IBM Research Australia at LifeCLEF2014: Plant Identification Task

Qiang Chen, Mani Abedini, Rahil Garnavi, Xi Liang

IBM Research Australia

Abstract. In this paper, we present the system and learning strategies that were applied by the IBM Research team to the plant identification task of LifeCLEF 2014. Plant identification is one of the most popular fine-grained categorization tasks. To ensure high classification accuracy, we have utilised strong visual features together with fusion of robust machine learning techniques. Our proposed system involves automatic delineation of the region of interest (e.g. plant's leaf, flower, etc.) in the given image, followed by extracting multiple complementary low level features. The features have been then encoded into the sophisticated Fisher Vector representation which enables accurate classification with linear classifiers. We have also applied the recent development of deep learning. More importantly our system combines multiple source of information, i.e. integrates organ annotation with image data, and adopts fusion of classifiers which has led to great results. The extensive experiments demonstrate the effectiveness of the proposed system, where three (out of four) of our submissions outperforms all submissions by other teams, therefore the team achieves the first place in LifeCLEF 2014 Plant task.

Keywords: Fine-grained object recognition, plant recognition, deep learning, feature coding

1 Introduction

The fine-grained classification, i.e. classification among categories which are both visually and semantically very similar, is a very difficult task. It is even challenging for humans without careful training, and is critical for establishing a more detailed understanding of the visual world [11][18][10]. Identifying plant species is essential for successful agricultural development and conservation of biodiversity. However, it is a very difficult task even for professionals (e.g. farmers, wood exploiters, botanists). To evaluate recent advances of information retrieval and computer vision on this challenging task, the CLEF Cross Language Evaluation forum has been organizing yearly competition on plant identification since 2011. Following the success of the three previous ImageCLEF Plant identification tasks, LifeCLEF 2014 plant task consists of 500 plant species dedicated to botanical data. The task has focused on tree, herbs and ferns species identification based on different types of images. Main novelties compared to the last years are the following:

– Multi-image query: The motivation of the task is to fit better with a real scenario where one user tries to identify a plant by observing its different organs, such as it has been demonstrated in [MAED2012]. Indeed, botanists usually observe simultaneously several organs, e.g. the leaves and the fruits or the flowers in order to identify species which could be mystified if only one organ were observed.
– Observation-based evaluation: Unlike previous years, the species identification task is not image-centered but observation-centered. The aim of the task is to produce a list of relevant species for each observation of a plant of the test dataset, i.e. one or a set of several pictures related to a same event, where a same person photographs several detailed views on various organs the same day with the same device with the same lightening conditions observing the same plant.
– Large number of species: The number of species is about 500, which is an important step towards covering the entire flora of a given region.

This paper presents the system and learning strategies that were applied by the IBM Research team to the plant identification task of LifeCLEF 2014. The rest of the paper is organised as follows: Section 2 describes the main components of the system, involving segmentation, feature extraction and learning, and methodologies applied at each module. Section 3 discusses the details of experiments and obtained results on validation and test sets. Section 4 concludes the paper.

2 Approach

In this section we first briefly describe our submission to the LifeCLEF 2014 competition. Then various components of the system, including segmentation (delineation of the region of interest), feature coding and deep convolutional nerual network are explained.

2.1 IBM Research Australia Runs

We have submitted four different runs to LifeCLEF [22] Plant identification task [23].

1. **Run1**- Deep convolutional nerual network: In this run, we utilize the deep convolutional nerual network model explained in Sec.2.4 with five layers of convolutional network, three layers of fully connection and cost layer of logistic regression. The CNN is trained on the plant training data provided by LifeClef 2014, which is relative small scale data for this deep model.
2. **Run2**- Advanced feature encoding: In this run, we apply the advanced feature encoding methods explained in Sec2.3 using Fisher Kernel encoding [20][5][17]. We first extract dense feature, e.g. SIFT [2] and Color Moment from raw images. Each feature is modeled with Mixture of Gaussian (GMMs) and forms the Fisher Vector representation. Then we learn a linear

SVM [14] for each feature. The final submission is the average score from the two features.

3. **Run3**- Fusion of Run1 and Run2: We have applied an empirical fusion method, where each of the components are fed into the fusion module and a weight has been tuned for each component.
4. **Run4**- Segmentation and Fusion of Run2 and Run3: We have applied the feature encoding method on images with ROI extracted. We then fuse this result with Run3. We also made a few improvements, e.g. SVM averaging.

The details of the methods used above are explained in the following section.

2.2 Segmentation

In most images, using a region of interest (ROI) which encloses the main object is sufficient to determine their class label. In fact, segmentation of the main objects and extracting the ROI often results in removing the irrelevant background which could introduce noise to a supervised classier. In categories of flower, fruit, leaf, leafScan and stem, we apply different segmentation methods to remove the background. For the two categories of wholeTree and branch, however, we have observed that the whole image contains useful information. Therefore, we choose not to extract any ROI from these views, instead the original image is used in the next modules of the system, i.e. feature extraction and learning.

For the flower and fruit categories, we apply a similar segmentation method to extract the ROI that contains flower or a fruit, as follows: assuming that a flower or fruit is usually "more red" compared to leaves, we locate the regions of which their red channel is larger than the green channel. Each color image is converted to a gray-value image, and the localized "red" regions are then employed as initial masks in segmenting the flower or fruit in the gray-value image using a active contour method [21]. In the end, we compute the minimum bounding box of the flower or fruit mask as the ROI. Figure 1 shows ROI extraction results on some samples on flower pictures and Figure 1 shows those on fruit pictures.

Fig. 1. Original flower image (left) and automatically extracted ROI (right)

For the leafScan category, every image typically includes a leaf with a light background. However, there are variations among the background colors. We therefore normalize the background with a consistent white color which can potentially improve the accuracy of the classification. We convert the color image

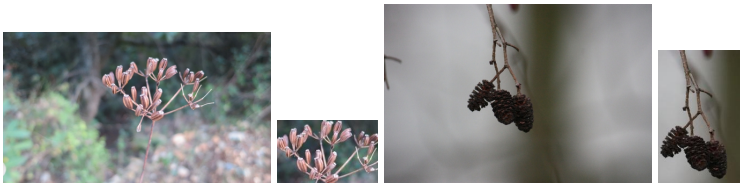

Fig. 2. Original fruit image (left) and automatically extracted ROI (right)

into a gray-value image and then apply Otsu method [1] to compute a threshold. The pixels in the gray-value image that are smaller than the threshold are labeled as background, and all background pixels in the corresponding color image are assigned with a while color. Some examples are shown in Figure 3.

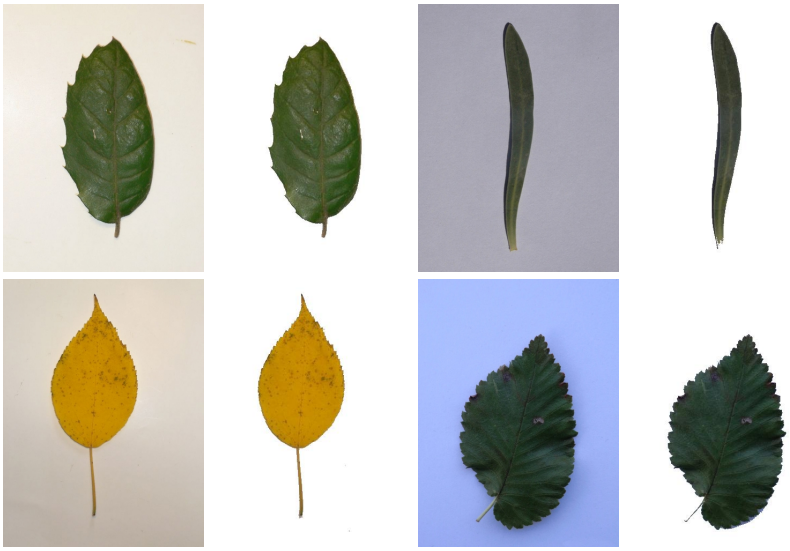

Fig. 3. Original LeafScan image (left) and the ROI (right)

In images of the leaf category, we observed that a leaf is usually located in the center area of the picture and there are typically some margin between the leaf boundary and the picture border. We therefore extract the object that is located in the center of the picture. In pre-processing, we convert the color picture to a gray-value image and apply a Gaussian filter on the image with $\sigma = 3$. We then extract an rectangular ROI in the image as an initial ROI that contain the leaf. The ROI is defined by its left top and bottom right coordination in the image. Let (r, l) is the number of rows and columns in the image, the left top coordination is $(\frac{r}{6}, \frac{l}{6})$ and the right bottom coordination is $(\frac{5r}{6}, \frac{5l}{6})$. We then apply active contour on the gray-value image using the rectangular ROI as the

initial mask. The boundary box of the segmented leaf is the final ROI and some examples are shown in Figure 4.

Fig. 4. Original leaf image (left) and automatically extracted ROI (right)

In the stem category, the stem in usually located in the center of the image. For segmenting the stem, we convert the color image to a gray-value image and creating a central mask on the image by cropping %25 from left and %25 from right. Active contour is then applied on the gray-value image using the mask. The bounding box of the resulting area is the ROI for the stem image. This method can effectively remove most of the background for vertical stems without branches. Some examples are shown in Figure 5.

Fig. 5. Original Stem image (left) and automatically extracted ROI (right)

2.3 Advanced Feature Coding

Feature coding is the generalization of the popular BoW model [8][3]. The recent evaluation [4] shows that the FisherKernel [20][5] feature coding achieves best results in most of the cases. We introduce this coding method in this section.

Suppose we have a probability density function $u_\lambda(x)$ which models a generative process in feature space. Let $X = \{x_1, \cdots, x_N\}$ be the set of N local features

extracted from an image. Then the image can be described by the gradient vector of log likelihood with respect to the model parameters λ:

$$G_\lambda^X = \frac{1}{N}\nabla_\lambda \log u_\lambda(X). \tag{1}$$

A natural kernel on these gradients is $K(X,Y) = G_\lambda^{X'} F_\lambda^{-1} G_\lambda^Y$ where F_λ is the Fisher information matrix of u_λ: $F_\lambda = E_{x\sim u_\lambda}[\nabla_\lambda \log u_\lambda(x) \nabla_\lambda \log u_\lambda(x)']$. As F_λ is symmetric and positive definite, it can be decomposed as $F_\lambda = L'_\lambda L_\lambda$, and the kernel $K(X,Y)$ can be expressed as a dot-product between normalized vectors $\mathcal{G}_\lambda^X = L_\lambda G_\lambda^X$ called Fisher vectors. We stress that learning a kernel classifier using the Fisher kernel is equivalent to learning a linear classier on the Fisher vectors $\mathcal{G}_\lambda^X$. As been recognized widely, linear classifiers offer significant advantages in terms of efficiency both for training and testing.

Fisher Vector encoding utilizes a Gaussian mixture model (GMM), $u_\lambda(x) = \sum_{k=1}^K \pi_k u_k(x)$ trained on local features of a large image set using Maximum Likelihood (ML) estimation. The parameters of the trained GMM are denoted as $\lambda = \{\pi_k, \mu_k, \Sigma_k, k = 1, \cdots, K\}$, where $\{\pi, \mu, \Sigma\}$ are the prior probability, mean vector and diagonal covariance matrix of the Gaussian mixture respectively. This GMM is used for description of low level features. Then for a set of low level features $X = \{x_1, \cdots, x_N\}$ extracted from an image y, the soft assignments of the descriptor x_i to the kth Gaussian component γ_{ik} is computed by:

$$\gamma_{ik} = \frac{\pi_k u_k(x_i)}{\sum_{k=1}^K \pi_k u_k(x_i)} \tag{2}$$

And the Fisher vector (FV) for X is denoted as $\phi(X) = \{\mathcal{G}_{\mu_1}^X, \mathcal{G}_{\sigma_1}^X, \cdots, \mathcal{G}_{\mu_K}^X, \mathcal{G}_{\sigma_K}^X\}$ where $\mathcal{G}_{\mu_k}$ and $\mathcal{G}_{\sigma_k}$ is defined as:

$$\mathcal{G}_{\mu_k}^X = \sum_{i=1}^N \frac{1}{N\sqrt{\pi_k}} \gamma_{ik} \frac{x_i - \mu_k}{\sigma_k}, \tag{3}$$

$$\mathcal{G}_{\sigma_k}^X = \sum_{i=1}^N \frac{1}{N\sqrt{2\pi_k}} \gamma_{ik} [\frac{(x_i - \mu_k)^2}{\sigma_k^2} - 1], \tag{4}$$

Where σ_k is the square root of the diagonal values of Σ_k. The FV has several good properties: (a) Fisher Vector encoding is not limited to computing visual word occurrences. It also encodes the distribution information of the feature points, which will perform more stable when encoding a single feature point. (b) it can naturally separate the image specific information from the noisy local features. (c) we can use a linear model for this representation.

Power Normalization and L2 Normalization: It is easy to observe that as the number of Gaussians increases, Fisher vectors become sparser, and the distribution of features in a given dimension becomes more peaky around zero. This issue is addressed by a combination of power normalization and l2 normalization for each Fisher vector descriptor. Suppose z is one dimension of the fisher vector ϕ, the power normalization is defined as $f(z) = sign(z)|z|^\alpha$ where

$0 \leq \alpha \leq 1$ is a parameter of the normalization and we choose $\alpha = 0.5$ in all the experiments. Subsequently, the Fisher vectors are $l2$ normalized.

2.4 Deep Convolutional Neural Network

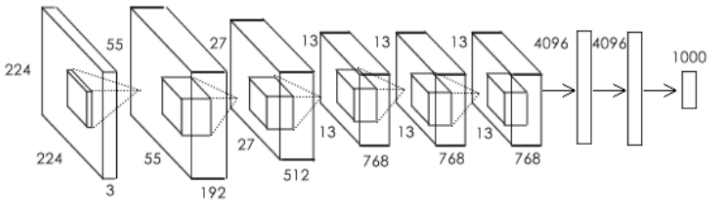

Fig. 6. A deep convolutional neural network with similar configuration in [6]

.

We also applied a deep convolutional nerual network (CNN) for the plant identification task. We use the training data from the plant task only, which is relative small scale data for this deep model. We also tried a pretrained CNN using ImageNet [7][13] dataset, as this usage of extra data is not allowed in the LifeCLEF challenge, we didn't submit the result but only evaluated on our internal validation set.

Our CNN has around 60 million parameters. it consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final softmax layer. To make training faster, we have used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected layers we employed a recently-developed regularization method called dropout that proved to be very effective. We also use the data augmentation as in [6]. In order to incorporate the information of the "Organ" annotation, we propose two methods: (1) we train the CNN with two objective function which targets the label accuracy and view accuracy at the same time. (2) we train the CNN with one objective function but we set the class label as the enumerate of the species and view annotation. These two implementation turns out providing similar performance in our validation set.

2.5 Fusing of Multiple Systems

Until now we have presented the sub models of the system, we then fuse the results at the late stage. Each sub model k provides a confidence score $s^k_{i,j}$ for each observationID/image$i$ and each category j. We optimize to get the final confident score as the weighted sum: $S_{i,j} = \sum_k w^k * s^k_{i,j}$. The optimization is performed on the validation set. Each time, we select on sub model with best

accuracy and tune the weight to get the best fused accuracy. The same set of the weight parameters have been used to get the confidence score on the test set.

3 Experiments

In this section we discusses the details of experiments and obtained results on validation and test sets.

3.1 Experimental setting

Dataset The PlantCLEF dataset focuses on 500 herb, tree and fern species centered on France (some plants observations are from neighboring countries). It contains more than 60000 pictures belonging to one of the seven types of view reported into the meta-data, in a xml file (one per image) with explicit tags. A part of the dataset has been provided as training data and the remaining part will be used later as test data. Test observation will be chosen by randomly sampling 1/3 of the observations of each species.

The training data finally results in 47815 images, including 1987 of "Branch", 6356 photographs of "Entire", 13164 of "Flower", 3753 "Fruit", 7754 of "Leaf", 3466 "Stem" and 11335 scans and scan-like pictures of "leaf". The test data contains 8163 plant-observation-queries. These queries are based on 13146 images; 731 of "Branch", 2983 photographs of "Entire", 4559 of "Flower", 1184 "Fruit", 2058 of "Leaf", 935 "Stem" and 696 scans and scan-like pictures of "leaf".

Validation Set In order to verify the effective of each components in our system, we split the "training data" provided by PlantCLEF into two parts: a train set and a validation set. The validation set is roughly 1/5 of the total training data. We split the training data according to the observation id which is critical since the final evaluation is based on observation id. In the following section, we will report the results on both the validation set and the testing set.

Implementation details As previously mentioned, in the feature coding approach, we extract two types of local features: dense SIFT feature and Color moment feature. The dimension of each local feature has been reduced to 64 using PCA [16]. Then for each type of feature, we generate a GMM model which has 512 components.

For deep convolutional neural network, we follow the pipeline of Alex [6]. The filter size and filter number is the same as Alex's. We restrict the node number of the fully connection network to 2048 as this number is far more enough to model the plant images.

We use open source libraries, e.g. SIFT from VL-feat [9] and SVM solver from LibSVM [15]

3.2 Results on Validation Set

For proper assessment of each components of our system, we perform diagnosis evaluation on the validation set. The evaluation results are shown in Table 1 based on two metrics as proposed by PlantCLEF organiser, i.e. the Accuracy w.r.t. image (*Acc_image*) and Accuracy w.r.t. observation id (*Acc_observ*).

Table 1. Results for each components on validation set.

ValidationSet	FeatureCoding	FeatureCoding+ROI	DNN	**Combine**
Acc_image	0.445	0.458	0.32	**0.483**
Acc_observ	0.624	0.63	0.44	**0.656**

Based on result shown in Table 1, we can observe the following:

1. The feature coding pipeline (by combing two type of features, i.e. SIFT and color moment) achieves stable results. As reported in the literature, the Fisher Kernel encoding performs best among many of the encoding methods in visual recognition.
2. Segmentation (ROI extraction) improves the pipeline of feature coding and increase the classification accuracy.
3. The CNN method results in lower classification accuracy, compared to feature coding. We believe the effect is the due to limited number of training data. Deep learning has been demonstrated great success when using large scale dataset. The PlantCLEF dataset is a middle scale dataset and the deep model can be easily overfitted.

3.3 Results on Test Set

In this subsection, we present our final results submitted to LifeCLEF2014 plant identification task. Results of each run is shown in Table 2, based on two metrics: the Accuracy w.r.t. image (*Acc_image*) and Accuracy w.r.t. observation id (*Acc_observ*).

Table 2. Results for each runs on test set.

TestSet	RUN1	RUN2	RUN3	**RUN4**
Acc_image	0.263	0.438	0.446	**0.456**
Acc_observ	0.271	0.454	0.459	**0.471**

From Table 2 we observe:

– RUN1 is the result of using deep convolutional nerual network. It obtains reasonable result but inferior to other runs. The main problem is that the deep model is very easy to overfitting on the training set with so many parameters.

– RUN2 is the raw result with feature coding pipeline which gives quite good result. It demonstrates that the feature coding pipeline is very mature and easily adaptable to many tasks in visual recognition.
– RUN3 is the result from combing RUN2 and RUN1. It shows that the CNN framework is complementary to the traditional feature coding framework.
– RUN4 is the result from combing RUN3 and feature coding pipeline applied on segmented images. It shows the correct delineation of ROI improves the classification results.

Further evaluation results:

1. **Overall performance**: Our submissions achieve the top three results when comparing with other teams as shown in Figure 7 in terms of both metrics, i.e. the Accuracy w.r.t. image (*Acc_image*) and Accuracy w.r.t. observation id (*Acc_observ*). Our classification results obtains 15-20% improvement over other teams.

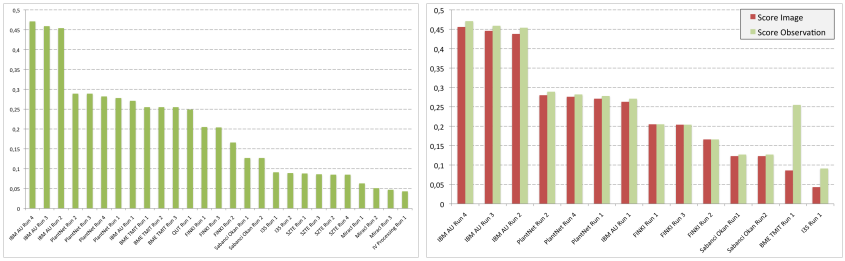

Fig. 7. Official Test results for the submitted Plant Identification runs, based on observation id (left) and image id (right). IBM runs achieved top three performances.

2. **Results for each organ (view)**: Figure 8 shows the result for each organ (view). Our classification results again leads other teams in most of cases. It is worth noting that our system's performance on LeafScan and Flower is better than 50%. Considering such fine-grained categroization task, we believe this result has practical value for real system.
3. **Observation-based evaluation**: We observe a notable improvement from image-based evaluation to observation-based evaluation on the validation set, which shows the ability of our system in using multiple source of image data and perform a more accurate classification. Unfortunately, the trend is not that obvious on the test set. This is caused by dissimilarity between the training and testing set in terms of average number of images per observation id, i.e. there is around 4.5 images per observation id in the training set, while we only have around 1.6 images per observation id in the test set.

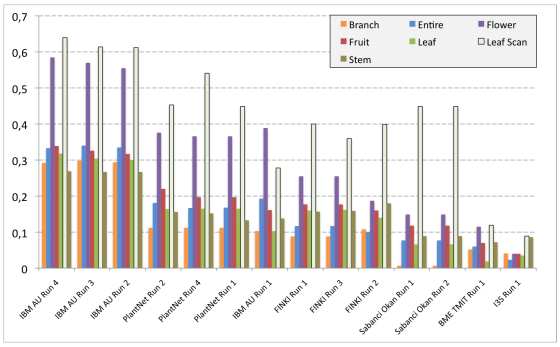

Fig. 8. The final results evaluated based on each image for each organs.

4 Conclusion

In this paper, we described the system and learning methodology applied by the IBM Australia Research team to the plant identification task of LifeCLEF 2014. We utilized the advanced feature coding method with automatic ROI extraction. We also applied the recent development of deep learning and achieved great result on the validation set. The most important contribution of this work is in effective fusion of various learning schemes and proper use of multiple source of information (annotation data and image data). The extensive experiments demonstrated the effectiveness of the proposed system and the final submitted run achieved the first place in LifeCLEF 2014 Plant task.

References

1. Otsu, Nobuyuki. A threshold selection method from gray-level histograms. Automatica 11.285-296 (1975): 23-27.
2. DG Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
3. L Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition*, 2005.
4. K Chatfield, V Lempitsky, and A Vedaldi. The devil is in the details: an evaluation of recent feature encoding methods. In *British Machine Vision Conference*, 2011.
5. Florent Perronnin, Jorge Sanchez, and Thomas Mensink. Improving the Fisher Kernel for Large-Scale Image Classification. In *European Conference on Computer Vision*, 2010.
6. A Krizhevsky, I Sutskever, and G Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.
7. Jia Deng, Wei Dong, R Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database . In *Computer Vision and Pattern Recognition*, 2009.
8. J Sivic and A Zisserman. Video Google: a text retrieval approach to object matching in videos. In *International Conference on Computer Vision*, 2003.

9. A Vedaldi and B Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.
10. P Welinder, S Branson, T Mita, C Wah, F Schroff, S Belongie, and P. Perona. Caltech-UCSD birds 200. Technical report, California Institute of Technology, 2010.
11. M-E Nilsback and A Zisserman. Automated Flower Classification over a Large Number of Classes. In *ICVGIP*, pages 722–729, 2008.
12. L Fei-Fei, R Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1):59–70, 2007.
13. Jia Deng, Alexander C Berg, and Li Fei-Fei. Hierarchical semantic indexing for large scale image retrieval,. In *Computer Vision and Pattern Recognition*, 2011.
14. CJC Burges. A tutorial on support vector machines for pattern recognition. *Data Min Knowl Discovery*, 1998.
15. Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.
16. I T Jolliffe. *Principal Component Analysis*. Springer Verlag, October 2002.
17. F Perronnin, Z Akata, and Z Harchaoui. Towards Good Practice in Large-Scale Learning for Image Classification. In *Computer Vision and Pattern Recognition*, 2012.
18. P. Belhumeur, D. Chen, S. Feiner, D. Jacobs, W. Kress, H. Ling, I. Lopez, R. Ramamoorthi, S. Sheorey, S. White, and L. Zhang. Searching the World's Herbaria: A System for Visual Identification of Plant Species. In *European Conference on Computer Vision*, 2008.
19. Y. Su and F. Jurie. Improving Image Classication using Semantic Attributes. *International Journal of Computer Vision*, 100, 1 (2012) 59-77, 2012.
20. Perronnin, F and Dance, C. Fisher Kernels on Visual Vocabularies for Image Categorization. In *Computer Vision and Pattern Recognition*, 2007.
21. Chan, T. F., Vese, L. A. (2001). Active contours without edges. *IEEE Transactions on Image Processing*, 10(2), 266277.
22. Joly, Alexis and Müller, Henning and Goëau, Hervé and Glotin, Hervé and Spampinato, Concetto and Rauber, Andreas and Bonnet, Pierre and Vellinga, Willem-Pier and Fisher, Bob LifeCLEF 2014: multimedia life species identification challenges. Proceedings of CLEF 2014.
23. Goëau, Hervé and Joly, Alexis and Bonnet, Pierre and Molino, Jean-François and Barthélémy, Daniel and Boujemaa, Nozha LifeCLEF Plant Identification Task 2014 Proceedings of CLEF 2014.