

CROSS DATASET EVALUATION OF FEATURE EXTRACTION TECHNIQUES FOR LEAF CLASSIFICATION

Christian Reul, Martin Toepfer and Frank Puppe

Department for Artificial Intelligence and Applied Computer Science,
University of Würzburg, Germany

ABSTRACT

In this work feature extraction techniques for leaf classification are evaluated in a cross dataset scenario. First, a leaf identification system consisting of six feature classes is described and tested on five established publicly available datasets by using standard evaluation procedures within the datasets. Afterwards, the performance of the developed system is evaluated in the much more challenging scenario of cross dataset evaluation. Finally, a new dataset is introduced as well as a web service, which allows to identify leaves both photographed on paper and when still attached to the tree. While the results obtained during classification within a dataset come close to the state of the art, the classification accuracy in cross dataset evaluation is significantly worse. However, by adjusting the system and taking the top five predictions into consideration very good results of up to 98% are achieved. It is shown that this difference is down to the ineffectiveness of certain feature classes as well as the increased severity of the task as leaves that grew under different environmental influences can differ significantly not only in colour, but also in shape.

KEYWORDS

Leaf recognition, cross dataset evaluation, feature extraction, segmentation, HOCS features

1. INTRODUCTION

Computer vision is a rapidly growing field as classification and recognition tasks gained a lot of interest due to the increasing computing capabilities of modern systems. As for plant leaf classification the introduction of a general benchmark in shape of the Flavia dataset [1] led to an increase of publications regarding that topic. Many systems were proposed, mainly to test and compare different approaches, feature classes and classifiers [1-9]. Furthermore, several mobile applications like *Leafsnap* [10] for iOS or *ApLeaf* [11] for Android were developed. They allow quick classification by taking a photo of a leaf on a bright background like a piece of paper.

The vast majority of publications deal with classification tasks within a dataset by using one part for training and the rest for testing. The chief purpose of this work is to examine the expressiveness of results obtained by using these standard evaluation procedures regarding a real world application scenario, i. e. classifying leaves by using a training set that was collected completely independent from the test set. The main difference between classification tasks within a dataset and between two different datasets is that the respective leaves grew in different locations and at a different time. Hence, the environmental influences like temperature, rainfall and solar irradiance can differ quite a lot. Moreover, leaves change over the course of a year because they lose water and therefore turn, at least in most cases, from green to yellow and brown. Unsurprisingly, features that use information about colour become pretty much useless in cross data set classification tasks. The aforementioned factors might have a great bearing on the shape of the leaves as well. This is also indicated by experiments performed by Sulc and Matas [6] which showed that it is possible to determine if leaves of the same species grew in northern or southern France using leaf recognition techniques.

In this work the performance of several established feature classes of varying complexity is evaluated within and across dataset evaluation. For this task a new data set was collected. It consists of ten species and is completely subsumed by the significantly larger MEW dataset. Therefore, it is perfectly suited for experiments on cross dataset evaluation.

The remainder of this paper is organized as follows: In section 2 several notable contributions in the field of plant leaf identification are briefly reviewed. Section 3 introduces the used datasets. The segmentation process and the features are explained in section 4 and 5 respectively. The classification procedure is defined in section 6. In section 7 and 8 the results in both within and cross dataset evaluation are presented and discussed. Section 9 introduces the developed web application and section 10 concludes the paper.

2. RELATED WORK

Many approaches on plant recognition were introduced in the past. This section focuses on contributions that either yielded outstanding results or introduced feature classes or datasets that were used in the course of this work.

The work by Wu et al. [1] proved to be very important for the field of leaf recognition as they introduced the Flavia dataset, which quickly became the standard benchmark for comparing leaf identification approaches. They used basic geometric features and principal component analysis and, despite the simplicity of their approach, achieved a classification accuracy of slightly over 90%.

Kadir et al. provided several publications as well as the Foliage dataset. Many different feature classes were used including polar Fourier transform, colour moments and vein features [2], principal component analysis [3] and gray level co-occurrence matrix, lacunarity and Shen features [4] achieving accuracies of up to 97.2% on the Flavia and 95.8% on the Foliage dataset.

The Middle European Woody Plants dataset was introduced by Novotny and Suk [5]. In the corresponding paper a recognition system using image moments and Fourier descriptors achieved a recognition rate of almost 85% on the MEW dataset which is significantly larger than the Flavia. Furthermore, a web application for uploading leaf pictures and classifying them was provided.

Sulc and Matas [6] proposed an approach that yielded excellent results of 99.5% on the Flavia and highly impressive 99.2% on the MEW. Their newly introduced so called *Ffirst* method is based on a rotation and scale invariant version of local binary patterns (LBP) that are computed both from the leaf interior and the leaf margin. In 2015, their system clearly presents the state of the art.

The freely available iOS application *Leafsnap* was developed by Kumar et al. [10]. After having taken a picture with a smartphone or tablet while using a white background, the user can upload it to a server. An automatic segmentation procedure is performed and the leaf is classified. The dataset currently covers 185 tree species from the north-eastern United States. The only features used are the so-called HOCS-features which proved to be highly descriptive and will be thoroughly evaluated during the rest of this work.

A similar application is available for Android. Zhao et al. [11] employ the same general approach as pictures have to be taken on a bright background. For classification a variation of the established HOG (Histogram of Oriented Gradient) features is combined with colour features using the HSV picture representation and wavelet features. The dataset contains 126 tree species from the French Mediterranean area.

3. USED DATASETS

In the following sections the most popular publicly available datasets are briefly discussed. Furthermore, the newly created Bavaria Leaf Dataset (BLD) and a combination of those datasets are introduced.

3.1. Publicly Available Datasets

Flavia [1] - consists of 32 species and a total of 1907 instances, mainly collected in the Yangtse Delta, China. It is the most frequently used dataset for the purpose of comparing the performance of leaf recognition systems. The established evaluation method is to randomly pick 40 instances per species for training and 10 of the remaining instances for testing (10 x 40).

Foliage [2] - is divided into a training and a test set to maximize comparability. The former contains 100 images of each of the 60 species, the latter 20.

Middle European Woody Plants (MEW) [5] – was collected in Central Europe. Each of the 153 species is represented by at least 50 instances. For all 9745 instances binary images are provided as well. Due to its large number of leaves the variety of species and the high quality of images the MEW provides a great common ground to compare performances of different leaf recognition systems.

Intelligent Computing Laboratory (ICL) [12] – the largest dataset used in this work contains 16.851 leaves from 220 species of Chinese trees. The number of instances per species differs from 26 to 1078.

Swedish Leaf Dataset (SLD) [13] – consists of 75 images of 15 common trees from Sweden. The established evaluation method is 25 x 50.

The leaves pictured in the images of the Flavia and Foliage datasets are already segmented and their petioles were removed beforehand. The images in the MEW, ICL and SLD were created by scanning each leaf without removing the petiole first. Two examples of leaves from each dataset can be seen in Figure 1.



Figure 1. Example leaves from the Flavia (column 1), Foliage (2), SLD (3, 4), MEW (5, 6) and ICL (7, 8) dataset.

3.2. Bavaria Leaf Dataset (BLD)

On occasion of this work a new dataset was collected. It consists of leaf images of trees which are common in Bavaria, Germany. In contrast to the publicly available datasets mentioned above the leaf images in the BLD are not scans, but actual photographs taken by different digital and smartphone cameras of varying quality. About half of the leaves were picked from trees, placed on sheets of paper and photographed to simplify automatic segmentation. No special attention was paid to petioles. The rest of the leaves were photographed while still being attached to the respective tree. This led to a variety of very different and complex backgrounds. Figure 2 shows some leaves from the BLD. It can be seen that leaves with missing pieces (upper middle), abnormal spots (bottom left) or of questionable image quality (bottom right) were kept. For each

species in each subset at least 65 instances were collected. Altogether, the dataset consist of 878 leaves photographed on paper and 858 leaves attached to a tree.



Figure 2. Examples from the BLD on paper (top) and still attached to the tree (bottom).

Table 1 shows the species used in the BLD. An important characteristic of the BLD is that all of its ten species also feature in the much bigger MEW. Therefore, it can be used as test set for the cross dataset evaluation task.

Table 1. Species of the BLD.

Scientific Name	Acer platanoides		Scientific Name	Fagus sylvatica	
Common Name	Norway maple		Common Name	European beech	
Scientific Name	Acer pseudo-platanus		Scientific Name	Populus tremula	
Common Name	Sycamore maple		Scientific Name	European aspen	
Scientific Name	Alnus glutinosa		Scientific Name	Quercus robur	
Common Name	Black alder		Common Name	English oak	
Scientific Name	Betula pendula		Scientific Name	Quercus rubra	
Common Name	Silver birch		Common Name	Northern red oak	
Scientific Name	Carpinus betulus		Scientific Name	Tilia cordata	
Common Name	European hornbeam		Common Name	Small-leaved lime	

3.3. Combination of the Publicly Available Datasets

To ensure an even more realistic evaluation scenario the five publicly available datasets used in this work were combined to a superset called “All Combined” (AC). It consists of 430 species with a total of almost 36,000 instances. Notably, the overlap between the five initial datasets is relatively small. By combining the datasets the number of species only drops from 480 to 430.

4. SEGMENTATION

Before the different features can be extracted, the leaves have to be segmented. This includes removing the background as well as the petiole if present. In this work two types of segmentation procedures are performed. Leaves photographed on a piece of paper get automatically segmented, while the segmentation of leaves which are still attached to the tree need user interaction to yield quality results. In this section both approaches will be briefly described. Firstly, the algorithm behind both segmentation techniques, *GrabCut*, will be introduced.

4.1. The Graph-/GrabCut Algorithm

The Graph-/GrabCut algorithm was developed by Boykov and Jolly [14], refined by Rother et al. [15]. The basic idea is to transfer the input image into a graph, in which the vertices represent the pixels and the edges quantify the degree of similarity between adjacent pixels. The more similar two pixels are the higher the edge weight of their linking edge is. Every pixel is connected to its four direct neighbours and two terminal nodes which represent the current foreground and background model. After constructing the graph the actual segmentation is done by performing iterated minimum cuts. A cut severs edges until there is no path joining the terminals anymore. The result is called minimum cut when the sum of the weights of the severed edges is minimal.

The OpenCV library [16] offers an effective implementation of the described algorithm. To allow user interaction a mask is used to initialize the segmentation process. This input mask has the same dimensions as the input image. One of four possible values has to be assigned to each pixel: sure foreground, sure background, probable foreground or probable background. These values influence the edge weights and therefore the segmentation result. For example, if a pixel is considered as sure foreground, the weight of edge linking it to the background terminal will be set to zero. The edge connecting the pixel to the foreground terminal will be assigned a very high weight, ensuring the inseparability of the edge.

4.2. User-assisted Segmentation

GrabCut was primarily developed to allow user-assisted segmentations that are too difficult or too specific to be handled automatically. In this work the segmentation of leaves that are still attached to the tree can be very challenging because of the varying background. However, a simple GUI program allows highly effective segmentation as shown in Figure 3.



Figure 3. Assisted segmentation using GrabCut: initialization (left), first result and adjustments (middle), final segmentation result (right).

The initial segmentation is achieved by the user drawing a rectangle (red) that encloses the desired leaf. The GrabCut algorithm considers every pixel inside the rectangle as probable foreground and every pixel outside as probable background. Using this simple input mask the first result is calculated and the user is able to perform slight adjustments in the overlap image in which the current foreground is marked. This is done by manually labelling sure foreground (green) and sure background (blue) pixels. Based on the changed input the algorithm computes an updated segmentation until a satisfying result is obtained. The described system is very efficient and provides excellent results.

The BLD tree subset was segmented by using the method described above. Furthermore, a gold standard for the paper subset was created.

4.3. Automatic Segmentation on Paper

The input mask for the automatic segmentation process is constructed using the A- and S-channel from the LAB and HSV representations of the image. In the A-channel shadow pixels become almost invisible, while in the S-channel it is ensured that the white background is completely black. Binary representations are achieved by applying Otsus Method [17]. The obtained foreground from the A-channel can be considered as sure foreground, the obtained background from the S-channel as sure background. The not yet assigned pixels are mostly shadow or darker spots at the tips of the leaf. Especially for the shadow pixels, it is often next to impossible to make a profound prediction if they are indeed shadow and therefore background or if they belong to the leaf. To provide the GrabCut algorithm with at least some kind of tendency two heuristics are applied: Firstly, edges are way more likely to occur within or at the outer contour of the leaf than in the shadow region. Hence, an edge detection is performed on the A-channel image and the detected edges are considered to be probable foreground. Secondly, in general, the leaf regions in the S-channel image are brighter than the shadow regions. Therefore, another Otsu binarisation is applied that only considers those pixels which have not yet been assigned a value in the GrabCut input mask. According to the hereby achieved separation, the pixels are considered as probable foreground and background respectively. An outline of the segmentation process is shown in Figure 4.

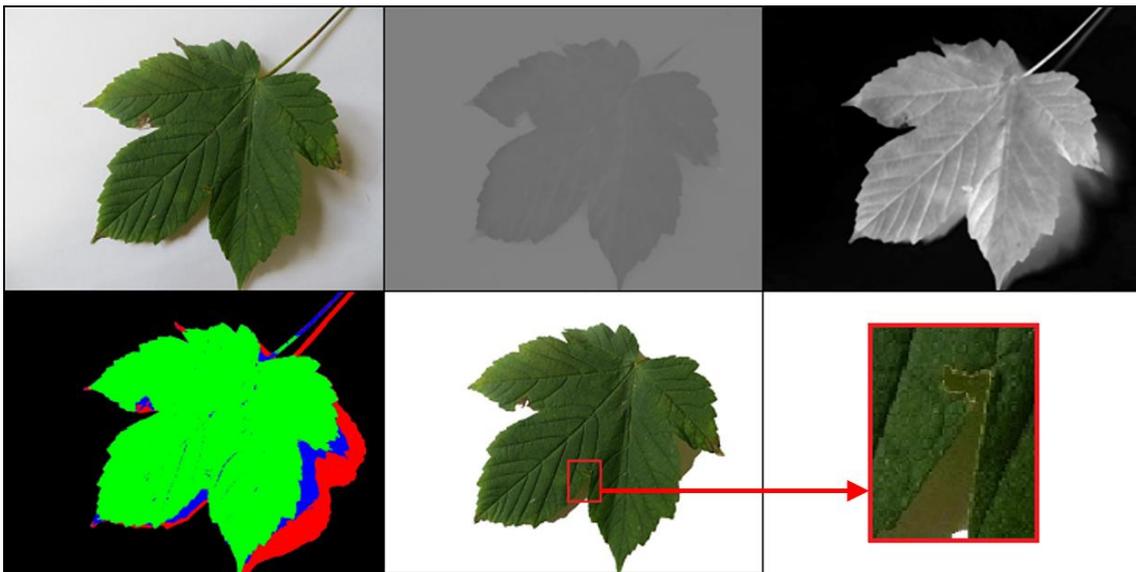


Figure 4. Automatic segmentation process. Top: Original (left), A-channel (middle), S-channel (right); bottom: GrabCut input mask (left), segmentation result (middle), failed segmentation example (right).

The assigned colours in the GrabCut input mask are: sure foreground (green), sure background (black), probable foreground (blue), probable background (red). It has to be mentioned that the obtained result in this example is significantly worse than the average result which will be discussed in further detail in section 4.5. However, this example shows the severity of the task as it can be observed that the shadow in the critical regions shows a clear tinge of green and therefore looks very similar to the leaf itself.

4.4. Removing the Petiole

If a petiole is present, it quite often vanishes during the GrabCut segmentation as it might get cut off from the leaf and only the biggest connected contour is kept. However, if a petiole is still attached, it has to be removed. In order to achieve that a simple procedure similar to the one used in [10] is applied. At first, a morphological transformation called top-hat is performed on the binary image of the segmented leaf. As a result, areas which are brighter than their immediate neighbourhood are highlighted. The obtained regions are considered as potential petioles. To detect the most likely petiole the candidates are checked for several conditions: To be considered any further a candidate has to be bigger than a certain minimum size. Furthermore, the removal of a candidate from the original picture obviously must not cause a change of the number of connected components as the petiole is simply attached to the leaf. Finally, the most elongated candidate is chosen and removed from the original image.

4.5. Results

The achieved results are compared to the gold standard and the error percentage for each image is calculated: The number of misclassified pixels is determined and for purposes of normalization is divided by the total number of foreground pixels in the gold standard image. The average error rate was found to be 4.10%, which is significantly lower than the results obtained by standard segmentation approaches like Otsu (10.37%), K-Means (8.75%) and Watershed Transformation (5.06%, using a pre-processing technique similar to the one described in 4.3.). Figure 5 shows two example leaves whose error rate is almost equal to the average.

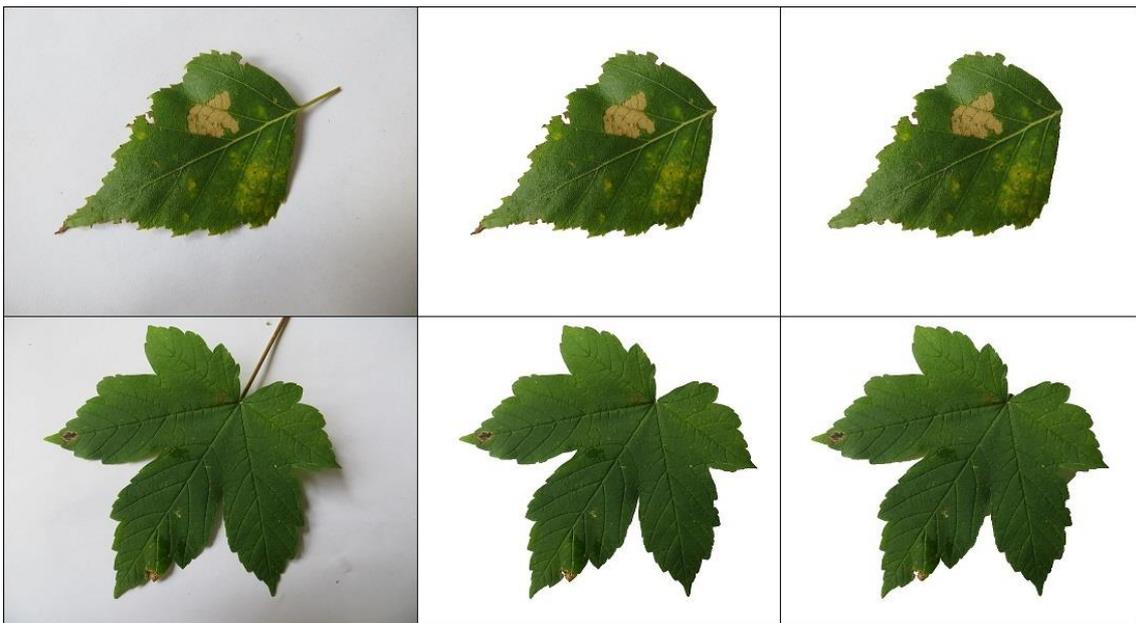


Figure 5. Two example segmentations: original (left), gold standard (middle), automatic segmentation (right).

The proposed segmentation method shows very good results on a consistent basis as 92% of the test images had a segmentation error of lower than 6%. The downside of this approach is that the GrabCut algorithm is complex and therefore relatively slow. Depending on the used hardware the segmentation of an 800 x 600 image can take one to two seconds. This time could be brought down significantly by scaling down the images first. However, in this work one to two seconds are deemed acceptable as the segmentation results are excellent.

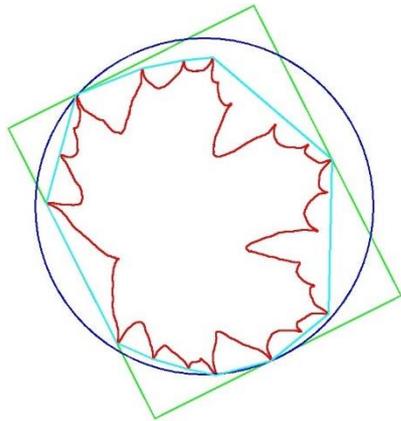
5. FEATURES

In the past a wide range of different feature classes were used to classify leaves. The focus of this work is not the introduction of new ones, but to provide a more in depth evaluation of a subset of already existing ones.

In this section the six feature classes which were used during this work will be described briefly. As a prerequisite, all features had to be rotation-invariant. For some feature classes rotational invariance can only be achieved by major adaptations or even not at all. Nevertheless, this prerequisite was kept, because the alignment of the leaves is random in most of the datasets. Of course, it is possible to automatically detect the orientation of a leaf and rotate it into a uniform position, but it is a tricky task to perform consistently. Furthermore, misclassifications caused by errors during the alignment detection process would be misleading while evaluating the feature performance.

5.1. Contour Features

One of the most obvious ways to characterize the general form of a leaf is to describe its proportions. Inspired by [1] five features were derived by using the outer contour of the leaf, its convex hull and its minimum bounding rectangle, which can be seen in Figure 6. Several other features can be derived, for example by using the contour's minimum enclosing circle. But preliminary tests showed no significant improvement was achieved by adding further features.



$$\text{AspectRatio} = \frac{\text{Width}}{\text{Length}}$$

$$\text{Rectangularity} = \frac{\text{ContourArea}}{\text{RectangleArea}}$$

$$\text{ConvexHullAreaRatio} = \frac{\text{ContourArea}}{\text{ConvexHullArea}}$$

$$\text{ConvexHullPerimeterRatio} = \frac{\text{ContourPerimeter}}{\text{ConvexHullPerimeter}}$$

$$\text{PerimeterLengthWidthRatio} = \frac{\text{ContourPerimeter}}{\text{Length} + \text{Width}}$$

Figure 6. Leaf Contour (red) with its related minimum bounding rectangle (green), convex hull (light blue) and minimum enclosing circle (dark blue).

5.2. Curvature Features

Another way to characterize the margin of a leaf are the distances between the contour pixels and the centre of gravity of the contour. For N Pixels $P_i(x_i, y_i)$ the centre of gravity $C(x, y)$ can be calculated as follows:

$$C(x, y) = C\left(\frac{1}{N} \cdot \sum_{i=1}^N x_i, \frac{1}{N} \cdot \sum_{i=1}^N y_i\right)$$

Subsequently, the distances to the contour pixels can be computed:

$$\text{dist}(P(x_i, y_i), C(x_c, y_c)) = \sqrt{(x_c - x_i)^2 + (y_c - y_i)^2}$$

From these distances five curvature features similar to the ones used by [7] are derived:

- *MinDistanceRatio*: minimum distance divided by the average distance.
- *MaxDistanceRatio*: maximum distance divided by the average distance.
- *StandardDevRatio*: standard deviation of the distances divided by the average distance.
- *ZeroCrossingRate*: number of conversions during a clockwise iteration over all points in which “+” describes a distance value bigger than the average and “-“ a smaller one divided by the total number of contour points.
- *TopPeaks*: number of distances bigger than the average divided by the total number of contour points.

5.3. Colour Features

To examine the significance of the colouration of the leaves the four statistical moments mean μ , standard deviation σ , skewness ν and kurtosis γ are used:

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i \quad \sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2} \quad \nu = \frac{\sum_{i=1}^N (x_i - \mu)^3}{N\sigma^3} \quad \gamma = \frac{\sum_{i=1}^N (x_i - \mu)^4}{N\sigma^4} - 3$$

The calculation is performed for each colour channel (red, green and blue) as well as for the grayscale image. This approach was used by [3] and leads to 16 colour features altogether.

5.4. Hu Features

In 1962 Hu introduced seven moments which are able to describe the shape of a contour in a scale-, translation- and rotation-invariant way by combining its central moments in a linear way. For an in detail description of the highly mathematical procedure see [18].

5.5. HOCS Features

The Histogram Of Curvature over Scale features were introduced by [10] who used them as their only feature class in their leaf recognition iOS app *Leafsnap*, which yielded excellent results. The basic idea of the feature extraction process is shown in Figure 7.

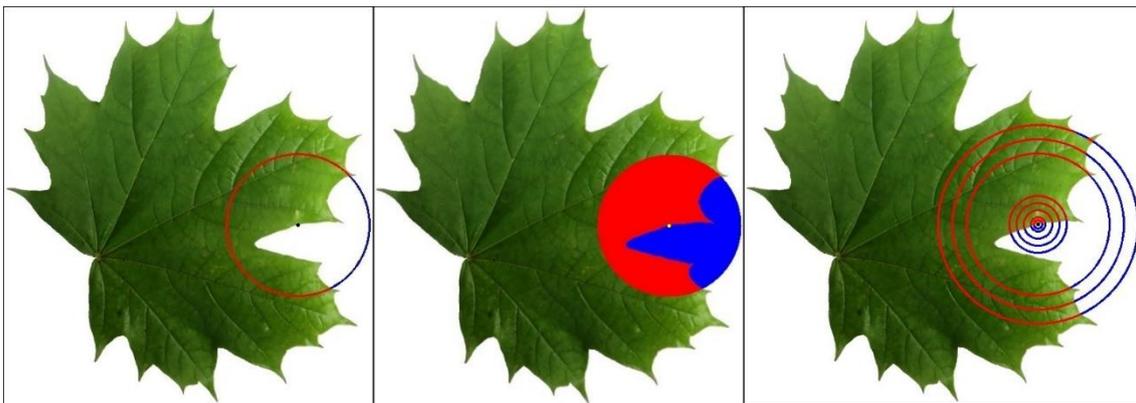


Figure 7. HOCS-Features: arc (left), area (middle), used radii (right).

There are two subclasses of HOCS features: arc and area features. To determine the arc features a circle of a given radius is drawn around every contour point. Then the ratio of foreground pixels covered by the outline of the circle (red) to the total number of circle pixels (red + blue) is

calculated and the result is stored in a histogram using ten bins. The calculation of the area features takes place analogously.

Prior to the feature extraction process all leaf contours are resized to a common area of 30,000 pixels. In preliminary tests the best results were achieved by using a set of eight different radii: 3, 5, 10, 15, 20, 50, 60 and 70 pixels. It is worth mentioning that additionally medium sized radii of for example 30 or 40 pixels did not seem to have a positive impact on the classification performance. All in all, 160 single HOCS features are extracted: ten bins for arc and area features respectively, calculated for eight radii.

To demonstrate the principle of operation of the HOCS features the feature extraction process is performed on two leaves, which can be seen in Figure 8.

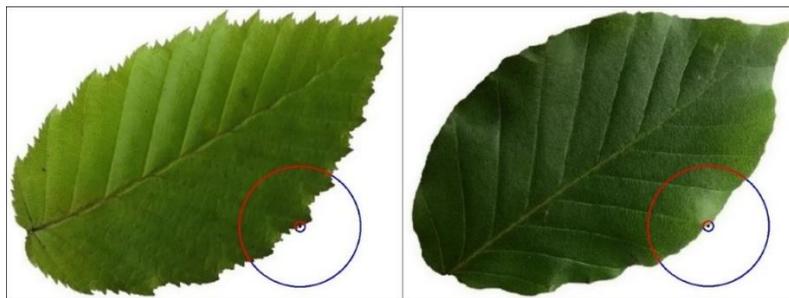


Figure 8. Example leaves of *Carpinus betulus* (left) and *Fagus sylvatica* (right).

It can easily be seen that both leaves are quite similar in terms of their coarse form: egg-shaped and hastate, but the fine structure of their leaf margin differs quite a lot: serrated on the left and smooth on the right. To show how the HOCS features model these kinds of similarities and differences the feature extraction process is performed for two radii: a big one (60 pixels) and a small one (5 pixels). The obtained results are shown in Figure 9.

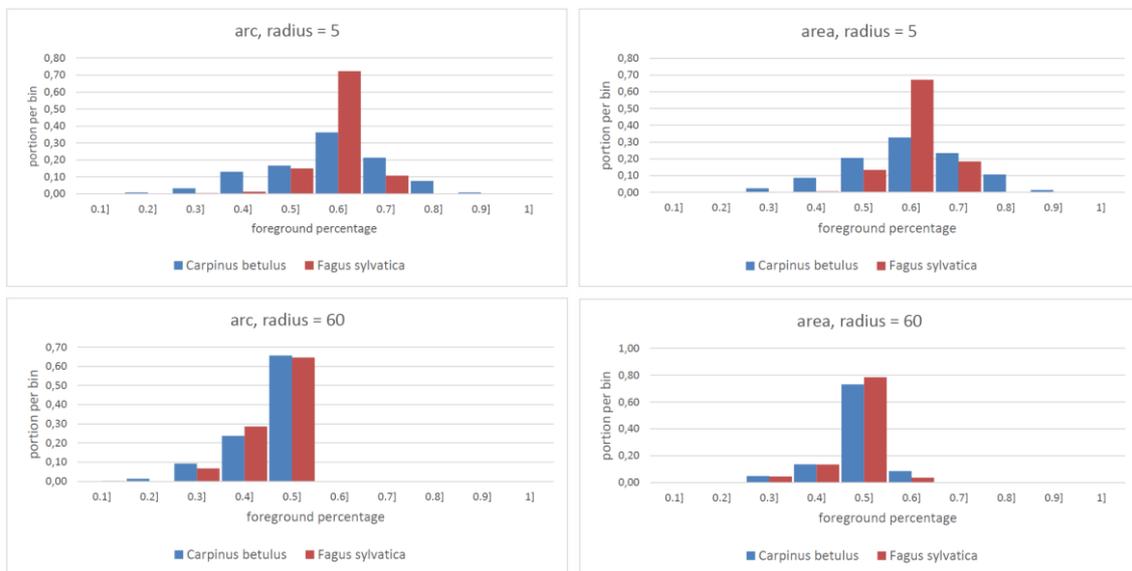


Figure 9. HOCS-Features example: small radius (top), big radius (bottom).

5.6. Binary Pattern Features

In this work a simplified version of the original binary patterns initially introduced by Ojala et al. [19] was tested: Instead of a grayscale image a binary representation is used to calculate the features in which “1” describes a leaf pixel and “0” a background pixel. For each contour pixel

its circular neighbourhood with radius r is computed and the resulting circle is divided in p equidistant pixels. The different features for one circle is simply the count, how many consecutive pixels belong to the leaf. If there are two or more partitions of pixels belonging to the leaf, which are interrupted by background pixels, this is viewed as a separate feature. These binary patterns can be viewed as a simplification of the HOCS arc feature, which measure the leaf part of the circle and the background part of the circle as a number and not as count of equidistant points. Note that the special case, whether the circle cuts the leaf several times, is not modelled within the HOCS features. Figure 10 shows an example of the HOCS arc features and the equivalent binary pattern feature with $p = 8$ as well as the 17 radii used in this work: 1, 4, 6, 7, 9, 12, 17, 20, 22, 25, 30, 32, 35, 37, 40, 42 and 45 (right). Notably, the white pixel marked with a square presents a borderline case. This already hints at possible shortcomings of the BP features, especially when using large radii.

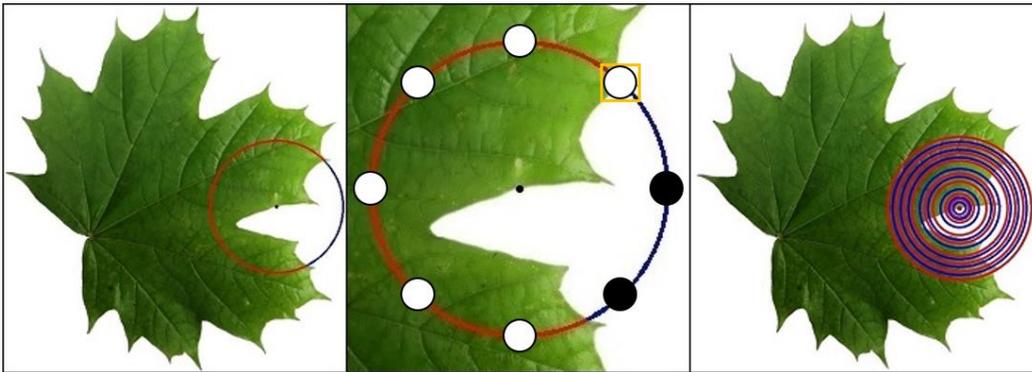


Figure 10. Relation between the HOCS arc features (left) and the BP features (middle).

5.7. Overview

The final feature vector of each instance consists of 365 features: 5 contour features, 5 curvature features, 16 colour features, 7 Hu features, 170 binary pattern features and 160 HOCS features. Furthermore, there is an ID feature which stores all necessary information about an instance, for example “[MEW] Acer platanoides_0”.

6. CLASSIFICATION

For classification the Weka data mining and machine learning tool was used [20]. It offers more than 70 classifiers and many helpful pre-processing applications like the normalization of attributes. Moreover, it allows easy integration of additional machine learning libraries such as LibSVM [21].

6.1. Used Classifiers

In this work two classifiers were used: the Weka implementation of the K-Nearest-Neighbours classifier (KNN) and a support vector machine (SVM) provided by LibSVM and executed via Weka.

6.2. Parameter Optimization

One of the biggest problems in classification tasks is overfitting. In this work especially the binary pattern and HOCS features need highly optimized parameters to work properly. To make sure that the proposed system generalizes well the following optimization procedure was used:

All parameters were optimized by using the Flavia dataset exclusively. For a start, this includes feature parameters such as the binary pattern and HOCS radii or the bin distribution in histograms. Preliminary tests on feature classes like the ones using the *MinimumEnclosingCircle* mentioned

in 5.1 were performed on the Flavia data set as well without exceptions. The same applies to all classifier parameters like the number of considered nearest neighbours K for the KNN or the kernel type and kernel parameters for the SVM.

After the optimization phase all parameters remained completely untouched during the tests on the other datasets.

6.3. Classifier Parameters

Using the KNN the best results were achieved with a K value of 1 and the Euclidian distance. Considering only the single nearest neighbour obviously increases the variance of the classification procedure. It also increases the presentiveness to a maximum.

The SVM yielded the best results while using a Radial Basis Function kernel:

$$K(x_i, x_j) = e^{(-\gamma \|x_i - x_j\|^2)}, \gamma > 0$$

The optimal values for γ and the cost parameter C were determined using a grid search as suggested by [22]. The best result was achieved with values of $\gamma = 0.335$ and $C = 20$.

6.4. Choice of Classifier

Both classifiers were tested using the configurations described above. A 10-fold-cross-validation was chosen as evaluation method. After ten classification runs on the Flavia dataset the SVM achieved a classification rate of $99.41\% \pm 0.02\%$. The KNN performed slightly better by reaching $99.61\% \pm 0.02\%$. Obviously, this tiny difference does not prove the superiority of the KNN. However, considering the KNN offers way better traceability it was used as the main classifier for the remaining classification tasks. On a different note, the KNN additionally allows the efficient usage of the 1 x all evaluation method, which proved to be very useful because of its complete lack of variance.

7. EVALUATION OF THE CLASSIFICATION WITHIN A DATASET

The standard approach in leaf classification tasks is to use only one dataset at a time and split it up to obtain a test and training set. In this section the achieved results of the proposed approach will be presented and compared to other leaf recognition systems. However, at first, the significance of the individual feature classes will be evaluated in detail.

7.1. Feature Performance

The feature performance was tested on two datasets: the Flavia dataset, which was used for parameter optimization, and the significantly larger MEW.

First of all, the performance of each feature class was evaluation on its own. The results can be seen in Table 2. As expected, all feature classes perform better on the Flavia dataset as it is significantly smaller than the MEW (32 species compared to 153). Furthermore, the HOCS features achieve the by far highest accuracy as they allow an exact representation of the coarse shape of a leaf as well as of small variations of the margin. The BP features perform very well on the Flavia dataset, but only provide an average result on the MEW. Apparently, the BP features suffer much more from overfitting than the HOCS features. It is worth mentioning that the colour features perform quite well on both datasets, considering they scored the third (Flavia) and second (MEW) best classification results. The accuracies of the contour and curvature features are almost cut in half when switching to the MEW. As both feature classes are not prone to overfitting, the best explanation for this gap is that simple feature classes are able to distinguish relatively well between a small to medium number of leaf species, but are not sophisticated enough to repeat this performance on a very large dataset like the MEW.

Table 2. Classification results of all feature classes on their own.

Flavia; 1 x all; 1NN		MEW; 1 x all; 1NN	
Features	Result	Features	Result
Contour	77.45%	Contour	40.55%
Curvature	72.78%	Curvature	35.76%
Colour	78.81%	Colour	62.34%
Hu	43.84%	Hu	31.08%
BP	88.46%	BP	51.11%
HOCS	95.86%	HOCS	84.99%
All	99.69%	All	95.66%

Due to these results the entire system is constructed by iteratively adding feature classes which improve the current setup the most. For both datasets the HOCS features mark the starting point. Table 3 shows the obtained results.

Table 3. Step by step construction of the system.

Flavia; 1 x all; 1NN		MEW; 1 x all; 1NN		Flavia; 1 x all; 1NN		MEW; 1 x all; 1NN	
Features	Result	Features	Result	Features	Result	Features	Result
<i>HOCS +</i>	95.86%	<i>HOCS +</i>	84.99%	<i>HOCS +</i>		<i>HOCS +</i>	
Contour	96.43%	Contour	85.49%	<i>Colour +</i>	99.63%	<i>Colour +</i>	95.21%
Curv.	96.70%	Curv.	86.83%	<i>BP +</i>		<i>Hu +</i>	
Colour	98.74%	Colour	94.19%	Contour	99.63%	Contour	95.48%
Hu	96.28%	Hu	88.13%	Curv.	99.69%	Curv.	95.52%
BP	97.17%	BP	87.08%	Hu	99.63%	BP	95.63%
<i>HOCS + Colour +</i>	98.74%	<i>HOCS + Colour +</i>	94.19%	<i>HOCS + Colour +</i>	99.69%	<i>HOCS + Colour +</i>	95.63%
Contour	98.90%	Contour	94.33%	<i>BP +</i>		<i>Hu +</i>	
Curv.	98.79%	Curv.	94.84%	<i>Curv. +</i>		<i>BP +</i>	
Hu	98.85%	Hu	95.21%	Contour	99.69%	Contour	95.63%
BP	99.63%	BP	94.51%	Hu	99.69%	Curv.	95.68%
				Full system	99.69%	Full system	95.66%

On both datasets the inclusion of the colour features improves the classification results more than any other feature class. Although suspicious on the first look, this can easily be explained. The colour features are the only feature class which does not represent information about the shape of a leaf. All other classes keep redundant information, whereas the colour features add a whole new dimension. Again, it can be observed that the BP features seem to be much more significant when using the MEW. It is worth mentioning that the Hu features surprisingly have a notable effect on a system consisting of HOCS and colour features when the larger MEW is used. Contour and curvature features barely have an impact on the achieved classification rates.

7.2. Comparison of Results

Table 4 compares the achieved results to other systems. The used datasets (DS) and evaluation procedures (EP) are specified at the top. For evaluation procedures effected by variance the given value was calculated using ten runs with random generator seeds of 1-10. If mentioned in the

original source the standard error is also shown for the other systems. All values are given as percentages.

Table 4. Results (%) of the proposed method in comparison to systems from other publications.

DS	Flavia		Foliage	SLD		MEW		ICL	
	10 x 40	1 x all	Tex Tr	50 x 25	1 x all	1/2 x 1/2	1 x all	1/2 x 1/2	1 x all
[7]	97.19								
[9]	97.50								
[3]	97.19		95.00						
[8]				97.92					
[6]	99.70 ±0.30		99.00	99.80 ±0.30		99.20 ±0.10			
[5]	91.53	93.66			96.53	84.92	88.91	79.68	84.62
1NN	99.37 ±0.08	99.69	95.83	97.81 ±0.15	98.74	93.80 ±0.09	95.66	90.19 ±0.07	93.48
SVM	99.29 ±0.09	-	95.17	98.35 ±0.12	-	96.54 ±0.10	-	95.28 ±0.10	-

[9]: Not exactly 10 x 40 as more than 40 instances were used for training.

[6]: Results averaged from ten experiments.

For all datasets the proposed approach produces results that are equal to or even better than those achieved by most other systems. Moreover, it has to be said that the method introduced by Sulc and Matas [6] seems to be clearly superior to all other currently known approaches. This becomes especially evident because of the highly impressive result of 99.2% they achieved on the very challenging MEW dataset.

8. CROSS DATASET EVALUATION

In the previous section it was shown that the proposed system provides satisfying results when the classification process is performed within a dataset. However, in this section the main focus of this work will be evaluated: the inter-dataset classification.

8.1. Evaluating the Initial System

To simulate a realistic use case for leaf classification the large MEW dataset is used for training. The BLD subsets *Paper* and *Tree* serve as test sets. Furthermore, for the sake of comparability each of the subset was used to classify the other subset and vice versa. The results are shown in Table 5.

Table 5. Initial results of the cross-dataset classification.

1NN; Full system		
Train	Test	Result
Paper	Tree	87.30%
Tree	Paper	75.97%
MEW	Paper	10.93%
MEW	Tree	25.99%

The classification rates are significantly worse than the ones obtained by the classification within a dataset. Especially, when using the MEW as a training set the results become pretty much useless. The obvious explanation for this deterioration is that some of the feature classes are not suited to be used in an inter-dataset classification scenario.

8.2. Removing Potentially Harmful Feature Classes

The feature classes to be checked first are the colour features and the BP features, which already yielded questionable results. The outcome is displayed in Table 6.

Table 6. Results of the cross-dataset classification after having removed potentially harmful feature classes.

1NN		Results - Full system without...			
Train	Test	-	Colour-F.	BP-F.	Colour-, BP-F.
Paper	Tree	87.30%	85.90%	91.72%	95.34%
Tree	Paper	75.97%	77.68%	95.10%	97.95%
MEW	Paper	10.93%	14.92%	46.70%	60.36%
MEW	Tree	25.99%	22.49%	33.57%	51.17%

The classification results improve drastically after removing the colour and BP features. As expected, the expressiveness of the colour features is reduced because of different environmental influences. The BP features perform even worse in an inter-dataset classification task. There are several reasons for this: First of all the BP features are very prone to overfitting as mentioned before. Furthermore, in this work only eight points per radius were used. For bigger radii the BP features therefore cannot be considered rotation-invariant anymore. This can lead to massive contortions in the classification process. As opposed to the BLD pretty much all leaves in the MEW dataset are placed with their petiole pointing straight down.

After eliminating the prime candidates responsible for the bad classification results the other feature classes are checked again as well. The results are shown in Table 7.

Table 7. Evaluation of the impact of the remaining feature classes after removing colour and BP features.

1NN		Results - Full system without Colour-, BP- and				
Train	Test	-	Contour-F.	Curvature-F.	Hu-F.	HOCS-F.
Paper	Tree	95.34%	95.45%	95.10%	95.34%	74.24%
Tree	Paper	97.95%	97.95%	97.61%	97.72%	79.04%
MEW	Paper	60.36%	60.25%	59.68%	64.46%	15.26%
MEW	Tree	51.17%	51.17%	51.40%	55.01%	14.45%

Unsurprisingly, the HOCS features are indispensable for the classification system. It is worth mentioning that the impact of removing the HOCS features is way higher than in the classification task within a dataset. Again it becomes obvious that the HOCS features are the best engineered feature class used in this work. In comparison, the impact of the remaining feature classes is small. The Hu features are clearly deteriorating the results and are therefore removed. The contour and curvature features give mixed results. As the MEW → Paper task comes closest to a realistic scenario of application both feature classes are kept. Thus, the final system used for the rest of this work uses 171 features: 5 contour features, 5 curvature features, 160 HOCS features and 1 ID feature.

8.3. Further Evaluation of the Final System

After the final system had been established, further evaluation had to be conducted. Firstly, this includes not only paying attention to the top classification result, but also to the top x results. Moreover, the system is further evaluated by using an even bigger dataset for training and a separately collected evaluation dataset for testing.

8.3.1. TopX Evaluation

Even though the results improved significantly after removing colour, BP and Hu features, a classification accuracy of under 65% on paper is still not enough to yield satisfying results in a real world application. Additionally to the already mentioned usual problems of cross dataset evaluation, the high difficulty level of the classification task does not allow better results. The main problem is that there are many different species without any distinctive characteristics. Especially, egg-shaped leaves with a smooth or slightly wavy margin are often almost impossible to distinguish. To still improve the usability of the system a concept called *TopX* is introduced. The idea is to not only compare the top rated species by the classifier to the actual species, but to include the second, third, etc. species as well. The results can be seen in Table 8.

Table 8. Results when considering the TopX species.

1NN		Results when considering the TopX species					
Train	Test	Top1	Top2	Top3	Top5	Top7	Top10
Paper	Tree	95.34%	98.37%	99.19%	99.65%	99.77%	100%
Tree	Paper	97.72%	99.77%	99.88%	100%		
MEW	Paper	65.38%	81.09%	86.22%	91.69%	94.42%	96.13%
MEW	Tree	53.32%	72.76%	80.09%	86.50%	89.87%	93.71%

Obviously, this does not improve the classification accuracy of the system in itself. But in a real world application it allows the user to make the final decision on a predefined selection of manageable size. Displaying the top five results seems to be a reasonable choice as it increases the classification accuracy to almost 92% and should not overexert the average user. It is also worth mentioning that by just taking the second most likely classification result into account the accuracy can already be improved significantly.

8.3.2. Using the AC Dataset for Training

Using the MEW for training and the BLD for testing comes close to resembling a realistic application scenario. However, there are a lot more than 153 plant species. Therefore, the AC dataset introduced in 3.3 is used as training dataset, which allows the system to distinguish between 430 species. Of course, that still does not cover even close to all existing plant species. Yet, as it contains all species and instances of all the established publicly available datasets it comes as close as it gets. The achieved results are shown in Table 9.

Table 9. Results when using the AC as training set.

1NN		Results			
Train	Test	Top1	Top2	Top5	Top10
MEW	Paper	65.38%	81.09%	91.69%	96.13%
MEW	Tree	53.32%	72.76%	86.50%	93.71%
AC	Paper	63.10%	76.31%	87.02%	92.81%
AC	Tree	49.48%	64.26%	79.86%	87.19%

As expected, the classification accuracy suffers because of the much larger training dataset. However, the results only deteriorate by about 4.5% on the paper and 7% on the tree subset. Considering that the AC includes almost three times as many species as the MEW this represents a rather mild drop in accuracy.

8.3.3. Using a Separately Collected Evaluation Set for Testing

During the collocation of the original BLD the focus was on collecting as many leaves as possible in a short period of time. Therefore, leaves with obvious shortcomings were not sorted out as long

as they were in a somewhat decent condition, because it was interesting to see how the classification system deals with these leaves. This led to the inclusion of many leaves which showed to be in borderline condition or even worse. An example is shown in Figure 11.

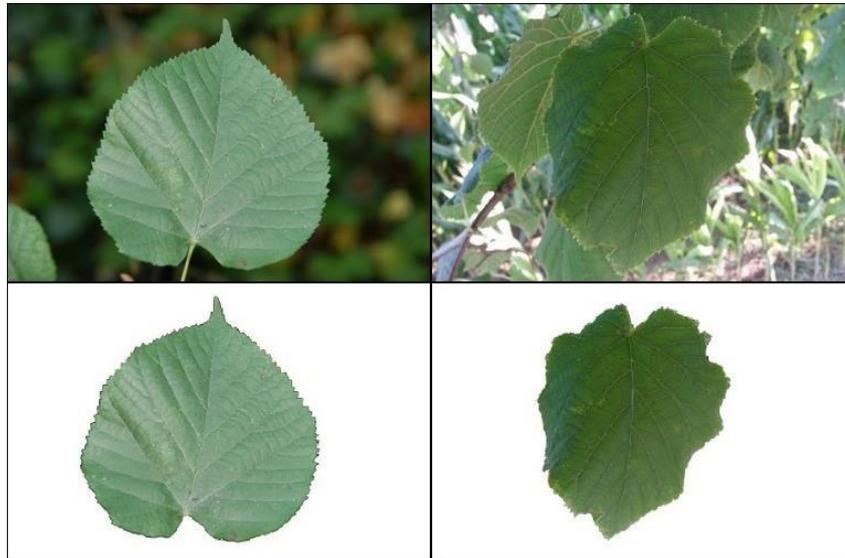


Figure 11. Example of two leaves of *Tilia cordata*: in great (left) and poor (right) condition.

Even though this is an extreme example, obviously, it is important that the leaves are in a condition which allows the classification system at least a fair chance to make a correct prediction.

Therefore, two evaluation subsets were collected in the same region as the original BLD. This time only leaves in good condition were used. It has to be mentioned that the guideline was not to add only leaves in perfect condition, but just to renounce the ones with moderate defects or worse ones. The same ten species were used to provide maximal coverage by the MEW and AC. For each species 20 leaves were photographed while still attached to their respective tree. Afterwards the leaves were collected, placed on a sheet of paper and photographed again. This is another alteration compared to the original BLD in which leaves were not used in both subsets.

After collecting the evaluation sets the classification tasks described in the previous section were exercised. The outcome can be seen in Table 10.

Table 10. Results when using the BLD evaluation datasets.

1NN		Results			
Train	Test	Top1	Top2	Top5	Top10
MEW	EvalPaper	73.5%	87.5%	98.0%	100%
MEW	EvalTree	60.5%	78.5%	93.5%	99.0%
AC	EvalPaper	68.5%	86.5%	94.0%	97.0%
AC	EvalTree	58.0%	75.5%	86.5%	92.5%

As expected, the classification accuracy improved significantly when testing on the evaluation sets. Even when using the AC for training a very good accuracy of 94% was achieved on the paper subset when considering the top five species.

8.3.4. Influence of the Region of Growth

Some of the previous experiments already quite clearly showed that the region and timespan a leaf grew in has a massive impact on its colour and most likely also on its shape. To make this

even clearer one last experiment was performed. As mentioned before, the BLD evaluation set was indeed collected in the same region as the original BLD, but almost an entire year later. In this classification task the AC dataset was extended by adding the instances of the original BLD paper subset. Table 11 shows the achieved results.

Table 11. Results after adding the original BLD to the AC.

1NN		Results			
Train	Test	Top1	Top2	Top5	Top10
AC	EvalPaper	68.5%	86.5%	94.0%	97.0%
AC	EvalTree	58.0%	75.5%	86.5%	92.5%
AC + Paper	EvalPaper	88.0%	95.0%	99.0%	100%
AC + Tree	EvalTree	82.5%	90.5%	94.5%	97.0%

By adding instances from the original BLD the classification results receive a massive boost. The accuracies rise by 5% (paper) and 8% (tree) when the top five species are considered. The Top1 values even go up by almost 20% and 25% respectively. It is worth mentioning that the addition of 878 instances from the original BLD paper subset to the almost 36,000 instances of the AC only represents an increase of a merely 2.5%. As the system used does not include colour features anymore this clearly shows that the region of growth has a huge impact on the shape of the leaves.

9. WEB APPLICATION

The proposed system is available as a web application at www.leafidentification.informatik.uni-wuerzburg.de. After uploading a photo of a leaf the segmentation is performed either automatically or user-assisted. Then the user can choose how many of the best fitting species should be calculated. The classification process is executed and the results are shown. Finally, the user can manually compare the species to find out, which one fits best. An example classification result can be seen in Figure 12.

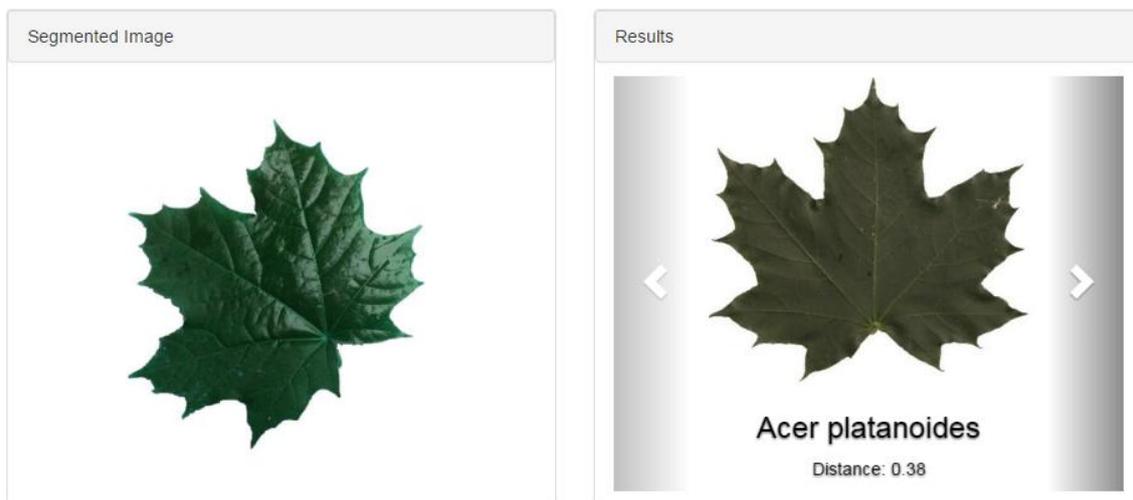


Figure 12. Example classification of a leaf using the web application.

Currently, the classification of a total of 430 species is supported. Of course, deploying the system as a mobile application would offer even more flexibility, but at the same time would make the precise manual segmentation almost impossible. To find the optimal solution will be a challenging task for the future.

10. CONCLUSION

A system for leaf identification was introduced and assessed using standard evaluation procedures. The achieved results came close to the state of the art. Further tests showed that the obtained classification accuracies could not be replicated at all when performing cross dataset evaluation. Especially, the performance of some individual feature classes deteriorated massively. The main reason for this are differing environmental influences depending on the area and time of growth of the respective leaves. Moreover, it was clearly shown that factors like rainfall, temperature and solar irradiance do not only influence the colour, but also the shape of the leaves. This leads to the conclusion that the standard procedures for evaluating and comparing leaf recognition systems can offer misleading results. That became especially evident by evaluating the performance of the colour features and the simplified version of the BP features, as they performed very well on the Flavia dataset, but failed completely during cross dataset classification. The HOCS features proved to be the by far best suited approach for this classification task yielding excellent results in both evaluation scenarios.

ACKNOWLEDGEMENTS

The authors would like to thank Georg Dietrich and Björn Eyselein for developing the web application as well as Simone Bayer for her comments that greatly improved the manuscript.

REFERENCES

- [1] Gang Wu, S., Sheng Bao, F., You Xu, E., Wang, Y., Chang, Y. & Xiang, Q.: A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network. IEEE 7th International Symposium on Signal Processing and Information Technology; pp 11-16. 2007.
- [2] Kadir, A., Nugroho, L., Susanto, A. & Santosa, P.: Foliage Plant Retrieval Using Polar Fourier Transform, Color Moments and Vein Features. Signal & Image Processing: An International Journal; Vol. 2, No. 3. 2011.
- [3] Kadir, A., Nugroho, L., Susanto, A. & Santosa, P.: Performance Improvement of Leaf Identification System Using Principal Component Analysis. International Journal of Advanced Science and Technology; Vol. 44. 2012.
- [4] Kadir, A.: A Model of Plant Identification System Using GLCM, Lacunarity and Shen Features. Research Journal of Pharmaceutical, Biological and Chemical Sciences; Vol. 5, No. 2. 2014.
- [5] Novotný, P. & Suk, T.: Leaf Recognition of Woody Species in Central Europe, Biosystems Engineering; Vol. 115, No. 4, pp 444–452. 2013.
- [6] Sulc, M. & Matas, J.: Texture-Based Leaf Identification. Lecture Notes in Computer Science; Vol. 8928, pp 185-200.
- [7] Lee, K. & Hong, K.: An Implementation of Leaf Recognition System using Leaf Vein and Shape. International Journey of Bio-Science and Bio-Technology; Vol. 5, No. 2. 2013.
- [8] Wu, J. & Rehg, J. M.: Centrist: A Visual Descriptor for Scene Categorization. IEEE Transactions on Pattern Analysis and Machine Intelligence; Vol. 33, Issue 8, pp 1489–1501. 2011.
- [9] Le, T., Tran, D., Pham, N.: Kernel Descriptor Based Plant Leaf Identification. 4th International Conference on Image Processing Theory, Tools and Applications (IPTA); pp 1-5. 2014.
- [10] Kumar, N., Belhumeur, P., Biswas, A. Jacobs, D., Kress, J., Lopez, I. & Soares, V.: Leafsnap: A Computer Vision System for Automatic Plant Species Identification. Computer Vision – ECCV 2012, pp 502-516. 2012.
- [11] Zhao, Z., Ma, L., Cheung, Y., Wu, X., Tang, Y. & Chen, C.: ApLeaf: An Efficient Android-based Plant Leaf Identification System. Neurocomputing; Vol. 00, pp 1-11. 2014.
- [12] Intelligent Computing Laboratory. Chinese Academy of Science.
- [13] Söderkvist, O.: Computer Vision Classification of Leaves from Swedish Trees. Master thesis, Linköping University. 2001.
- [14] Boykov, Y. & Jolly, M.-P.: Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images. Proceedings IEEE International Conference on Computer Vision; Vol. 1, pp 105-112. 2001.
- [15] Rother, C., Kolmogorov, V. & Blake, A.: „GrabCut“ – Interactive Foreground Extraction using Iterated Graph Cuts. ACM Transactions on Graphics (TOG) 23 (3), pp 309-314. 2004.

- [16] Open Source Computer Vision Library: Homepage. URL: <http://opencv.org/>.
- [17] Otsu, N.: A Threshold Selection Method from Gray-level Histograms. IEEE Transactions on Systems, Man and Cybernetics; Vol 9, pp 62-66. 1979.
- [18] Hu, M.: Visual Pattern Recognition by Moment Invariants. IRE Transactions on Information Theory; Vol. 8, pp 179-187. 1962.
- [19] Ojala, T., Pietikainen, M. & Mäenpää, T.: Multiresolution Gray Scale and Rotation Invariant Texture Classification with Local Binary Patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence; Vol. 24, Issue 7, pp 971-987. 2002.
- [20] Data Mining and Machine Learning Tool: Homepage. URL: <http://www.cs.waikato.ac.nz/ml/weka/>.
- [21] Chang, C. & Lin, C.: LIBSVM: A Library for Support Vector Machines. ACM Transactions on Intelligent Systems and Technology (TIST); Vol. 2, Issue 3. 2011.
- [22] Chih-Wei, H., Chang, C. & Lin, C.: A Practical Guide to Support Vector Classification. Technical Report, Department of Computer Science, National Taiwan University. 2003.

Authors

M. Sc. Christian Reul received his M. Sc. in computer science in 2015 from the University of Würzburg where he currently works as a research assistant at the chair for Artificial Intelligence and Applied Computer Science. His research interests comprise computer vision and machine learning tasks like classification and segmentation, especially in medical applications.



M. Sc. Martin Toepfer received his M. Sc. in computer science in 2012 from the University of Würzburg where he currently works as a research assistant at the chair for Artificial Intelligence and Applied Computer Science. His interests comprise software engineering, data mining and machine learning tasks.



Prof. Dr. Frank Puppe holds the chair for Artificial Intelligence and Applied Informatics at the University of Würzburg. His research interests comprise all aspects of knowledge management systems and tutoring systems and their practical application in medical, technical and other domains including heuristic, model-based and case-based problem solving methods, knowledge acquisition, machine learning and intelligent user interfaces.

