

A Plant Identification System using Shape and Morphological Features on Segmented Leaflets: Team IITK, CLEF 2012

Akhil Arora¹, Ankit Gupta², Nitesh Bagmar¹, Shashwat Mishra¹, and Arnab Bhattacharya¹

¹ Department of Computer Science and Engineering,
Indian Institute of Technology, Kanpur, India.

{aarora, nitesh, shashm, arnabb}@cse.iitk.ac.in

² Department of Computer Science and Engineering,
University of Florida, Gainesville, USA.
angupta@ufl.edu

Abstract. Automatic plant identification tasks have witnessed increased interest from the machine learning community in recent years. This paper describes our (team IITK's) participation in the Plant Identification Task, CLEF 2012, organized by the Combined Lab Evaluation Forum (CLEF) where the challenge was to identify plant species based on leaf images. We first categorize the different types of images and then use a variety of novel preprocessing methods such as shadow and background correction, petiole removal and automatic leaflet segmentation for identifying the leaf blobs. We next use complex network framework along with novel tooth detection method and morphological operations to compute several useful features. Finally, we use a random forest for classification. Based on the proposed approach, we achieved 2nd rank on the overall score in the competition.

Keywords: plant identification, leaflet segmentation, shadow correction, petiole removal, complex network features, tooth features.

1 Introduction

Automatic plant identification tasks have gained recent popularity due to its use in quick characterization of plant species without requiring the expertise of botanists. Leaf-based features are preferred over flowers, fruits, etc. due to the seasonal nature of the later and also the abundance of leaves (except may be for the winter season). The Combined Lab Evaluation Forum (CLEF) hosts an annual competition on classifying plant species based on images of leaves. While there are some other important publicly available leaf image datasets such as the Flavia Dataset [12], the SmithSonian Leaf Dataset [3], and Swedish Leaf Dataset [11], the ImageCLEF dataset [7] provided by CLEF is more challenging due to the difficulty of automatically segmenting the leaves in the images. Apart



Fig. 1. Sample images for the Plant Identification Task, CLEF 2012.

from containing scanned images and images taken in a controlled setup (pseudo-scan), the dataset also contains natural photographs of plant species. Thus, the performance achieved on the CLEF dataset is a more realistic benchmark of the current state-of-the-art in this domain. Fig. 1 shows certain example images from the dataset.

This paper describes our (team IITK) approach for the ImageCLEF Plant Identification Task, CLEF 2012. Our focus for this endeavor was on two main points: (i) providing good recognition accuracy for natural images and (ii) automating the process for the controlled setup images. We have been able to achieve both our targets satisfactorily as corroborated by the fact that one of our submitted runs achieved the 2nd position overall in the competition.

Our contributions in this paper are as follows:

1. We propose novel pre-processing strategies for shadow removal and background noise correction.
2. We propose a fully automatic leaflet extraction approach for compound leaves.
3. We propose the use of tooth features, that provide a second level of discrimination for leaves with similar shape.
4. We also incorporate the use of an effective feedback based image segmentation interface for natural photographs.

2 Proposed Approach

In this section, we present our proposed approach in detail. We begin with a description of the dataset followed by the preprocessing techniques. Next, we discuss the image features used and conclude with the classifier.

2.1 The ImageCLEF Pl@ntLeaves II Dataset

The ImageCLEF Pl@ntLeaves II dataset consists of a total of 11572 images from 126 tree species in the French Mediterranean area. The dataset is subdivided into three different types based on the acquisition methodology used: scans (57%), scan-like photos, i.e., pseudo-scans (24%) and natural photographs (19%). The entire dataset is divided into a training and a test set as specified in Table 1.

Type	Scan	Pseudo-scan	Natural	Total
Training	4870	1819	1733	8422
Test	1760	907	483	3150
Total	6630	2726	2216	11572

Table 1. Statistics of images in the dataset.

Associated with each image are metadata fields that include acquisition type, GPS coordinates of the observation, name of the author of the picture. The training images also contain the *taxon name* of the leaf species, and the task is to predict this field for the test images.

To classify an image, we first need to segment the leaf from the image. The process of segmentation, however, is not straightforward at all owing to the presence of several bottlenecks such as shadow, occlusion and complex leaves (Fig. 1 highlights some of these). While some of the roadblocks such as shadow removal, petiole removal, etc. are common to most of the images, a quick glance at the dataset suggests that no common segmentation scheme can be applied to all the images. Images having a single leaf have different issues than compound leaf images. It is, therefore, useful to group the images with similar issues into a category and address each category separately. Based on this observation, the dataset was each divided into three categories as follows

- **Category 1:** Scan + Pseudo-scan, Single Leaf
- **Category 2:** Scan + Pseudo-scan, Compound Leaf
- **Category 3:** Natural Photographs

All natural photographs (type 3) were put in a single category. The remaining images were put in two separate groups depending on whether they contained single or compound leaves.

Fig. 2 shows the overview of our system. Based on the category of the image, we follow different paths. We next discuss the image preprocessing techniques for each category.

2.2 Image Preprocessing Techniques

The image preprocessing involved steps such as basic segmentation, petiole removal, shadow removal, background noise removal, etc. which collectively aid the extraction of the leaf part from the image. The procedure is fully automatic for category 1 and category 2 images while it is semi-automatic (interactive) for category 3.

Category 1 Images:

This category is composed of scan and pseudo-scan images of single leaf species. Fig. 3(a) shows one such image. We first perform OTSU thresholding³

³ OTSU performs binarization by selecting an optimum threshold to separate the foreground and background regions of the image such that their combined (intra-region) variance is minimal.

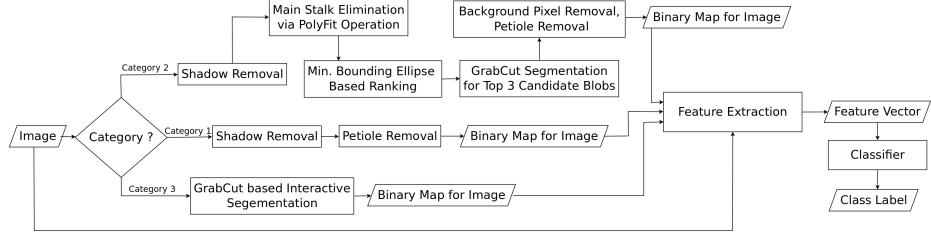


Fig. 2. Plant identification system overview.

[9] on the grayscale image. For many cases, the output is not as expected due to severe background color variation, confusion of shadow regions as leaf, etc. The aim of the pre-processing step is to handle these. We use the following three-stage process to obtain the correct bitmaps for the images in this category:

1. **Binariization:** The image I is converted to grayscale and OTSU thresholding is performed to obtain a “distorted” bitmap, I_o . We use the term distorted as the output is easily affected by shadow and noise in the background. Fig. 3(b) shows the output for the example image Fig. 3(a).
2. **Shadow and Noise Removal:** Since both scan and pseudo-scan images were taken against a plain background (low saturation), we observed that the falsely detected problematic background regions almost always had a lower saturation value than the true leaf region. We leverage this information to identify the problematic regions in the OTSU thresholded I_o by transforming it into the HSV color space and then deselecting the low saturation regions. More formally, we performed OTSU thresholding on the saturation space of I_o to obtain I_s . We subtract I_s from I_o to get a mask, I_n , that contains the noise regions. Since some leaf regions with low saturation value may also be sometimes present in I_n , we erode I_n to deselect such regions and invert the resultant to obtain I_{nf} . A logical AND operation of I_{nf} and I_o gives the shadow- and noise-free bitmap I_{ad} . Fig. 3(c) shows the result for Fig. 3(b).
3. **Petiole Removal:** Several images contained very long petiole sections which were part of the output of the previous step and, therefore, were (falsely) detected as being part of the leaf. Since petioles can adversely affect the shape characteristics of the leaf if their length is comparable to that of the leaf, it is needed to deselect them. This is achieved by searching for abrupt surges in the thickness as we scan each row from top to bottom. Rows whose thickness fell below a certain threshold (as a ratio of the maximum thickness of the leaf) were identified as petiole sections and were removed from the bitmap I_{ad} to obtain the final bitmap I_f which is used for feature vector computation. Fig. 3(d) shows the bitmap after petiole removal from Fig. 3(c).

Category 2 Images:

This category is composed of scan and pseudo-scan images of compound leaf species. Fig. 4(a) shows one such example. Such species contains a main stalk

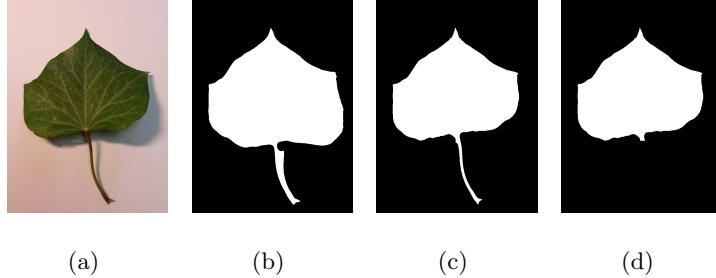


Fig. 3. Category 1 preprocessing: Fig. 3(a) shows the original image, Fig. 3(b) shows the bitmap prior to shadow removal, Fig. 3(c) shows the bitmap after shadow removal, Fig. 3(d) shows the bitmap after petiole removal.

and several *leaflets* that branch out from the main stalk. Using shape-descriptors on the entire leaf does not capture the characteristics of the different compound leaf species. Thus, it is necessary to perform all analysis at the leaflet level. The challenge then is to segment a single leaflet from the compound leaf image. Our system undertakes the following steps to achieve the same:

1. **Binarization and Shadow and Noise Removal:** Since these two steps do not involve the intricacies of leaf structure, we follow the exact same procedure as in Category 1 images. Fig. 4(a) shows such an example.
2. **Main Stalk Elimination:** Since the ultimate aim is to extract a single leaflet, first the main stalk needs to be identified and removed from the image. A simple erosion operation does not work as the thickness of the main stalk can vary quite largely. We fit a curve of order 4 to approximate the main stalk (using the ‘polyfit’ operator from Octave [6]). The curve so obtained is thickened over neighboring pixels to ensure the formation of multiple connected components (blobs) in the binary image. Fig. 4(b) shows the approximated main stalk and Fig. 4(c) shows the image after its elimination.
3. **Ellipse based Blob Ranking:** The previous step outputs multiple blobs that need to be ranked according to their relevance, i.e., how closely they resemble a leaflet. We use the simple assumption that the shape of a leaflet can be approximated by an ellipse, and thus proceed to figure out how much does a blob resemble an ellipse. For each blob, a “relevance score” is computed that measures the area of the blob as a ratio of the area of the minimum bounding ellipse (MBE) around it. Higher this score, higher is the blob likely to be an ellipse. We retain the top three blobs according to this scoring function and this is output to the next stage for further refinement.
4. **GrabCut Segmentation:** The minimum bounding ellipses of the top three contenders are now used as inputs to the GrabCut algorithm [10] which in turn returns the images containing the leaflets. We observe that the images thus obtained are not always perfect and may contain background noise and

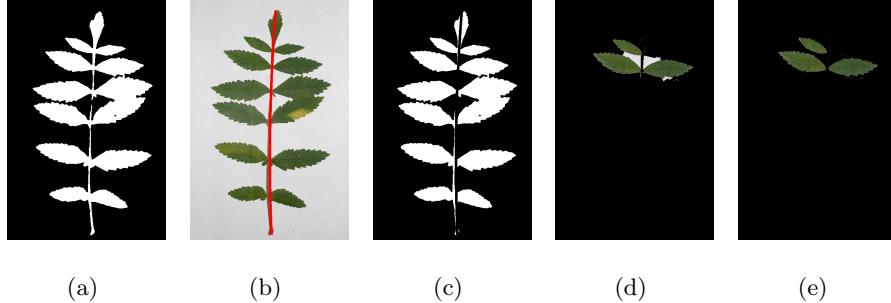


Fig. 4. Category 2 preprocessing: Fig. 4(a) shows the original bitmap image, Fig. 4(b) shows the image with polyfit curve marked in red, Fig. 4(c) shows the bitmap after stalk elimination, Fig. 4(d) shows the three extracted leaflets prior to noise and stalk removal, Fig. 4(e) shows the final extracted leaflets.

petiole fragments around the leaflet. We resolve these in the next steps. Fig. 4(d) shows the three extracted candidates.

5. **Noise Removal:** We use a simple range filter to address the background fragments. We first compute the average RGB values for the background by traversing around the periphery of the original image. We then account for the background noise pixels by examining each pixel in the periphery and deselecting it (from the binary mask) if it falls within a particular range of the average background color.
6. **Stalk Removal:** The strategy used for stalk removal for category 1 images cannot be applied to the leaflets of compound leaves. This is because, unlike the single leaf images, the leaflets are not vertically oriented and upright. Hence, an erosion-based scheme is used to eliminate the stalk from the leaflet bitmap. The inherent assumption in this scheme is that the erosion operation does not affect the shape and margin of the leaflets. Although this step did not completely solve the problem, it did manage to reduce the stalk content (if present) in the image. Fig. 4(e) shows the three candidate leaflets after noise and stalk removal.
7. **Best Candidate Selection:** The output of the last step produces three candidate leaflets of which we need to choose only one. While it is easy for a human being to choose the “best” among the three, we wanted to automate the entire process. Hence, we use the following simple method for the same. For each candidate, we compute a 50-dimensional feature vector using complex network shape descriptors (see Sec. 2.3 for a description of the features). We then compute the Euclidean distances between each pair of candidates and find the maximum pairwise distance. The candidate that does not feature in this is in some sense in the middle and was, thus, chosen as the “best” candidate.



Fig. 5. Sample category 3 images.

Category 3 Images:

Category 3 images consist of natural photographs. There is no distinction between single and compound leaves in this category. Fig. 5 illustrates a few images in this category. The images suffer from multiple issues including occlusion, out of focus problem, etc. It is difficult to design an automated segmentation system that is robust to all these issues. We, therefore, resort to a semi-automated, feedback-based segmentation scheme using the GrabCut algorithm [10]. We develop an interface that iteratively seeks input from the user and makes corrections to the segmented image. The user “corrects” the segmentation output by undertaking either of the two following actions:

- Specifying *missing* regions in the current segmentation result (Red)
- Specifying *extraneous* regions in the current segmentation result (Blue)

Fig. 6 shows the output at various stages for segmentation of leaf from a sample category 3 image. The top row denotes the user interactions and the bottom row denotes the output for the corresponding user actions.

2.3 Feature Extraction

The preprocessing phase, as described in Sec. 2.2, results in an image containing a single leaflet which acts as input to the feature extraction phase. The feature extraction phase results in a 90-dimensional feature vector comprising of geometrical, shape and texture features for each image. On a coarse level of granularity, the feature vector can be divided into three categories: 50 complex network, 28 tooth and 12 morphological features. These are described next.

Complex Network Features:

The motivation behind using complex network features [1] as shape descriptors lies in the fact that the data contains noise (even after preprocessing) and these features are robust, noise tolerant, and scale and rotation invariant. The complex network construction procedure requires as input the contour of an image represented as a collection of pixels on the 2D plane. A graph $G = (V, E)$ is built where each pixel of the contour collectively forms the vertex set V and undirected edges between each pair of vertices form the edge set E . The Euclidean distance between the points of each such pair defines the weight of an edge (normalized within the interval $[0, 1]$).

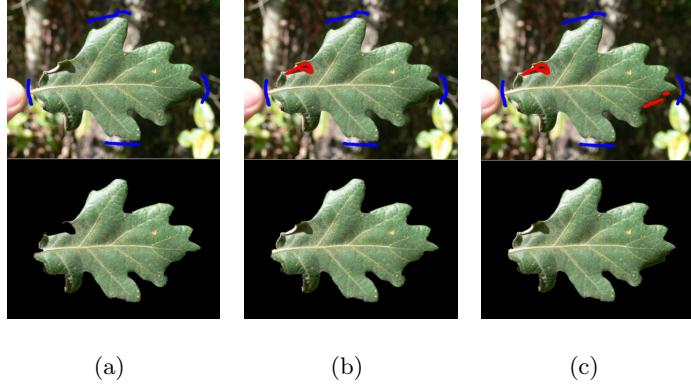


Fig. 6. Category 3 user feedback based segmentation: Fig. 6(a) through Fig. 6(c) represent the iterations 1, 2, and 3 respectively.

The graph thus obtained has connections between every pair of vertices and, therefore, cannot be considered as a complex network. We accomplish the transformation of this regular network into a complex network by thresholding the edge weights. We consider views of the network at multiple resolutions (thresholds), each containing different number of edges. An edge is retained if its weight is less than the threshold at that level and discarded otherwise.

The complex network thus constructed provides five different features that can be broadly categorized as *degree descriptors* and *joint-degree descriptors*.

1. **Degree Descriptors:** The degrees of the vertices are first normalized into $[0, 1]$ by dividing by the total number of vertices in the network. Then, the *maximum degree* and *mean degree* are realized as the two degree descriptors:

$$\text{Maximum} = \max_{i=1}^N d_i, \quad \text{Mean} = \sum_{i=1}^N d_i / N$$

where d_i denotes the degree of vertex v_i and N is the total number of vertices.

2. **Joint-Degree Descriptors:** The joint-degree descriptors rely on the correlations between the degrees of vertices connected by an edge. The correlation for a degree d_i is captured by the probability $P(d_i)$ that two neighboring vertices have the same degree d_i . This probability is computed empirically as the ratio of the number of all connected vertices with the same degree d_i to the total number of edges. Using this, the following three descriptors are realized:

$$\text{Entropy} = \sum_{i=0}^D [P(d_i) \log_2 P(d_i)], \quad \text{Energy} = \sum_{i=1}^D [P(d_i)]^2, \quad \text{Sum} = \sum_{i=1}^D [P(d_i)]$$

where D is the maximum degree in the network.

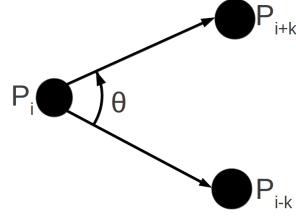


Fig. 7. A tooth point.

We realize the network at 10 different resolutions ranging uniformly from 0.05 to 0.50. At each threshold, we calculate 2 degree descriptors and 3 joint-degree descriptors, thus producing a total of 50 complex network features.

Tooth Features:

We next describe the *tooth features*. A *tooth point* is a pixel on the contour that has a high curvature, i.e., it is a peak. To determine whether a point P_i on the contour is a tooth point or not, we examine the angle θ subtended at P_i by its neighbors P_{i-k} and P_{i+k} (where k is a threshold). Fig. 7 shows an example. If the angle θ is within a particular range, then P_i is a tooth; otherwise, it is not. The upper bound of θ is $\pi/2$, i.e., the right angle. The lower bound was determined by manual observations. The value that gave the best result was $\sin^{-1} 0.8$, i.e., the sine of the angle was constrained to be $0.8 \leq \sin \theta \leq 1$. The feature value for a leaf at a particular threshold k is the *number* of tooth points on its contour.

Fig. 8 shows the output of the tooth point detection method on a leaf for two different thresholds. As the threshold, k , is increased from 5 to 45, the number of contour points detected as tooth points decreases from 25 to 11.

Since it is possible for two different types of leaves to have nearly the same number of teeth at a particular threshold, we compute the tooth-based features at multiple increasing values of k . While no single threshold may be good enough to distinguish different leaf margins, they should be sufficiently separated when multiple thresholds are used. We use 28 thresholds from 3 to 30.

Morphological Features:

For morphological features, we use the same ones as described in [12]⁴. The automated process, however, expects the image to be aligned along the vertical axis with only a small allowable error. This is to facilitate the curve-fitting operation approximate the upper-most and lower-most points of the main vein of the leaf.

⁴ The authors of [12] were generous enough to provide the code for human assisted extraction of several morphological features. We automate this process and subsequently use the system to compute the required features.

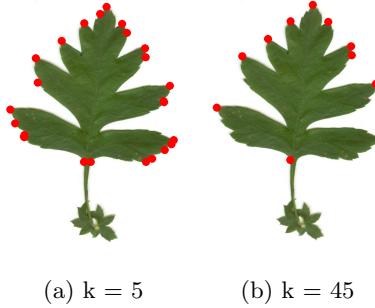


Fig. 8. Tooth detection at two different thresholds.

Vertical orientation was performed using principal component analysis (PCA). It was observed that one of the two principal components generated by PCA is along the direction of the main vein. This works due to the fact that most of the leaflets are either ellipsoidal or palmately lobed in shape. The ellipsoidal leaflets have the main vein along the major axis of the ellipse that fits the leaflet. For palmately lobed leaflets, the two sides of the leaflet along the main vein are almost symmetric, and hence, the main vein is along one of the principal components.

To determine whether a leaflet is ellipsoidal in nature, we first compare its area with the minimum enclosing ellipse. If they match well, we consider the leaflet to be elliptical in shape. To vertically orient the leaflet along the main vein, we then rotate the image along the first principal component of the ellipse.

For palmately lobed leaflets, on the other hand, we associate a symmetry measure with each principal component. This measure checks the similarity between the two sides of the leaflet along the axis represented by that principal component. The one with a higher symmetry measure is considered to be along the main vein and the leaflet is rotated along that direction.

This vertical orientation process renders the task of marking endpoints of the main vein automatic, which in turn, makes computing the set of 12 morphological features as mentioned in [12] automatic as well. The features are computed from a set of 5 basic geometric parameters (Fig. 9) described next.

1. *Diameter, D*: The diameter of the leaf is the longest distance between any two points on the closed contour defined by the leaf.
2. *Physiological Length, L*: The physiological length refers to the length of the line connecting the two terminal points of the main vein in the leaf.
3. *Physiological Width, W*: The physiological width refers to the distance between the two endpoints of the longest line segment perpendicular to the physiological length.
4. *Leaf Area, A*: The leaf area is the number of pixels in the final preprocessed (binary) image constituting the leaf part.
5. *Leaf Perimeter, P*: The leaf perimeter is the number of pixels along the closed contour of the leaf.

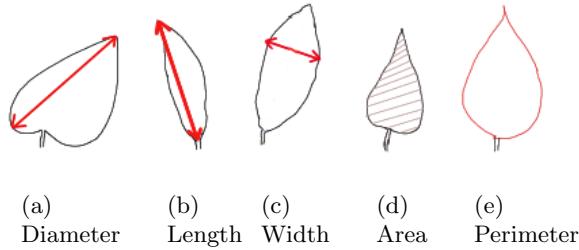


Fig. 9. The five basic morphological parameters.

These geometric parameters yield the following 12 morphological features:

1. **Smooth Factor:** This is defined as the ratio of the area of the image smoothed by a 5×5 averaging filter to that smoothed by a 2×2 averaging filter.
2. **Aspect Ratio:** This is defined as the ratio of physiological length to physiological width, i.e., L/W .
3. **Form Factor:** This measures the “roundness” of the leaf and is computed as $4\pi A/P^2$.
4. **Rectangularity:** This measures how rectangular the leaf is and is computed as $L.W/A$.
5. **Narrow Factor:** This measures the “narrowness” of the leaf and is defined as D/L .
6. **Perimeter Ratio of Diameter:** This is defined as the ratio of the perimeter of the leaf to its diameter of the leaf, i.e., P/D .
7. **Perimeter Ratio of Physiological Length and Physiological Width:** This is defined as the ratio of the perimeter of the leaf to the sum of its physiological length and physiological width, i.e., $P/(L + W)$.
8. **Vein Features:** The skeletal structure of a leaf is defined by the vein patterns which play an important role in distinguishing the leaves that have similar shape. The standard and most widely used procedure for computing the vein features is to perform a morphological opening operation on the grayscale image. A flat, disk shaped structuring element of radius r is used and the resultant image is then subtracted from the contour of the leaf. The output resembles the vein structure of a leaf on which we compute the following 5 features: A_1/A , A_2/A , A_3/A , A_4/A , A_4/A_1 where A_r denotes the area of the remaining leaf obtained using a structuring element of radius r , and A is the area of the original leaf as mentioned earlier.

Complex Images:

The proposed preprocessing techniques failed to successfully segment some of the images in the dataset (see Fig. 10 for examples). These leaves are, in a way, *compound of compound* leaves – each branch of a leaf is analogous to

**Fig. 10.** Examples of complex images.

a compound leaf. This distinction of “complex leaves” is done by examining the number of connected components calculated after performing the erosion operation on category 2 images. The leaves with high number of connected blobs (≈ 30) are termed as complex, thereby marking the others as normal compound leaves. Since a successful leaf segmentation is critical to the computation of the proposed features, we bypass the usual feature extraction process for the complex leaves, and instead, use a bag of features model [8].

We first compute the SURF descriptors [2] of many randomly sampled patches from all the complex images. We next cluster the descriptors into 8 groups using k-means clustering. We use these clusters as a code-book [8] of size 8.

Given a complex image, we then extract all its SURF points, and denote each SURF descriptor by the nearest cluster center (equivalently, the code corresponding to it). This results in a histogram of size 8 for every image where bin i represents the number of SURF points that correspond to that particular code. The histogram represents the feature vector of the image.

2.4 Classifier Selection

We tested many classifiers for predicting the type of species, and after extensive experimentation, chose *random forests* [4] as the classifier.

3 Results

We submitted four different runs for the CLEF 2012 plant identification task. All of them used the same preprocessing and classification methodologies and differed on the basis of (i) inclusion or exclusion of the author id metadata field, and (ii) choice between two different parameter sets for the random forest classifier. The two different parameter sets differ on the basis of number of trees in the random forest and the number of attributes (features) considered at each node of the trees to make a split. Table 2 provides a complete description of the classifier parameters.

Run ID	Single Leaf		Compound Leaf	
	# Trees	# Split Features	# Trees	# Split Features
1	350	7	400	7
2	350	30	400	25
3	500	7	400	7
4	350	7	400	7

Table 2. Description of random forest classifier parameters.

Each run was scored based on a scoring function that essentially measures the mean average precision per photographer per species. Mathematically, it is denoted as:

$$S = \frac{1}{U} \cdot \sum_{u=1}^U \left[\frac{1}{P_u} \cdot \sum_{p=1}^{P_u} \left[\frac{1}{N_{u,p}} \cdot \sum_{n=1}^{N_{u,p}} S_{u,p,n} \right] \right]$$

where U denotes the number of photographers that have submitted at least one test image, P_u denotes the total number of images taken by the u^{th} photographer, $N_{u,p}$ denotes the number of images of the p^{th} plant taken by the u^{th} photographer, and $S_{u,p,n}$ denotes a value between 0 and 1 explained next.

For each test image, multiple guesses about its correct class were allowed. The term $S_{u,p,n}$ measures how good the guess (for the n^{th} image of the p^{th} plant taken by the u^{th} photographer) is, and is inversely proportional to the rank of the correct guess. The later the rank, the lower is the score, and the function decreases rapidly.

Table 3 shows the details and final standings of our four submitted runs. Our best submission, Run 3, was the only one that did not use any metadata information. For the other runs, we used the *photographer id* metadata field as the 91st feature as we observed that it improved the classification accuracy significantly for a 10-fold cross validation evaluation. We utilized SMOTE [5] to address the class imbalance in the dataset for only Run 4. Consequently, it ranked higher than Run 1 and Run 2. For a complete description of competition results, we refer the reader to [7].

4 Conclusion and Future Work

In this paper, we have tackled the problem of plant identification using leaf-based features. We used the Pl@ntLeaves II dataset by CLEF as the benchmark to support our results. We proposed novel preprocessing strategies for shadow removal and background noise correction. We also proposed an automated leaflet segmentation procedure for compound leaves. We introduced the use of tooth features to discriminate leaves with similar shapes but different margins. We also implemented an important improvement on the calculation of morphological features by automating the process of detecting the endpoints of the main vein. We finally used a combination of shape, morphological and tooth features with

Run ID	Score _{scan}	Score _{pseudo}	Score _{natural}	Score _{average}	Rank
1	0.37	0.34	0.43	0.38	10
2	0.30	0.25	0.24	0.27	16
3	0.43	0.40	0.49	0.44	2
4	0.37	0.35	0.43	0.38	8

Table 3. Final standings of our submitted runs (Run 3 achieved the 2nd rank).

random forests for classification. Our approach helped us achieve an overall rank of 2nd in the Plant Identification Challenge, CLEF 2012.

In future, we would like to achieve effective automatic segmentation of natural photographs as well. We would also like to work on optimal fusion of different types of features as well as classifier boosting to improve the results.

Acknowledgments

We would like to thank Aditya Nigam, Tejas Gandhi and Manash Pal for their guidance and feedback during the course of the project. We would also like to thank our department for giving us the resources and the freedom to pursue such non-curriculum academic interests.

References

- Backes, A., Casanova, D., Bruno, O.: A Complex Network-based Approach for Boundary Shape Analysis. *Pattern Recognition* **42**(1) (2009) 54–67
- Bay, H., Tuytelaars, T., Gool, L.V.: Surf: Speeded Up Robust Features. In: *ECCV*. (2006) 404–417
- Belhumeur, P., Chen, D., Feiner, S., Jacobs, D., Kress, W., Ling, H., Lopez, I., Ramamoorthi, R., Sheorey, S., White, S., Zhang, L.: Searching The Worlds Herbaria: A System for Visual Identification of Plant Species. In: *ECCV*. (2008) 116–129
- Breiman, L.: Random forests. *Machine learning* **45**(1) (2001) 5–32
- Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.: Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* **16** (2002) 321–357
- Eaton, J.: *GNU Octave Manual*. Network Theory Limited (2002)
- Goëau, H., Bonnet, P., Joly, A., Yahiaoui, I., Barthélémy, D., Boujema, N., Molino, J.: The ImageCLEF 2012 Plant Identification Task. In: *CLEF 2012 Working Notes*, Rome, Italy. (2012)
- Nowak, E., Jurie, F., Triggs, B.: Sampling Strategies for Bag-of-Features Image Classification. In: *ECCV*. (2006) 490–503
- Otsu, N.: A Threshold Selection Method from Gray-Level Histogram. *IEEE Trans. on Systems, Man, and Cybernetics* **9** (1979) 62–66
- Rother, C., Kolmogorov, V., Blake, A.: “GrabCut”: Interactive Foreground Extraction using Iterated Graph Cuts. *ACM Trans. Graph.* **23**(3) (2004) 309–314
- Söderkvist, O.: Computer Vision Classification of Leaves from Swedish Trees. Master’s Thesis, Linkoping University (2001)
- Wu, S., Bao, F., Xu, E., Wang, Y., Chang, Y., Xiang, Q.: A Leaf Recognition Algorithm for Plant Classification using Probabilistic Neural Network. In: *ISSPIT*. (2007) 11–16