

# Brainsignals Submission to Plant Identification Task at ImageCLEF 2012

Cristian Grozea<sup>1</sup>

Fraunhofer FOKUS, Berlin Germany  
cristian.grozea@brainsignals.de

**Abstract** This article describes our first participation to the plant identification task of ImageCLEF. We have approached it as intended with the least possible computer-vision related techniques (especially without costly image preprocessing and feature extraction) and as much as possible machine learning techniques (random forests). Our method is purely visual and entirely automatic, using only the image information. One should mention that the total time spent with preparing this submission was only about one week. The results were accordingly fairly poor, but probably usable for mobile applications, as the whole process of preprocessing and classification is very fast. We propose a slight modification to the Otsu algorithm, which makes it better suitable for the scan-like images.

## 1 Introduction

Our intention with this participation to ImageCLEF, the plant identification task, was to test our hypothesis that good machine learning techniques paired with rudimentary computer vision preprocessing could lead to fair/good results.

## 2 Methods

### 2.1 Preprocessing

The images are converted to grayscale, resized to 30% and then binarized with a slightly modified Otsu algorithm. Our algorithm gives unequal weights to the variances of the two classes when looking for the best split in the grays values space. We used that to tune the split towards being more inclusive with the leaf pixels class, which we have empirically found it works better than the standard Otsu algorithm on the scan-like images. This choice seemed to have paid back, as in the final results in the competition the performance of our algorithm degrades less on the scan-like images versus the one on the scan images than that of the algorithms of some other teams - which has the effect of pushing it to higher ranks.

**Features** Two types of features are extracted (for a total of 344 features):

- Local features: histograms of content types of small rectangular patches of bits, with and without sub-sampling.

- Global shape features from lateral projections: histograms of the number of b/w alternations on every pixel line/column.

The features we extract are thought to be extremely quick to compute and still give a good description of the content of the binarized image.

While the second class of features is self-explanatory, the first class needs perhaps some more detail. Let us assume a small window, say a square of 2 by 2 pixels. When this window is moved over the image, it contains several patches of 2x2 pixels. The content of those patches can be linearized as a 4 bit vector and has only 16 possible types of content, which are the binary representations of the numbers from 0 to 15. It is then natural and easy to compute the distribution (histogram) of those numbers. The same can be repeated for non-square windows, say of size 2x3, 3x2 or 3x3 and even generalized to binary masks where the bits collected are farther away from each other, disposed or not in a regular-step grid. Attempting to capture local and non-local topologies, we have used masks with distance between pixels of up to 30 pixels, but always few possible values for the selected content (sparse masks).

The discriminative power of those features, at least for some of the classes, can be observed in a plot of the data after dimensionality reduction by PCA in Figure 1.

## 2.2 Classification

A random forest classifier [1], with 100 trees and standard number of selected features ( $1 + \sqrt{m}$ ) is trained on the training data, where  $m$  is the total number of features. The model obtained is then applied on the test data.

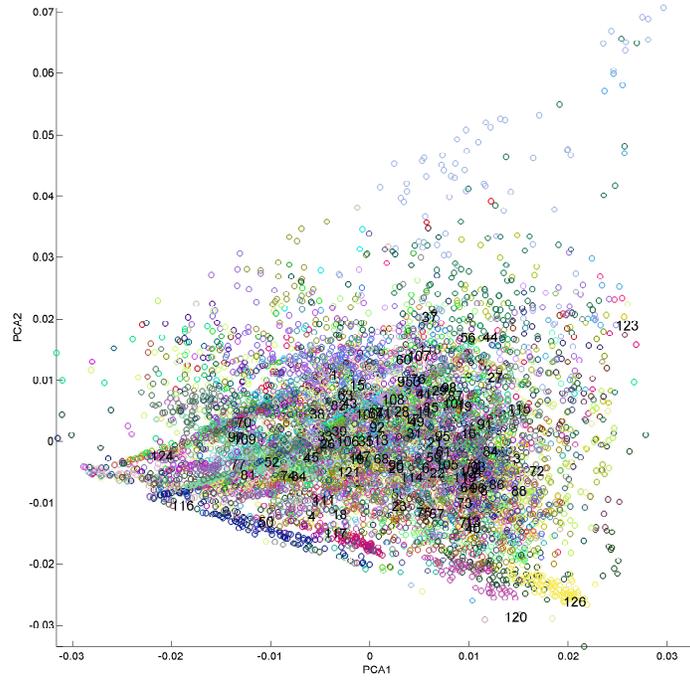
## 3 Results

The 5-fold cross-validated classification accuracy on the training data set was 70%, which is very good for so many classes (126). It could be that the performance can be increased even more by using more trees in the random forest and/or by using weighting or balancing techniques to balance the samples similar to the scoring function.

Although the final test performance, as measured with the complicated balanced error function was certainly not up to the best (scans: 0.25, scans-like: 0.22, photographs: 0.05, average: 0,17), the method seemed to perform fine on the training data, failing most of the time only on images for which, after resizing and binarization, it was difficult for a human as well to spot that the class of the image should be different than the class of the random member of the wrongly misclassified-to class, as shown in Figure 2

## 4 Conclusion

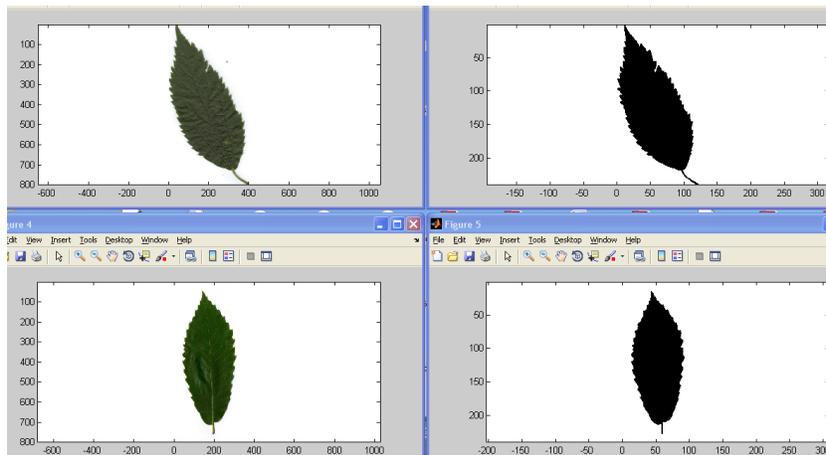
While our solution was not performing up to the best ones in this competition, it is fast, accurate enough for non-critical applications and has potential for improvement. We intend to proceed developing this solution further and plan a mobile phone implementation of it.



**Figure 1.** Class clustering in the space of the selected features

## References

1. Breiman, L.: Random forests. *Machine learning* 45(1), 5–32 (2001)



**Figure 2.** Example of mis-classification; humans have a hard time as well trying to distinguish those two.