Leaf Segmentation under Loosely Controlled Conditions

Simone Buoncompagni[1]
simone.buoncompagni2@unibo.it
Dario Maio[1]
dario.maio@unibo.it
Vincent Lepetit[2]
lepetit@icg.tugraz.at

[1] Department of Computer Science and Engineering
University of Bologna
Bologna, Italy

[2] Institute for Computer Graphics and Vision
Graz University of Technology
Graz, Austria

Abstract

We propose a robust and accurate method for segmenting specular objects acquired under loosely controlled conditions. We focus here on leaves because leaf segmentation plays a crucial role for plant identification, and accurately capturing the local boundary structures is critical for the success of the recognition. Popular techniques are based on Expectation-Maximization and estimate the color distributions of the background and foreground pixels of the input image. As we show, such approaches suffer in presence of shadows and reflections thus leading to inaccurate detected shapes. Classification-based methods are more robust because they can exploit prior information, however they do not adapt to the specific capturing conditions for the input image. Methods with regularization terms are prone to smooth the segments boundaries, which is undesirable. In this paper, we show we can get the best of the EM-based and classification-based methods by first segmenting the pixels around the leaf boundary, and use them to initialize the color distributions of an EM optimization. We show that this simple approach results in a robust and accurate method.

1 Introduction

Extracting accurately the shape of a leaf is a crucial step in image-based plant identification systems. The partial or total absence of textures on leaf surface and the high color variability of leaves belonging to same species make shape as the main recognition element [1, 5, 17, 18, 19]. For such reason, leaf segmentation plays a decisive role in the leaf recognition process.

Even though many general segmentation methods [2, 3, 10, 11, 12, 22, 24] have been proposed in the last decades, leaf segmentation presents specific challenges. In particular, a pixel-level precision is required in order to highlight fine scale boundary structures and discriminate similar global shapes. Moreover, even if the input image can typically be taken in controlled conditions, where the leaf is the only visible object over a white background, the user taking the picture is not necessarily an expert and the conditions are often not ideal:

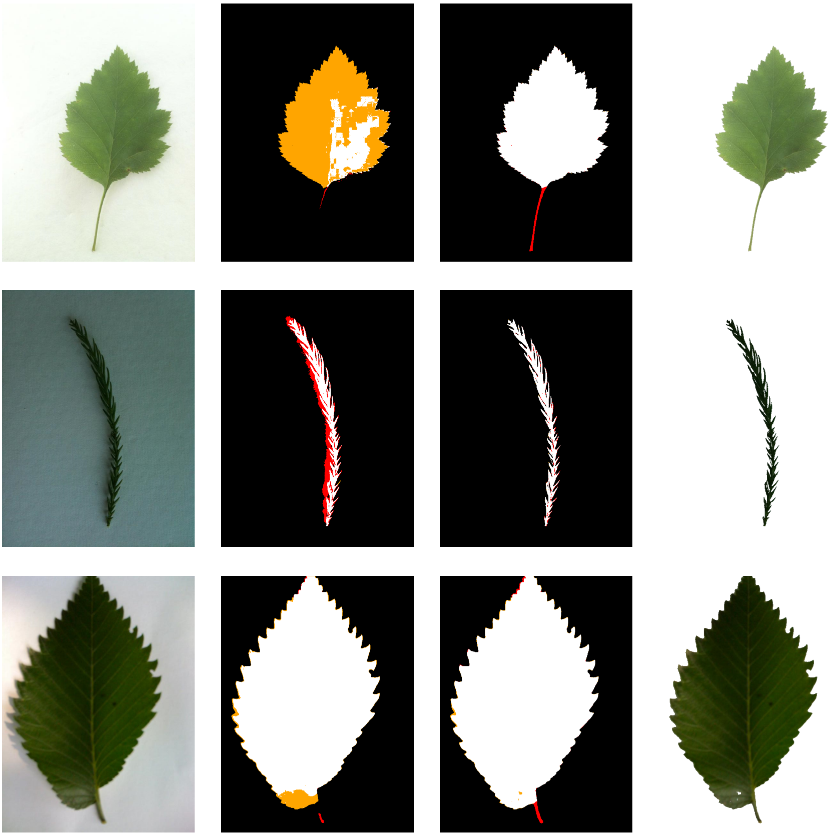


Figure 1: Leaves segmentation under loosely controlled conditions (best viewed in color). **First column**: Leaf images with the presence of shadows and irregular light. **Second column**: Segmentation result obtained by the Leafsnap method [19] with included post-processing procedure for stems and false positives suppression. **Third column**: Results obtained by our classification-based approach without any post-processing **Last column**: Input image masked with our segmentation. Our classification-based method is more robust to the presence of shadows and irregular light thus offering contours that better fit to the real shape. Red and orange colors are used to mark false positives and false negatives, respectively (ground truth does not include stems). We provide many other visual results in the supplemental material.

the leaf exhibits specular reflections, casts shadows, the background is never exactly white and is usually non-uniform, and the image can be blurry.

Recent leaf recognition applications in loosely conditions [19, 26] rely on the Expectation-Maximization algorithm to separate the color distributions of the foreground and the background pixels. Despite their efficiency, they do not assure robustness to shadows and specular reflections thus leading to incorrect boundaries. In this paper we introduce a new solution by training a pixel-wise classifier that learns filter responses associated to background and foreground regions in images of leaves. Similar classifiers have been recently used in different fields like medical applications [25] showing great performance for linear and curvilinear structures segmentation.

As shown in Fig. 1 and as proved in Section 4, we observed benefits when training our classifier by considering as positive (leaf) and non-positive (background) training samples only those pixels located in the neighborhood of the leaf boundary.

After applying the classifier to a given input image, we threshold the score map to get segments which belong to foreground or background with a high level of probability. These segments allow us to infer precious color and spatial information about the leaf and provide a support for a suitable initialization of EM algorithm, which yields to a very fine pixel-wise segmentation robust to shadows and specularities.

In the remainder of the paper, we first review related work in Section 2. Then, in Section 3 we detail our method. In Section 4 we present the dataset we used for experimental evaluation and we show benefits of our approach over the state-of-the-art.

2 Related work

Leaf segmentation represents a core activity for plant identification and research about such topics is constantly rising from the past ten years [1]. Even though great effort has been devoted to object segmentation on images in the Computer Vision history [3], leaves require a precise segmentation and/or boundary detection to effectively describe shape and its local structures. Since a detailed overview of general-purpose segmentation is beyond our scope, we focus here mainly on state-of-the-art of leaf segmentation. Moreover, we provide a review of the emerging results about filters response learning for some specific tasks like detection of curvilinear structures.

Supervised and Unsupervised Environments

Various leaf segmentation approaches related to different environmental conditions have been proposed. Image binarization with a fixed threshold, or “Otsu’s method” [21], demonstrated good accuracy for leaf images acquired in supervised setups characterized by uniform light conditions and white background, as those included in the FLAVIA dataset [28], the Swedish leaf dataset [27], *Lab* image category of the Leafsnap dataset or the *scan* image category from the ImageCLEF plant identification challenges [18].

Very different solutions have been introduced for leaf segmentation on images acquired in unsupervised conditions like those included in recent ImageCLEF challenges [13, 14, 18], with *natural background* or *photo* image categories, where no assumptions are made about the background behind the leaf during image acquisition. A number of automatic [7, 20, 29, 30] and interactive [4, 8, 9, 30] approaches have been presented to solve leaf segmentation in unconstrained setups. In [7] two different semi-automatic and automatic segmentation approaches based on Mean-Shift and K-Means clustering in RGB color space are introduced whereas in [29] a combination of shape, color and texture features are used for plant identification. In [8] polygonal shape models of leaves are employed as prior offering very good support in unsupervised conditions but limiting its applicability to modeled species. A similar approach based on the use of semantic information and guided active contour segmentation has been later presented in [9]. As very recently illustrated in [15], due to the considerable challenge of leaf segmentation and recognition against natural background, user supervision and interaction are recommended during the process to produce reliable input images and initialize the segmentation.

Semi-Supervised Environments

Regarding to leaf segmentation under semi-supervised conditions addressed on such paper, several automatic approaches have been already presented and tested. Most promising ones [5, 19] are based on the use of EM [6] in color space to estimate foreground and background pixel clusters. As shown in [26], standard EM and its extensions outperform other techniques as graph-based image segmentation [12], Mean Shift [10], GrabCut [22], segmentation by weighted aggregation (SWA) [24], multiscale normalized cut [11]. Particular improvements have been demonstrated by EM when dealing with images taken with mobile devices under various pose and illumination conditions, see the *Field* or *User* image categories of [19].

However, as reported in [26], EM-based methods do not assure robustness to shadows, specular reflections and requires the adoption of *ad hoc* solutions also for certain particular leaves such as pine leaves, thus proving the weakness of EM initialization.

In this paper, we show that our classification-based initialization for background and foreground color distribution represents a better solution for the problem at hand. Cascade classifiers [23] exhibit good performance in localization thus allowing a better discrimination of points with similar appearance, as those placed across object contours. Advantages offered by the learning of filter responses have been recently proved in different fields like biomedical images, aerial images and general-purpose contour detection [25]. We aim to apply a similar idea for leaf segmentation, by combining prior knowledge with learning and the adaptability of EM-based methods.

3 Method

Let $I(\mathbf{x})$ be an image of a leaf, and $\mathbf{x} \in \mathbb{R}^2$ an image pixel location. Leaf segmentation can be carried out by computing the probability distribution of all pixels $\mathbf{x}$ and representing it as the mixture of two Gaussians:

$$p(I(\mathbf{x})|\Theta) = \sum_{i=1}^{2} \omega_i \, p_i(I(\mathbf{x}) \mid \Theta_i) \qquad \Theta_i = \{\mu_i, \Sigma_i\} \quad , \tag{1}$$

where $I(\mathbf{x})$ is the color of image I at location $\mathbf{x}$, and $\Theta_1 = \{\mu_1, \Sigma_1\}$ and $\Theta_2 = \{\mu_2, \Sigma_2\}$ are the parameters of the foreground—the leaf—and background color distributions, respectively. ω_1 and ω_2 weight these two Gaussian distributions.

One way to infer such distributions is to apply K-Means or Expectation-Maximization as in [5, 19], thus computing the parameters and weights of the two Gaussians for the given image and using them for pixel-wise segmentation. [19] considers only the saturation and value color components for EM clustering and a shared covariance matrix is used. However, in practice, some drawbacks appear in this formulation due to challenging leaves like pine needles, false positives detection related to shadows and false negatives detection related to specularities. To tackle such problems, some manually-defined assumptions are made about cluster regions and pixel weights (see Fig. 2 in [19]). Furthermore, post-processing operations are carried-out to remove false positive detections due to shadows and irregular backgrounds, at the risk of hurting the final leaf shape.

To assure more robustness to shadows and specularities, our solution is to pre-train a pixel-wise classifier by learning a function $y(\cdot)$ such that:

$$y(f(\mathbf{x},I)) = \begin{cases} 1 & \text{if } \mathbf{x} \text{ is on the leaf surface,} \\ -1 & \text{otherwise,} \end{cases} \tag{2}$$

where $f(\mathbf{x},I)$ is a feature vector computed from a neighborhood surrounding $\mathbf{x}$ in image I.

By performing a simple thresholding of the score map returned by the classifier we detect segments that belong with high probability to foreground and background. These segments are then exploited to properly initialize a standard EM algorithm thus leading to a final and accurate leaf segmentation. Furthermore, we will show that our learning assures independence from leaf species, since the same classifier is used for all species and no *ad hoc* solutions are required when challenging species have to be treated.

In the remainder of this section we firstly illustrate our pixel-wise classifier by showing our training that focuses on leaf boundary. Then, we describe how we produce and employ segments to initialize the EM algorithm thus leading to the final segmentation.

3.1 Pre-trained pixel-wise classifier

To train our pixel-wise classifier we employ a similar approach to [25]. Given a set of training samples $\{(f_i, y_i)\}_{i=1,\ldots,n}$ where $f_i = f(\mathbf{x}_i, I_i) \in \mathbb{R}^J$ is the feature vector corresponding to a point $\mathbf{x}_i$ in image I_i and $y_i \in \{1,-1\}$ is the label associated to $\mathbf{x}_i$, we use GradientBoost and regression trees [16] to approximate $y(f(\mathbf{x},I)$ by a function of the form:

$$\varphi(f(\mathbf{x},I)) = \sum_{k=1}^{K} \alpha_k h_k(f(\mathbf{x},I)) \ , \tag{3}$$

where $h_k : \mathbb{R}^J \to \mathbb{R}$ are weak learners and $\alpha_k \in \mathbb{R}$ are weights.

As shown in Fig. 2, we focus our attention on the leaf boundaries by selecting only samples in their neighborhoods. We extract from each training image the leaf contour from the ground truth segmentation and simply thicken this contour with standard morphology dilation. Function φ is built iteratively by minimizing an exponential loss function $\mathcal{L}$ of the form:

$$\mathcal{L} = \sum_{i=1}^{n} L(y_i, \varphi(f(\mathbf{x},I))) \ , \tag{4}$$

where $L = e^{-y_i \varphi(f(\mathbf{x},I))}$. We also experimented with the log loss function with similar results.

We use a set of convolutional filters learned from the training images as described in [25]. The RGB images are converted into the LUV color space and we learn a different filter bank for each channel.

3.2 Score map thresholding and segmentation

Our segmentation pipeline is summarized in Fig. 3. We apply the classifier described above to each pixel location of a given unknown test image I_{test}. This provides a score map that we then threshold using two different thresholds to detect pixels that belong to foreground and background with a high level of probability. With these pixels at hand we initialize an EM algorithm to estimate foreground and background cluster parameters Θ_1 and Θ_2 by working in the saturation-value color space.

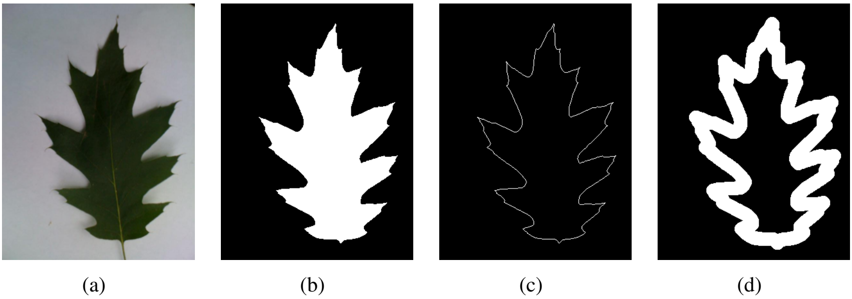


Figure 2: Example of images used for training: (a) a leaf image, (b) its manually defined ground truth segmentation, (c) the leaf contour extracted from the segmentation, (d) thicker contour obtained by simple dilation. We train our classifier by selecting positive (leaf) and negative (non-leaf) feature samples computed on image (a) that lie on the thicker contour only.

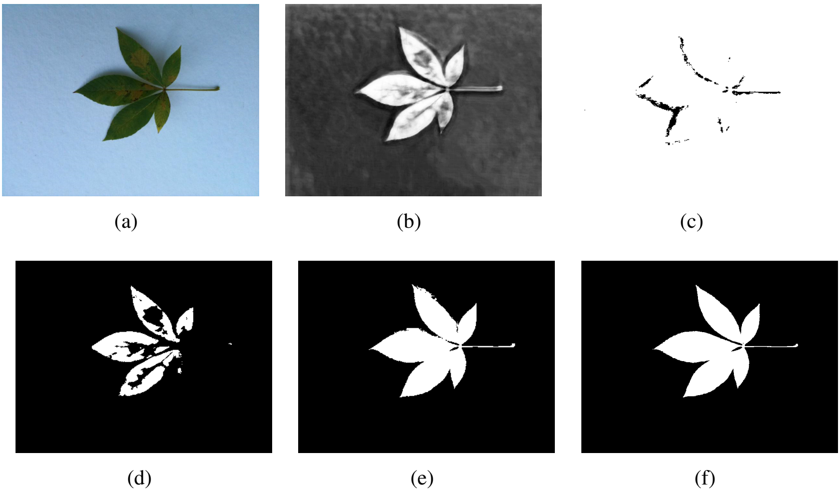


Figure 3: Segmentation pipeline: (a) Input image I_{test}, (b) score map obtained by applying our pre-trained classifier at each pixel location, (c) pixels belonging to background with high probability (black), (d) pixels belonging to foreground with high probability (white), (e) coarse leaf segmentation obtained using the prior $\Theta_{1_{\text{start}}}, \Theta_{2_{\text{start}}}$ built from images (c) and (d), (f) final leaf segmentation after EM optimization from this initialization. Since our training is focused on leaf boundary, high probability background and foreground pixels are more likely to be found near the leaf boundary thus guiding and improving the following final segmentation.

This allows us to compute good initial estimates for Θ_1 and Θ_2, the mean and covariances of the colors over the leaf and over the background. This is by contrast with [19], which has to initialize the EM segmentation with the same values for all the images. The other difference with [19] is that we can consider as unlabeled data only the pixels that are in the neighborhood of the detected leaf boundary. This allows to keep focusing on segmenting correctly the pixels around the leaf boundary, and in practice it is enough to get a good segmentation of the other pixels, which are easier to classify.

4 Results

In this section, we first describe the dataset and the evaluation protocol we used for our experiments. We then compare our method with techniques that demonstrated state-of-the-art performance on loosely controlled conditions, Leafsnap[19] and GrabCut [22]. In particular, we show the benefit given by our classification-based initialization of EM as described in Section 3.2. Moreover we evaluate the importance of training the classifier from samples close to the leaf boundaries. We finally provide qualitative results of our segmentation approach.

4.1 Leaves dataset and performance metrics

For evaluation we use the *Field* image dataset publicly available online [19]. It is made of 185 different species for a total of 7719 images acquired against solid background and variable light conditions thus simulating typical images that a user could provide for plant recognition.

To train our pixel-wise classifier we randomly select one image for each species and we manually produce segmentation and thicker contour to discriminate between positive and negative training samples placed in the neighborhood of boundary as described in Section 3.1.

Since segmentation ground truth is not available and its manual production for thousands of images would require an inestimable amount of time, we considered a subset of the original *Field* dataset. Our testing set is made of 300 images: 150 images for which the EM approach of [19] performs already well thus producing faithful segmentation in accordance with the leaf shape plus 150 more challenging images for which EM partially or totally fails.

185 training images are randomly selected excluding those images already used to test the classifier. A total of 485 leaves (185 for training and 300 for testing) was therefore manually segmented to produce the ground truth; stems and unrelated components are not part of the ground truth in accordance to the policy employed in [19]. [1]

To compute performance indicators we rely on the publicly available and very popular code of Berkeley Segmentation and Boundary Detection Benchmark [2]. In particular, as in [26], we evaluate the leaf segmentation results by analyzing boundary agreement with ground truth in terms of recall, precision and F-measure since contour is the main recognition cue in typical plant recognition systems.

4.2 Segmentation performance

Accuracy measures are reported in Table 1 and Table 2. Specifically, in Table 1 we provide recall, precision, and the F-measure (ODS) which is the harmonic mean of precision and

[1] Our manual ground truth segmentation is publicly available at http://smartcity.csr.unibo.it/leaf-segmentation/

recall to evaluate the trade off between these two measures:

$$F\text{-}measure = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad . \tag{5}$$

Such metrics are computed for the global testing set whereas in Table 2 the same results are reported when only the 150 more challenging images are considered to highlight the benefits of our approach.

We compare the two different strategies to train the classifier and initialize the EM segmentation: using samples from the entire image, strategy that we denote *Ours-entire*, and using samples only close to the leaf boundaries, which we denote *Ours-boundary*. Moreover, we report results when only the pre-trained classifier is employed for segmentation (*Classification*).

	Image segmentation quality			**Leaf boundary quality**		
	Recall	*Precision*	*F-measure*	*Recall*	*Precision*	*F-measure*
Classification	0.701	0.480	0.570	0.702	0.778	0.738
Leafsnap	0.618	0.858	0.718	0.618	0.929	0.742
Leafsnap*	0.644	0.764	0.699	0.644	0.931	0.762
GrabCut	0.624	0.848	0.719	0.624	0.964	0.757
Ours-entire	0.690	0.800	0.741	0.690	0.940	0.796
Ours-boundary	0.692	0.822	**0.752**	0.693	0.944	**0.799**

Table 1: Recall, precision, and F-measure for the entire testing set (300 images). Our method provides the best trade-off between recall and precision.

	Image segmentation quality			**Leaf boundary quality**		
	Recall	*Precision*	*F-measure*	*Recall*	*Precision*	*F-measure*
Classification	0.700	0.532	0.604	0.700	0.788	0.742
Leafsnap	0.560	0.777	0.651	0.560	0.884	0.686
Leafsnap*	0.614	0.703	0.656	0.614	0.903	0.731
GrabCut	0.598	0.830	0.695	0.598	0.959	0.737
Ours-entire	0.682	0.772	0.724	0.682	0.923	0.785
Ours-boundary	0.686	0.792	**0.735**	0.686	0.927	**0.788**

Table 2: Recall, precision and F-measure on 150 challenging images from the testing set. Our method provides the best trade-off between recall and precision.

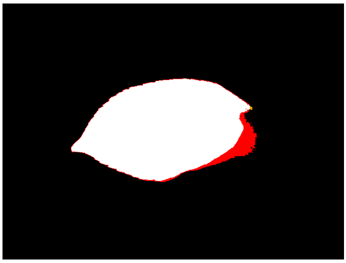
The benefits of our method already appear clearly in Table 1, with a significant raise of the F-measure with respect to the other methods. Even without performing any post-processing to remove false positives, shadow and stems—we remind here that in our ground truth stems are removed, the F-measure for the entire image is better with respect to Leafsnap thus proving the robustness of our method to false positives. Moreover, using only samples placed on leaf boundary to pre-train our classifier (*Ours-boundary*) we outperform all the other methods.

Looking at Table 2 where only challenging images are considered, the improvements due to our method are confirmed to a greater extent. The results prove that a post-processing


(a) **Leaf image**


(b) **Ground truth**


(c) **Leafsnap* [19]**


(d) **Leafsnap [19]**


(e) **GrabCut [22]**


(f) **Ours-boundary**

Figure 4: Leaves segmentation under loosely controlled conditions with different methods. Note that our approach strongly reduces negative effects of irregular light and shadow regions thus offering a more well defined leaf shape with respect to other methods that do not adapt to specific light conditions. Red and orange colors are used to mark false positives and false negatives, respectively (ground truth does not include stems). Best viewed in color.

based on morphological operations as erosions and dilations hurts quality of boundary especially in terms of recall, thus motivating the adoption of methods already robust to shadows and irregular light.

The behavior of different methods can be qualitatively appreciated looking at Fig. 4 where results returned by Leafsnap, Leafsnap without post-processing (marked with *), GrabCut and our method *Ours-boundary* are reported. As the reader can see comparing ground truth details with real segmentations, it is confirmed that post-processing hurts quality of boundaries and should be avoided. On the other hand, GrabCut tends to return round contours. With our method some errors still remain, due to those background pixels that look strongly similar to leaf (and vice-versa). However, our method represents a good compromise since we do not use post-processing but at the same time we assure a good robustness to false positives. Furthermore, our contours tend to fit better with the ground truth. Our non-optimized MATLAB code on a 4-core virtual machine with 16GB of RAM requires about 50 seconds to produce segmentation for one image. The majority of time is required to do classification and produce score map whereas only few milliseconds are required for EM segmentation. Even though at this stage we are not able to guarantee competitive processing time, we are confident that with proper code optimization and the use of more performing hardware like physical machines we can reach much shorter run-times.

5 Conclusion

We have introduced a robust classification-based method for leaf segmentation under loosely controlled conditions. We showed how to adapt to the conditions of the input images, and that focusing on the contours of the object to segment yields better results.

References

[1] G. Agarwal, P. Belhumeur, S. Feiner, D. Jacobs, W. J. Kress, R. Ramamoorthi, N. A. Bourg, N. Dixit, H. Ling, D. Mahajan, et al. First Steps Toward an Electronic Field Guide for Plants. *Taxon*, pages 597–610, 2006.

[2] P. Arbelaez, C. Fowlkes, D. Martin, and J. Malik. http://www.eecs.berkeley.edu/research/projects/ Cs/vision/grouping/resources.html, 2011.

[3] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour Detection and Hierarchical Image Segmentation. *PAMI*, 33(5):898–916, 2011.

[4] A. Arora, A. Gupta, N. Bagmar, S. Mishra, and A. Bhattacharya. A Plant Identification System Using Shape and Morphological Features on Segmented Leaflets: Team IITK, CLEF 2012. In *CLEF*, 2012.

[5] P. N. Belhumeur, D. Chen, S. Feiner, D. W. Jacobs, W. J. Kress, H. Ling, I. Lopez, R. Ramamoorthi, S. Sheorey, S. White, et al. Searching the World's Herbaria: A System for Visual Identification of Plant Species. In *ECCV*, pages 116–129. Springer, 2008.

[6] J. A. Bilmes et al. A Gentle Tutorial of the EM Algorithm and Its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. *International Computer Science Institute*, 4(510), 1998.

[7] D. Casanova, J. B. Florindo, W. N. Gonçalves, and O. M. Bruno. IFSC/USP at Image-CLEF 2012: Plant Identification Task. In *CLEF*, 2012.

[8] G. Cerutti, V. Antoine, L. Tougne, J. Mille, L. Valet, D. Coquin, and A. Vacavant. Reves Participation-Tree Species Classification Using Random Forests and Botanical Features. In *CLEF*, page 1, 2012.

[9] G. Cerutti, L. Tougne, J. Mille, A. Vacavant, and D. Coquin. Understanding Leaves in Natural Images–A Model-Based Approach for Tree Species Identification. *CVIU*, 117(10):1482–1501, 2013.

[10] D. Comaniciu and P. Meer. Mean Shift: A Robust Approach Toward Feature Space Analysis. *PAMI*, 24(5):603–619, 2002.

[11] T. Cour, F. Benezit, and J. Shi. Spectral Segmentation with Multiscale Graph Decomposition. In *CVPR*, pages 1124–1131, 2005.

[12] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient Graph-Based Image Segmentation. *IJCV*, 59(2):167–181, 2004.

[13] H. Goëau, P. Bonnet, A. Joly, V. Bakic, D. Barthélémy, N. Boujemaa, and J.-F. Molino. The ImageCLEF 2013 Plant Identification Task. In *CLEF*, 2013.

[14] H. Goëau, P. Bonnet, A. Joly, I. Yahiaoui, D. Barthélémy, N. Boujemaa, and J.-F. Molino. The ImageCLEF 2012 Plant Identification Task. In *CLEF*, 2012.

[15] M. Grand-brochier, A. Vacavant, G. Cerutti, C. Kurtz, J. Weber, and L. Tougne. Tree Leaves Extraction in Natural Images: Comparative Study of Preprocessing Tools and Segmentation Methods. *Image Processing*, 24(5):1549–1560, 2015.

[16] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, and R. Tibshirani. *The Elements of Statistical Learning*. Springer, 2009.

[17] A. Joly, H. Goëau, P. Bonnet, V. Bakić, J. Barbe, S. Selmi, I. Yahiaoui, J. Carré, E. Mouysset, J.-F. Molino, et al. Interactive Plant Identification Based on Social Image Data. *Ecological Informatics*, 23:22–34, 2014.

[18] A. Joly, H. Goëau, H. Glotin, C. Spampinato, P. Bonnet, W.-P. Vellinga, R. Planque, A. Rauber, R. Fisher, and H. Müller. Lifeclef 2014: Multimedia Life Species Identification Challenges. In *Information Access Evaluation. Multilinguality, Multimodality, and Interaction*, pages 229–249. Springer, 2014.

[19] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. C. Lopez, and J. V. Soares. Leafsnap: A Computer Vision System for Automatic Plant Species Identification. In *ECCV*, pages 502–516. Springer, 2012.

[20] J. C. Neto, G. E. Meyer, and D. D. Jones. Individual Leaf Extractions from Young Canopy Images Using Gustafson–Kessel Clustering and a Genetic Algorithm. *Computers and Electronics in Agriculture*, 51(1):66–85, 2006.

[21] N. Otsu. A Threshold Selection Method from Gray-Level Histograms. *SMC*, 9(1):62–66, 1979.

[22] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive Foreground Extraction Using Iterated Graph Cuts. *TOG*, 23(3):309–314, 2004.

[23] M. Seyedhosseini, M. Sajjadi, and T. Tasdizen. Image Segmentation with Cascaded Hierarchical Models and Logistic Disjunctive Normal Networks. In *ICCV*, pages 2168–2175, 2013.

[24] E. Sharon, M. Galun, D. Sharon, R. Basri, and A. Brandt. Hierarchy and Adaptivity in Segmenting Visual Scenes. *Nature*, 442(7104):810–813, 2006.

[25] A. Sironi, V. Lepetit, and P. Fua. Multiscale Centerline Detection by Learning a Scale-Space Distance Transform. In *CVPR*, pages 2697–2704. 2014.

[26] J. V. Soares and D. W. Jacobs. Efficient Segmentation of Leaves in Semi-Controlled Conditions. *MVA*, 24(8):1623–1643, 2013.

[27] O. Soderkvist. Computer Vision Classification of Leaves from Swedish Trees. *Master's thesis*, 2001.

[28] S. G. Wu, F. S. Bao, E. Y. Xu, Y.-X. Wang, Y.-F. Chang, and Q.-L. Xiang. A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network. *ISSPIT*, pages 11–16, 2007.

[29] B. Yanikoglu, E. Aptoula, and C. Tirkaz. Automatic Plant Identification from Photographs. *MVA*, 25(6):1369–1383, 2014.

[30] B. A. Yanikoglu, E. Aptoula, and C. Tirkaz. Sabanci-Okan System at Imageclef 2012: Combining Features and Classifiers for Plant Identification. In *CLEF*, 2012.