Project: Image Analysis 2015/2016 Anomaly Detection for Crop Monitoring: User manual

Stefano Cerri 849945

June 10, 2016

Contents

Chapter 1

Purpose

The purpose of this document is to help the user understand how to replicate the experiment step-by-step.

1.1 Prerequisites

1.1.1 Configuration

The configuration used in the project is:

- Ubuntu 15.10
- Matlab 2016a
- MatconvNet 1.0-beta18

It works also with Mac OS, but have some problems with Windows (in particular with Windows 10).

1.1.2 Download the material

The code, the dataset and MatConvNet library can be donloaded at this link: `https://github.com/ste93ste/cwfid_classification`.

Chapter 2

Matlab code

All the matlab code can be found in: cwfid_classification/MatconvNet-1.0-beta18/matlab code/cwfid_classification.

2.1 cnn_cwfid _init _*number.m

There are six different files with this name (the only part that changes is the final number). These files are functions that creates different networks. You can find the description of the six types of network in the report.

2.2 cnn_cwfid.m

This function is the core of the project. It calls the function cnn_cwfid _init _*number.m described before and it then calls the function cnn_train.m to train the network. If you want to change and use a different network you can modify line 36. Network 5 is the one that has best results and the default network.

```
net = cnn_cwfid_init_5('batchNormalization', opts.batchNormalization, ...
                       'networkType', opts.networkType) ;
```

If you change the network, remember to delete the previous training folder because otherwise the function will not train another network (it reuses the precedent training results). The folder to be removed can be found at the path: cwfid_classification/MatconvNet-1.0-beta18/data/ and its name is: cwfid-baselinesimplenn-bnorm.

The function have also some attributes that can be modified such as:

- vl_compilenn('enableGpu', true); if you want to enableGpu. This can make the training faster but it needs to have a dedicated and configured GPU. Default is false

- opts.networkType = 'dagnn' ; if you want to use dagnn network instead simplenn. Dagnn were not used in this project because they have similar or worse results comparing to simplenn network

2.3 cnn_cwfid_evaluation_simplenn.m

This script evaluates the network by computing the accuracy and the confusion matrix. If no network was trained before it trains first the network and then evaluates it. There is also the correspectively script for dagnn

2.4 plot_accuracy.m

This script plots the accuracy of the six networks. The accuracy of the networks were manually inserted after compute the script cnn_cwfid_evaluation_simplenn.m for the specific network.

2.5 plot_accuracy_number_of_maps.m

This script plots the accuracy of the different configurations of number of maps for network 5. The accuracy of the configurations were manually inserted after compute the script cnn_cwfid_evaluation_simplenn.m for the specific configuration.

2.6 classUmbalancingThreshold.m

This script count the percentage of frequency in the training set of crop,weed and ground used then in the slidingWindow.m script.

2.7 slidingWindow.m

This script uses the sliding window approach for all the images in the testing set and then calculate the accuracy. The value of the stride can be changed at line 8 (default is 10)

```
%define the stride of the sliding window
stride = 15;
```

Chapter 3

Appendix

3.1 Repository

All the code, the dataset, the intermediate results and other documents can be found at this repository https://github.com/ste93ste/cwfid_classification

3.2 Reference document

- http://rd.springer.com/chapter/10.1007%2F978-3-319-16220-1_8

3.3 Software and tool used

- LaTeX (http://www.latex-project.org/) : to redact and to format this document
- Matlab R2016a (http://uk.mathworks.com/products/matlab/) : to compute and evaluate the network
- MatConvNet (http://www.vlfeat.org/matconvnet/) : MATLAB toolbox implementing Convolutional Neural Networks (CNNs)