Participation of LSIS/DYNI to ImageCLEF 2012 plant images classification task

Sébastien Paris[1] *, Xanadu Halkias[2], and Hervé Glotin[23]

[1] LSIS/DYNI, Aix-Marseille University
sebastien.paris@lsis.org
[2] LSIS/DYNI, University of South Toulon-Var
halkias@univ-tln.fr
[3] Institut National de France
glotin@univ-tln.fr

Abstract. This paper presents the participation of the LSIS/DYNI team for the ImageCLEF 2012 plant identification challenge. ImageCLEF's plant identification task provides a testbed for the system-oriented evaluation of tree species identification based on leaf images. The goal is to investigate image retrieval approaches in the context of crowd sourced images of leaves collected in a collaborative manner. The LSIS/DYNI team submitted three runs to this task and obtained the best evaluation scores ($S = 0.32$) for the "photograph" image category with an automatic method. Our approach is based on a modern computer vision framework involving local, highly discriminative visual descriptors, sophisticated visual-patches encoder and large-scale supervised classification. The paper presents the three procedures employed, and provides an analysis of the obtained evaluation results.

Keywords: LSIS, DYNI, ImageCLEF, plant, leaves, images, collection, identification, classification, evaluation, benchmark

1 Introduction

This paper presents the contribution of the LSIS/DYNI group for the plant identification task that was organized within ImageCLEF 2012[4] for the system-oriented evaluation of visual based plant identification. Similar to the ImageCLEF 2011 challenge, this second year pilot task was also precisely focused on tree species identification based on leaf images. This year, the challenge was organized as a classification task over 126 tree species with visual content being the main available information. Three types of image content were considered: leaf "scans", leaf photographs with a white uniform background (referred to as "scan-like" pictures) and unconstrained leaf "photographs" acquired on trees with natural background (see Fig. 1). The LSIS/DYNI team submitted three

* Granded by COGNILEGO ANR 2010-CORD-013 and PEPS RUPTURE Scale Swarm Vision

[4] http://www.imageclef.org/2012/plant

runs, all of them based on local feature extraction and large-scale supervised classification. We obtained the best score for the "photographs" category with an automatic method ($S = 0.32$).

Fig. 1. From left to right: "scans", "scan-like" and "photographs" category.

2 Task description

The task has been evaluated as a plant species retrieval task.

2.1 Training and Test data

A part of Pl@ntLeaves II dataset was provided as training data whereas the remaining part was used later as test data. The training subset was built by including the training AND test subsets of last year's Pl@ntLeaves I dataset, and by randomly selecting 2/3 of the individual plants for each NEW species (several pictures might belong to the same individual plant but cannot be split across training and test data).

- The training data is comprised of 8422 images (4870 "scans", 1819 "scan-like" photos, 1733 natural photos) with full xml files associated to them (see previous section for few examples). A ground-truth file listing all images of each species was provided complementary.
- The test data is comprised of 3150 images (1760 "scans", 907 "scan-like" photos, 483 natural photos) with purged xml files (*i.e.* without the taxon information that has to be predicted).

2.2 Task objective and evaluation metric

The goal of the task was to retrieve the correct species among the top k species of a ranked list of retrieved species for each test image. Each participant was allowed to submit up to 4 runs built from different methods. As many species as possible can be associated to each test image, sorted by decreasing confidence score. However, only the most confident species were used in the primary evaluation metric described below. Providing an extended ranked list of species was encouraged in order to derive complementary statistics (*e.g.* recognition rate at other taxonomic levels, suggestion rate on top k species, *etc.*).

The primary metric used to evaluate the submitted runs was a normalized classification rate evaluated on the 1st species returned for each test image. Each test image is attributed with a score of 1 if the 1st returned species is correct and 0 if it is wrong. An average normalized score is then computed on all test images. A simple mean on all test images would indeed introduce some bias with regard to a real world identification system. Indeed, we recollect that the Pl@ntLeaves II dataset was built in a collaborative manner; So that few contributors might have provided much more pictures than many other contributors who provided few. Since we want to evaluate the ability of a system to provide correct answers for all users, we would rather measure the mean of the average classification rate per author. Furthermore, some authors sometimes provided many pictures of the same individual plant (to enrich training data with less efforts). Since we want to evaluate the ability of a system to provide the correct answer based on a single plant observation, we also decided to average the classification rate on each individual plant. Finally, our primary metric was defined as the following average classification score S:

$$S = \frac{1}{U} \sum_{u=1}^{U} \frac{1}{P_u} \sum_{p=1}^{P_u} \frac{1}{N_{u,p}} \sum_{n=1}^{N_{u,p}} s_{u,p,n}, \tag{1}$$

where

- U : number of users (who have at least one image in the test data)
- Pu : number of individual plants observed by the u^{th} user
- Nu, p : number of pictures taken from the p^{th} plant observed by the u-th user
- $s_{u,p,n}$: classification score (1 or 0) for the n^{th} picture taken from the p^{th} plant observed by the u^{th}h user

Finally, to isolate and evaluate the impact of the image acquisition type ("scans", "scan-like", "photograph"), a normalized classification score S was computed for each type separately. Participants were therefore allowed to train distinct classifiers, use different training subsets or use distinct methods for each data type.

3 Description of used methods

For all submitted runs, whatever the particular image type, we followed the same pipeline: i) feature extraction coupled with spatial pyramid (SP) for local analysis and a linear large-scale supervised classification. For our first participation, we didn't performe any (supervised) segmentation leading to the extraction of more elaborate and specific descriptors for leaf classification.

3.1 Common procedures

Spatial pyramid local analysis
We define our SP matrix $\boldsymbol{\Lambda}$ with L levels such as $\boldsymbol{\Lambda} \triangleq [\boldsymbol{r}_y, \boldsymbol{r}_x, \boldsymbol{d}_y, \boldsymbol{d}_x, \boldsymbol{\lambda}]$. $\boldsymbol{\Lambda}$ is a matrix of size $(L \times 5)$. For a level $l \in \{0, \ldots, L-1\}$, the image $\boldsymbol{I}$, with size $(n_y \times n_x)$, is divided into potentially overlapping sub-windows $\boldsymbol{R}_{l,v}$ of size $(h_l \times w_l)$. All these windows are sharing the same associated weight λ_l. In our implementation, $h_l \triangleq \lfloor n_y.r_{y,l} \rfloor$ and $w_l \triangleq \lfloor n_x.r_{x,l} \rfloor$ where $r_{y,l}$, $r_{x,l}$ and λ_l are the l^{th} element of vectors $\boldsymbol{r}_y$, $\boldsymbol{r}_x$ and $\boldsymbol{\lambda}$ respectively. Sub-window shifts in $x-y$ axis are defined by integers $\delta_{y,l} \triangleq \lfloor n_y.d_{y,l} \rfloor$ and $\delta_{x,l} \triangleq \lfloor n_x.d_{x,l} \rfloor$ where $d_{y,l}$ and $d_{x,l}$ are elements of $\boldsymbol{d}_y$ and $\boldsymbol{d}_x$ respectively. Overlapping can be performed if $d_{y,l} \leq r_{y,l}$ and/or $d_{x,l} \leq r_{x,l}$. The total number of sub-windows is equal to

$$V = \sum_{l=0}^{L-1} V_l = \sum_{l=0}^{L-1} \lfloor \frac{(1 - r_{y,l})}{d_{y,l}} + 1 \rfloor . \lfloor \frac{(1 - r_{x,l})}{d_{x,l}} + 1 \rfloor . \tag{2}$$

Fig. 2 shows an example of SP with our particular choice $\boldsymbol{\Lambda} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & \frac{1}{8} & 1 \end{bmatrix}$. For this particular $\boldsymbol{\Lambda}$ matrix, we divided twice more the vertical axis than the horizontal one according to the aspect ratio distribution of images in the dataset.

Linear support vector machines for large-scale classification
Let's assume available a training data set $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$, where $\boldsymbol{x}_i \in \mathbb{R}^d$ is a descriptor extracted from image $\boldsymbol{I}_i$ and $y_i \in \{1, \ldots, M\}$, where $M = 126$ is the number of classes and $N = 8422$ is the number of training samples. As in [13, 1], we will use a simple large-scale linear SVM such as LIBLINEAR [6] with the 1-vs-all multi-class strategy. The associated binary unconstrained convex optimization problem to solve is:

$$\min_{\boldsymbol{w}} \left\{ \frac{1}{2} \boldsymbol{w}^T \boldsymbol{w} + C \sum_{i=1}^{N} \max \left(1 - y_i \boldsymbol{w}^T \boldsymbol{x}_i, 0 \right)^2 \right\}, \tag{3}$$

where the parameter C controls the generalization error and is tuned on a specific validation set. LIBLINEAR converges to a solution linearly in $O(dN)$ compared to $O(dN_{sv}^2)$.

Moreover, in order to obtain an estimate of $p(y = l|\boldsymbol{x})$, we performed an SVM regression given the output of the previous classification stage for each binary classifier.

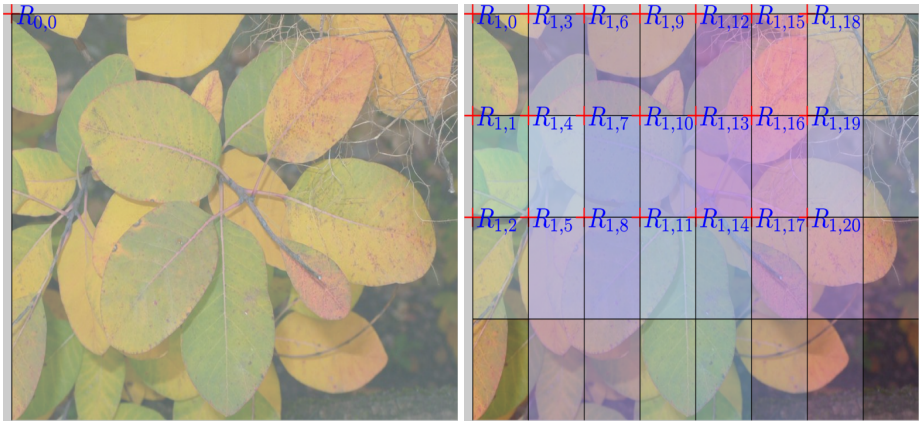

Fig. 2. Example of SPM $\boldsymbol{\Lambda}$ with $L = 2$ and $V = 1 + 21$. Upper-left corner of each window $\boldsymbol{R}_{l,v}$ is indicated with a red cross. Left: $\boldsymbol{R}_{0,0} = \boldsymbol{I}$ for $l = 0$ (first level). Right: $\{\boldsymbol{R}_{1,v}\},\ v = 0, \ldots, 20$ for $l = 1$ (second level).

3.2 Multiscale Color Local Phase Quantization (MSCLPQ) → LSIS_DYNI_run_1

Following [4, 5], we extend the basic decorrelated Local Phase Quantization (LPQ) descriptor for a multi-scale and color channel analysis over a spatial pyramid.

In LPQ, Short Fourier Transforms (SFT) are computed over $M \times M$ windows centered on $\boldsymbol{z}$ at four frequencies $\boldsymbol{u}_1 = [a, 0]^T$, $\boldsymbol{u}_2 = [0, a]^T$, $\boldsymbol{u}_3 = [a, a]^T$ and $\boldsymbol{u}_4 = [a, -a]^T$ with $a = \frac{1}{M}$ such that

$$F(\boldsymbol{u}, \boldsymbol{z}) = \sum_{y \in N_z} f(\boldsymbol{z} - \boldsymbol{y}) e^{-j2\pi \boldsymbol{u}^T \boldsymbol{y}}, \tag{4}$$

where $\boldsymbol{z} \in \boldsymbol{R} \subset \boldsymbol{I}$. For each pixel, we compute the LPQ code as[5]

$$LPQ(\boldsymbol{z}) = \sum_{i=0}^{3} 2^{2i} \mathbb{1}_{\{\mathcal{R}e(F(\boldsymbol{u},\boldsymbol{z}))\geq 0\}} + \sum_{i=0}^{3} 2^{2i+1} \mathbb{1}_{\{\mathcal{I}m(F(\boldsymbol{u},\boldsymbol{z}))\geq 0\}}, \tag{5}$$

where $LPQ(\boldsymbol{z}) \in \{0, \ldots, 255\}$. Local histograms of LPQ codes are retrieved by counting occurrences of each individual LPQ code j such as:

$$x_{LPQ}(j, \boldsymbol{R}) = \sum_{z \in \boldsymbol{R}} \mathbb{1}_{\{LPQ(\boldsymbol{z})=j\}},\ j = 0, \ldots, 255. \tag{6}$$

The local histogram vector is defined by

$$\boldsymbol{x}_{LPQ}(\boldsymbol{R}) \triangleq [x_{LPQ}(0, \boldsymbol{R}), \ldots, x_{LPQ}(255, \boldsymbol{R})]\,, \tag{7}$$

[5] $\mathbb{1}_{\{x\}} = 1$ if event x is true, 0 otherwise.

where $\boldsymbol{x}_{LPQ}(\boldsymbol{R})$ is furthermore ℓ_2 normalized. The full vector $\boldsymbol{x}$ is obtained by concatenating previous normalized histograms for 4 different scales $M \in \{3,5,7,9\}$, $\boldsymbol{\Lambda} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} \end{bmatrix}$ $(V = 1 + 21)$ and the 3 (R, G, B) color channels. The total dimension of this vector is equal to $d = 256.(1 + 21).4.3 = 67584$.

Finally, we normalize each element of $\boldsymbol{x}_i$ such that $x_{i,l} \in [-1, 1]$, $l = 1, \ldots, d$, $i = 1, \ldots, N$ followed by ℓ_2 normalization on $\boldsymbol{x}_i$. The *a posteriori* probabilities associated with the MSCLPQ approach are denoted $p_1(y = l|\boldsymbol{x})$.

3.3 Late fusion of MSCLPQ, MSCILBP and MSILBP+ScSPM → LSIS_DYNI_run_2

Multiscale Color Local Phase Quantization
See sec. 3.2

Multiscale Color Improved Local Binary Pattern (MSCILBP)
Basically, the operator $ILBP$ encodes the relationship between a central block of $(s \times s)$ pixels located in $\boldsymbol{z}_c = [y_x, x_c]^T$ with its 8 neighboring blocks [8] and also adds a ninth bit encoding a term homogeneous to the differential excitation. This operator can be considered as a non-parametric local texture encoder for scale s. In order to capture information at different scales, the range analysis $s \in \mathcal{S}$, is typically set at $\mathcal{S} = [1, 2, 3, 4]$ for this task, where $S = Card(\mathcal{S})$. This micro-codes are defined as follows:

$$ILBP(\boldsymbol{z}_c, s) = \sum_{i=0}^{i=7} 2^i \mathbb{1}_{\{A_i \geq A_c\}} + 2^8 \mathbb{1}_{\left\{\sum_{i=0}^{7} A_i \geq 8Ac\right\}}, \tag{8}$$

where $\forall \boldsymbol{z}_c \in \boldsymbol{R} \subset \boldsymbol{I}$, $ILBP(\boldsymbol{z}_c) \in \mathbb{N}_{2^9}$.

The different areas $\{A_i\}$ and A_c in eq.(8) can be computed efficiently using the image integral technique [12]. Let's define $\boldsymbol{II}$ the image integral of $\boldsymbol{I}$ by:

$$\boldsymbol{II}(y, x) \triangleq \sum_{y'=0}^{y'<y} \sum_{x'=0}^{x'<x} \boldsymbol{I}(y', x'). \tag{9}$$

Any square area $A(y, x, s) \in \boldsymbol{R}$ (see right Fig. 3) with upper-left corner located in (y, x) and side length s is the addition of only 4 values:

$$A(y, x, s) = \boldsymbol{II}(y + s, x + s) + \boldsymbol{II}(y, x) - (\boldsymbol{II}(y, x + s) + \boldsymbol{II}(y + s, x)). \tag{10}$$

As for MSCLPQ, efficient features are obtained by counting occurrences of the j^{th} visual ILBP at scale s in a ROI $\boldsymbol{R} \subseteq \boldsymbol{I}$:

$$x_{ILBP}(\boldsymbol{R}, j, s) = \sum_{\boldsymbol{z}_c \in \boldsymbol{R}} \mathbb{1}_{\{ILBP(\boldsymbol{z}_c, s) = j\}},$$

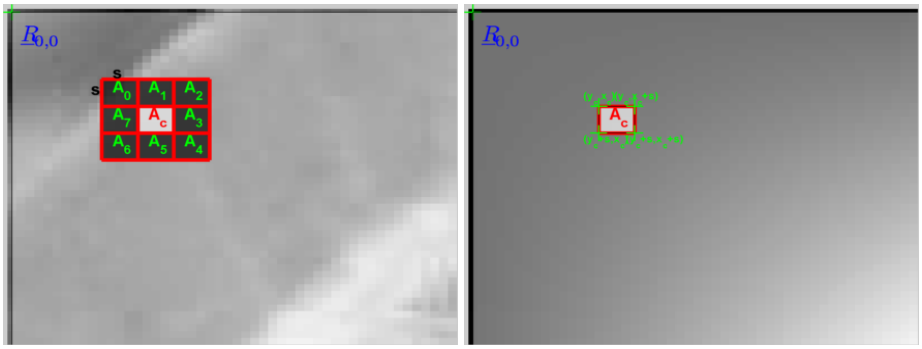

Fig. 3. Left: $\boldsymbol{I}$ and $ILBP(y_x, x_c)$ overlaid. Right: corresponding image integral $\boldsymbol{II}$ and the central block A_c. A_c can be efficiently computed with the 4 corner points.

where $j = 0, \ldots, b-1$ is the j^{th} bin of the histogram and $b = 512$. Full histogram of ILBP, denoted $\boldsymbol{z}_{ILBP}$ is computed by:

$$\boldsymbol{x}_{ILBP}(\boldsymbol{R}, s) \triangleq [x_{ILBP}(\boldsymbol{R}, 0, s), \ldots, x_{ILBP}(\boldsymbol{R}, b-1, s)]\,. \tag{11}$$

Finally, the full vector $\boldsymbol{x}$ is obtained by concatenating previous normalized histograms for 4 different scales $s \in \{1, 2, 3, 4\}$, $\boldsymbol{\Lambda} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} \end{bmatrix}$ $(V = 1 + 21)$ and the 3 (R, G, B) color channels. The total dimension of this vector is equal to $d = 512.(1 + 21).4.3 = 135168$. We also normalize each element of $\boldsymbol{x}_i$ such that $x_{i,l} \in [-1, 1]\,,\ l = 1, \ldots, d,\ i = 1, \ldots, N$ followed by ℓ_2 normalization on $\boldsymbol{x}_i$. The *a posteriori* probabilities associated with MSCILBP approach are denoted as $p_2(y = l|\boldsymbol{x})$.

Sparse coding of dense MSILBP patches

Following the same framework as in [7, 13, 1, 3, 10], we will show here that the traditional Bag of Features (BoF) approach can be advantageously replaced by i) Sparse coding (Sc), ii) max-pooling technique.

Specifically, F ILBP patches $\boldsymbol{z}_{ILBP}(\boldsymbol{O}_k)$ of size $(m \times m)$ centered on ROI's $\{\boldsymbol{O}_k\}$ (possibly overlapping) are extracted (*cf.* eq. 7) for $k = 0, \ldots, F-1$ and $\forall s \in \mathcal{S}$ (see Fig. 4). For a faster computation for each scale s, the integral image $\boldsymbol{II}$ is first computed from $\boldsymbol{I}$.

For a complete dataset containing N images and $\forall s \in \mathcal{S}$, we obtain a collection of $P = TS$ patches $\boldsymbol{Z} \triangleq \{\boldsymbol{z}_i\}$, $i = 1, \ldots, P$, where $T = NF$. We define, the subset of patches $\boldsymbol{z}_i$ at scale s by $\boldsymbol{Z}(s) \subseteq \boldsymbol{Z}$ with T elements. In order to obtain highly discriminative visual features, a common procedure consists of encoding each patch $\boldsymbol{z}_i \in \boldsymbol{Z}(s)$ at scale s through an unsupervised trained dictionary $\boldsymbol{D} \triangleq [\boldsymbol{d}_1, \ldots, \boldsymbol{d}_K] \in \mathbb{R}^{b \times K}$, where K denotes the number of dictionary elements, and its corresponding weight vector $\boldsymbol{c}_i \in \mathbb{R}^K$. In the BoF framework, the vector

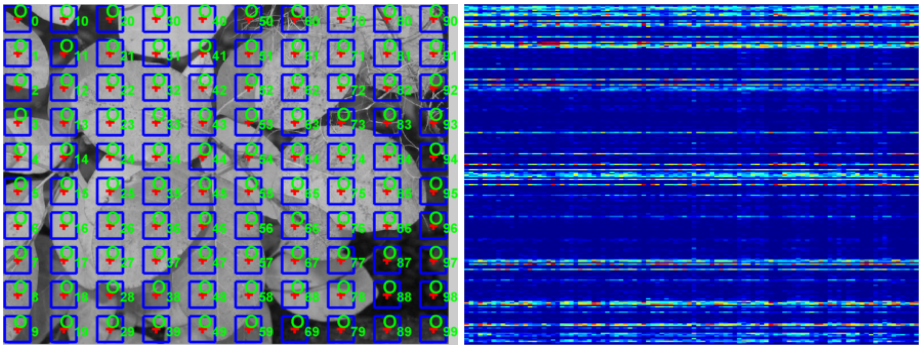

Fig. 4. ExampleLeft: ROI's $\{\boldsymbol{O}_k\}$, $k = 0, \ldots, F-1$ of extracted patches used to compute each ILBP where $F = 10 \cdot 10$. Right: associated normalized histograms $\{\boldsymbol{z}_{ILBP}(\boldsymbol{O}_k)\}$, one per column.

$\boldsymbol{c}_i$ is assumed to have only one non-zero element:

$$\arg\min_{\boldsymbol{D},\boldsymbol{C}} \sum_{i=1}^{T} \|\boldsymbol{z}_i - \boldsymbol{D}\boldsymbol{c}_i\|_2^2 \quad s.t. \quad \|\boldsymbol{c}_i\|_{\ell_0} = 1, \tag{12}$$

where $\boldsymbol{C} \triangleq [\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K]$ and $\| \bullet \|_{\ell_0}$ defines the pseudo zero-norm, where here only one element of $\boldsymbol{c}_i$ is non-zero. In eq. (12), under these constraints, $(\boldsymbol{D}, \boldsymbol{C})$ can be optimized jointly by a Kmeans algorithm for example.

In the Sc approach, in order to i) reduce the quantization error and ii) to have a more accurate representation of the patches, each vector $\boldsymbol{x}_i$ is now expressed as a linear combination of a few vectors of the dictionary $\boldsymbol{D}$ and not only by a single one. Imposing the exact number of non-zero elements in $\boldsymbol{c}_i$ (sparsity level) involves a non-convex optimization [9]. In general, it is preferred to relax this constraint and to use instead an ℓ_1 penalty which also involves sparsity. The problem is then reformulated using the following equation:

$$\arg\min_{\boldsymbol{D},\boldsymbol{C}} \sum_{i=1}^{T} \|\boldsymbol{z}_i - \boldsymbol{D}\boldsymbol{c}_i\|_2^2 + \beta\|\boldsymbol{c}_i\|_{\ell_1} \quad s.t. \quad \|\boldsymbol{c}_i\|_{\ell_1} = 1, \tag{13}$$

where the sparsity in controlled by the parameter β. The last equation is not jointly convex in $(\boldsymbol{D}, \boldsymbol{C})$ and a common procedure consists of optimizing alternatively $\boldsymbol{D}$ given $\boldsymbol{C}$ by a block coordinate descent and then $\boldsymbol{C}$ given $\boldsymbol{D}$ by a LASSO procedure [11]. At the end of the process, for each scale $s \in \mathcal{S}$, a trained dictionary $\widehat{\boldsymbol{D}}(s)$ is obtained.

For an image $\boldsymbol{I}$ and given a trained dictionary $\widehat{\boldsymbol{D}}(s)$ for a type of code at scale s, F sparse vectors $\{\boldsymbol{c}_k(s)\}$ are computed by a LASSO algorithm. The final efficient descriptor $\boldsymbol{x}(s) \triangleq \left[x^0(s), \ldots, x^{K-1}(s)\right] \in \mathbb{R}^K$ is obtained by the

following max-pooling procedure [13, 2]:

$$x^j(s) \triangleq \max_{k|\boldsymbol{O}_k \in \boldsymbol{R}} (|c_k^j(s)|),\ j = 0, \ldots, K-1, \tag{14}$$

where each element of $\boldsymbol{x}(s)$ represents the max-response of the absolute value of sparse codes belonging to the ROI $\boldsymbol{R}$. In order to improve accuracy, a spatial pyramidal matching procedure helps to perform a more robust local analysis. The spatial pyramid $\boldsymbol{\Lambda}$ has $V = \sum_{l=0}^{L-1} V_l$ ROIs $\{\boldsymbol{R}_{l,v}\}$ with $l = 0, \ldots, L-1$, $v = 0, \ldots, V_l - 1$ (see Fig. 5 for an example). The quantity $\boldsymbol{z}_{l,v}^j(s)$ for each ROI $\boldsymbol{R}_{l,v}$ is computed by:

$$x_{l,v}^j(s) \triangleq \max_{k|\boldsymbol{O}_k \in \boldsymbol{R}_{l,v}} (|c_k^j(s)|),\ j = 0, \ldots, K-1. \tag{15}$$

We reinforce our model by an important normalization step that improves con-

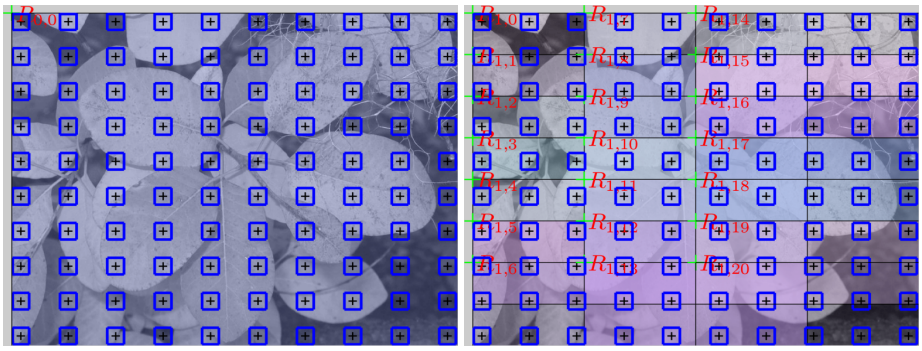

Fig. 5. Example of SPM $\boldsymbol{\Lambda}$ with $L = 2$, $F = 10 \cdot 10$ and $V = 1 + 21$. The F ROIs $\{\boldsymbol{O}_k\}$, $k = 0, \ldots, F-1$ associated with each patch $\boldsymbol{z}_k$ are represented by blue squares. Sparse codes $\boldsymbol{c}_k$ are computed for each ROI $\boldsymbol{O}_k$. Upper-left corner of each max-pooling window $\boldsymbol{R}_{l,v}$ taking $\{100, 10\}$ $\boldsymbol{c}_k$ is indicated with a green cross. Left: $\boldsymbol{R}_{0,0} = \boldsymbol{I}$ for $l = 0$. Right: $\{\boldsymbol{R}_{1,v}\}$, $v = 0, \ldots, 20$ for $l = 1$

siderably accuracy and consists of the ℓ_2 normalization of all vectors $\{\boldsymbol{x}_{l,v}(s)\}$, $v = 0, \ldots, V_l - 1, s \in \mathcal{S}$, *i.e.* belonging to the same pyramidal layer l. This step is also very important and often hidden in the existing literature.

The final descriptor $\boldsymbol{x}(\boldsymbol{\Lambda})$ will be defined by the weighted concatenation of all the $\boldsymbol{x}_{l,v}(s)$ vectors, *i.e.* $\boldsymbol{x}(\boldsymbol{\Lambda}) \triangleq \{\lambda_l \boldsymbol{x}_{l,v}(s)\}$, $l = 0, \ldots, L-1, v = 0, \ldots, V_l - 1$ and $\forall s \in \mathcal{S}$. The total size of the feature vector $\boldsymbol{x}(\boldsymbol{\Lambda})$ is $d = K.V.S$, where typically in our simulations, we fixed $K = 2048$, $V = 22$ and $S = 4$. A final ℓ_2 clamped normalization step is performed on the full vector $\boldsymbol{x}(\boldsymbol{\Lambda})$. In our experiment, we extracted $F = 35 \cdot 35$ patches per scale and per image with $m = 26$. 2000 patches

per class for each scale have been randomly selected to train dictionary ($\beta = 0.2$). The *a posteriori* probabilities associated with MSILBP+ScSPM approach are denoted $p_3(y = l|\boldsymbol{x})$.

Late fusion
To obtain a final decision, we simple performed an average of all $p_f(y = l|\boldsymbol{x})$ *a posteriori* probabilities, *i.e.*

$$p(y = l|\boldsymbol{x}) = \frac{1}{3}\sum_{f=1}^{3} p_f(y = l|\boldsymbol{x}). \tag{16}$$

3.4 Late fusion of MSCLPQ, MSCILBP, MSILBP+ScSPM and SIFT+ScSPM→ LSIS_DYNI_run_3

The three first stages are identical as in LSIS_DYNI_run_2.

Sparse coding of dense SIFT patches
As for MSILBP parches, we extracted $F = 35 \cdot 35$ SIFT patches ($m = 16$) per image and for each of the 4 scales ($\sigma = \{0.5, 0.65, 0.8, 1.0\}$). 2000 patches per class for each scale have been randomly selected to train dictionary ($\beta = 0.2$, $K = 2048$). The *a posteriori* probabilities associated with SIFT+ScSPM approach are denoted $p_4(y = l|\boldsymbol{x})$.

Late fusion
We also performed an average of all $p_f(y = l|\boldsymbol{x})$ *a posteriori* probabilities

$$p(y = l|\boldsymbol{x}) = \frac{1}{4}\sum_{f=1}^{4} p_f(y = l|\boldsymbol{x}). \tag{17}$$

4 Results

Fig. 6 presents the summarized results for the "scans" category. Without any segmentation and/or specific pre-processing, we obtained a score $S = 0.41$ with LSIS_DYNI_run_3, *i.e.* the 6^{th} best score for all submitted runs (29 in total), relatively close to the top-4 ($S = 0.43$). Higher scores can probably be obtained with the use of color MBILBP+ScSPM and color SIFT+ScSPM features.

In Fig. 7 we summarize the results for the "scan-like" category. We obtained a score $S = 0.42$ with LSIS_DYNI_run_3, *i.e.* the 7^{th} best score for all submitted runs (29 in total). In this case, with an unsupervised detector to "home-in" leafs more precisely, we could also improve results. Ideally, as for all runs above $S = 0.42$, a prior segmentation is known to help considerably results for "scans" and "scan-like" categories. Finally, Fig. 8 provides the summarized results for the "photographs" category. We obtained a score $S = 0.32$ with LSIS_DYNI_run_3, *i.e.* the 1^{th} best score for automatic method and for all submitted runs (29 in total). Our 3 runs obtained the best top-3 of all submitted runs.

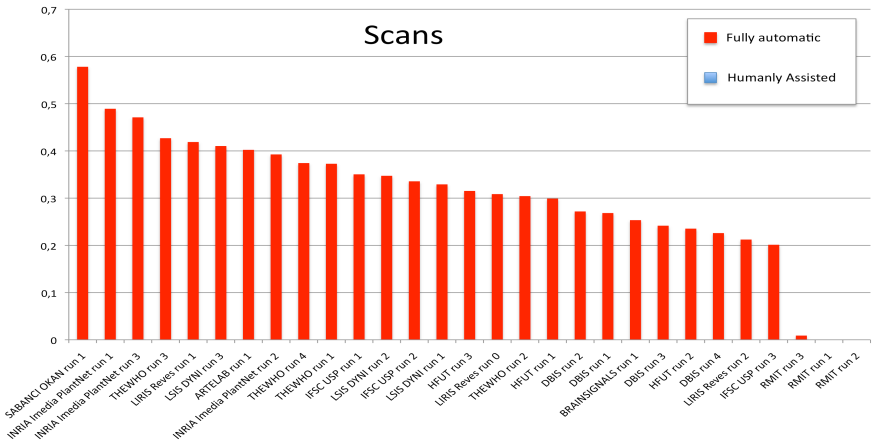

Fig. 6. Results for the "scans" category.

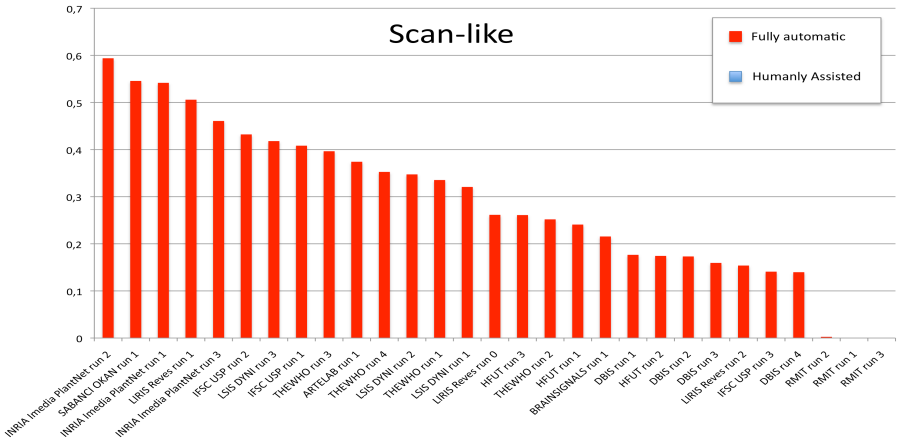

Fig. 7. Results for the "scan-like" category.

5 Conclusions

For our first participation to ImageCLEF plants identification 2012 challenge, we demonstrated that for "photographs" category, our framework offers best performances for automatic method. This category is considered the most challenging due to "real" *in-situ* conditions and shows that computer vision approaches for image catagorization/fine-grained visual categorization are well adapted for this challenge. Several improvements can be obtained, for example with some better encoding schemes (Fisher vectors) and/or pooling technics.

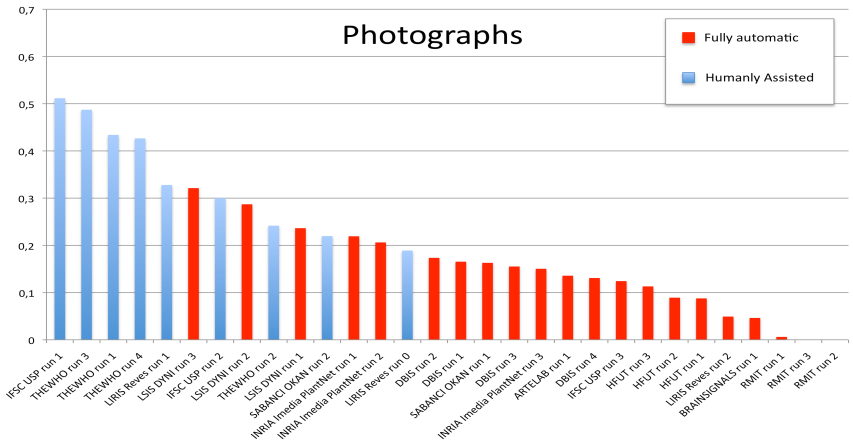

Fig. 8. Results for the "photographs" category.

References

1. Boureau, Y., Bach, F., LeCun, Y., Ponce, J.: Learning mid-level features for recognition. In: CVPR' 10 (2010)
2. Boureau, Y., Ponce, J., LeCun, Y.: A theoretical analysis of feature pooling in vision algorithms. In: ICML' 10 (2010)
3. Chatfield, K., Lempitsky, V., Vedaldi, A., Zisserman, A.: The devil is in the details: an evaluation of recent feature encoding methods. In: BMVC (2011)
4. Heikkilä, J., Ojansivu, V.: Methods for local phase quantization in blur-insensitive image analysis. In: LNLA' 09 (2009)
5. Heikkilä, J., Ojansivu, V., Rahtu, E.: Improved blur insensitivity for decorrelated local phase quantization. In: ICPR' 10 (2010)
6. Hsieh, C., Chang, K., Lin, C., Keerthi, S.: A dual coordinate descent method for large-scale linear svm (2008)
7. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR' 06 (2006)
8. Liao, S., Zhu, X., Lei, Z., Zhang, L., Li, S.Z.: Learning multi-scale block local binary patterns for face recognition. In: ICB (2007)
9. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online dictionary learning for sparse coding. In: ICML '09 (2009)
10. Paris, S., Halkias, X., Glotin, H.: Sparse coding for histograms of local binary patterns applied for image categorization: Toward a bag-of-scenes analysis. In: ICPR' 12 (2012)
11. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society (Series B) 58 (1996)
12. Viola, P., Jones, M.: Robust real-time face detection. International Journal of Computer Vision 57 (2004)
13. Yang, J., Yu, K., Gong, Y., Huang, T.S.: Linear spatial pyramid matching using sparse coding for image classification. In: CVPR' 09 (2009)