups, identifying the author of each one. Please note that clearly uneven distributions of the workload will result in different marks.

Evaluation criteria

The project will be evaluated based on the approach used, the results obtained, the critical analysis of the results. Your code may be tested on additional unknown images, and the results obtained are part of the evaluation. A minor weight will also be given to the quality of the code (e.g., all the code in the main function will result in a penalty).