



FAKULTÄT
FÜR INFORMATIK
Faculty of Informatics



Technical Report
CVL-TR-3

Automated identification of tree species from images of the bark, leaves or needles

Stefan Fiel and Robert Sablatnig

Computer Vision Lab
Institute of Computer Aided Automation
Vienna University of Technology

November 14, 2010

Abstract

In this thesis a method for the automated identification of tree species from images of leaves, needles and bark is presented. In contrast to the current state of the art for the identification from leaf images, no segmentation of the leaves is necessary. Furthermore the proposed method is able to handle damaged and overlapping leaves. This is done by extracting keypoints in the image which allow the computation of local descriptors. With the help of a trained set of descriptors, a histogram of the occurrence of these descriptors is generated which is then classified using a one-vs-all SVM.

For images of the bark the same method is used which allows to build a system which can automatically distinguish between images of bark and leaves without calculating new features.

Detected features within the needle images can be used for a classification. The needle endings are used to distinguish between fir and spruce needles.

The proposed method is evaluated on three datasets which were provided by “Österreichische Bundesforste AG” (“Austrian federal forests”). The first dataset consists of 134 images of the leaves of the 5 most common Austrian broad leaf trees, the second dataset containing 275 images of the needles of the 6 most common Austrian conifers. The last dataset is containing images of the 1183 bark of these 11 trees. The classification rate for the leaf dataset was 93.6% and the classification rate of the bark dataset was 69.7%.

Contents

1	Introduction	1
1.1	Motivation	2
1.2	Objective	3
1.3	Main contribution	4
1.4	Results	4
1.5	Structure	4
2	Related Work	6
2.1	Identification using books	6
2.2	Identification of leaf images	6
2.2.1	Methods with geometric and morphological features	7
2.2.2	Methods with contour features	12
2.3	Identification of bark images	14
3	Methodology	16
3.1	Identification of leaf images	16
3.1.1	Scale Invariant Feature Transform	17
3.1.2	Bag of Words	22
3.2	Identification of bark images	28
3.2.1	Scale Invariant Feature Transform	28
3.2.2	Gray Level Co-occurrence Matrix	29
3.2.3	Wavelet transform	29
3.3	Identification of needle images	31
4	Experiments and Results	34
4.1	Datasets	34
4.1.1	Leaf dataset	34
4.1.2	Bark dataset	35
4.1.3	Needles dataset	37
4.2	Distribution of the SIFT features	38
4.3	Experiments with experts	40
4.4	Experiments on the leaf dataset	44
4.5	Experiments on the bark dataset	47
4.6	Experiments on the needle dataset	50
4.7	Evaluation of the multi-class SVM	52

4.8 Evaluation of the results	53
5 Conclusion	54
Bibliography	56

Chapter 1

Introduction

Identification of tree species from photos of leaves, the bark or needles is a task which requires expertise. This expert knowledge can be expected from foresters and botanists. For people without this knowledge the identification of tree species is a difficult assignment since the difference between some tree species is small or information for the identification, like the shape of the leaf, or the color and haptics of the bark have been forgotten.

Within a project with the “Österreichische Bundesforste AG” (“Austrian federal forests”) people should be able to identify tree species using their mobile devices by photographing the leaves, bark, or needles of a tree and the identification is done automatically by the mobile device and additional information for this tree species is then displayed on the screen.

The program is designed to work for the 11 most common Austrian trees: 6 broadleaf trees, and 5 conifers. The following list is showing these trees, in braces are the German and Latin names :

- Ash (Esche, *Fraxinus excelsior*)
- Beech (Buche, *Fagus sylvatica*)
- Black pine (Schwarzkiefer, *Pinus nigra*)
- Fir (Fichte, *Picea abies*)
- Hornbeam (Hainbuche, *Carpinus betulus*)
- Larch (Lärche, *Larix decidua*)
- Mountain oak (Traubeneiche, *Quercus petraea*)
- Scots pine (Rotkiefer, *Pinus sylvestris*)
- Spruce (Tanne, *Abies*)
- Swiss stone pine (Zirbe, *Pinus cembra*)
- Sycamore maple (Bergahorn, *Acer pseudoplatanus*)

The images of the trees have been taken by employees and foresters of the “Österreichische Bundesforste AG”. They images were taken in autumn 2009 and spring 2010. Since some trees grow only in specific regions in Austria five people were asked to take the images. Short instructions were given to them:

- leaves have to be on a monochrome background and should cover the whole image
- bark images should contain a section of the bark of approximately the size of a a4 paper and this section should cover the whole images. Furthermore the selection should show a typical bark of the species, not showing knotholes and natural cover.
- a needle should not cover only some needles but the ending of a branch where some clusters of needles can be seen and it should be on a monochrome background

Some of the received images which are not fulfilling the instructions are shown in Figure 1.1. These images showed that users do not always follow guidelines which were given them when taking an image and we can not assume that e.g. each image has a monochrome background.



Figure 1.1: Images we received which do not fulfill the instructions.

1.1 Motivation

Identification of tree species is taught to pupils in the elementary and in the secondary school. The main focus is the classification of leaves by letting the pupils collect leaves and

then classify them by using the information of the shape. If this knowledge is forgotten or is not sufficient for an identification then books, like Godet [God07], are used to help peoples with the identification.

These books use diagnostic polytomous keys where the user has to specify the tree more precise with each step (see Sections 2.1). But still this type of identification takes several minutes, since it includes scrolling through the pages, and also knowledge of the vocabulary used in these books.

There also exist internet pages, like [Nix10], which assists users to identify tree species by providing forms following a diagnostic key. These diagnostic keys are similar to the ones in books, but since everyone can create such internet page the source of the knowledge can not be trusted.

In 2003 already 90% of the austrian young people (from the age of 10 to 24) owned a mobile device [Göt05], in the USA 83% of the 17 year olds owned a cell phone in 2009 [ea10]. The number has been increasing since 2004 from 45%. 69% of the adults in the USA have used data application on their mobile device in 2009 [Hor09] , this number increased from 58% in 2007. With this increasing number of mobile devices and with increasing capabilities of these devices, like resolution of the integrated cameras, more computing power, and availability of internet the task of automated classification of tree species can be taken over by these devices.

With the help of an augmented reality view additional information can be provided to the uses. This information can include distribution in Austria or in the world, the family of the tree, or information about the fruits.

The principal concept of this thesis is to develop a system for an automatic classification of tree species. Users should photograph a leaf, the bark, or needles of a tree with their mobile device and after the classification, done by the system, additional information for this specific tree species is displayed on the screen.

1.2 Objective

The objective is to develop a system for an automatic identification of tree species from photos of their leaves, bark, or needles. Since the photos are taken from the users themselves, the conditions can not be controlled. Only some instructions how to take the photo can be given to the user like the size of the image, the size of the object itself, or a monochrome background. Whether the users follow these guidelines exactly can not be controlled, Figure 1.1 showed images we received which do not fulfilled our instructions.

The color of the leaf, needles or bark can not be taken into account since in autumn the color of the leaves changes. Furthermore due to different lighting conditions during the moment of the photo taking and different specifications of the cameras the color can differ.

Additionally, since the photos are taken with a mobile device the system has to deal with image resolution which can vary from model to model. Also the anti shock system of cameras in mobile devices is not as advanced the ones in real cameras, so little details can blur.

1.3 Main contribution

The main contribution is to introduce a method for the automated identification of tree species from images of the leaves which avoids segmentation. If the segmentation is wrong, the leaf is damaged or overlapped by other leaves wrong features will be calculated and the classification is wrong. Avoiding the segmentation step will eliminate a preprocessing step where errors can occur.

Additionally the presented method for the identification of the leaves can also be used for the identification of bark images. It is not needed that the users have to chose whether they want to analyze images of a leaf or images of a bark. Thus, an additional step can be introduced to decide whether it is an image of a leaf or an image of a bark and for the classification of the tree species the particular database can be used without calculating new features.

Furthermore it is analyzed how to identify the tree species from images of needles.

1.4 Results

The proposed method is applied to a dataset of bark images and a dataset of leaf images provided by the “Österreichische Bundesforste AG“. Additional experiments with a forest ranger and a biologist has been made to get a reference result for the used dataset. Only a subset of the bark and needle dataset was presented to the experts. The forest ranger achieved a classification rate of 77.8% whereas the biologist had 56.6%. When the proposed method was applied on the same dataset a classification rate of 65.6% was achieved which is approximately between the rates of the experts.

Further experiments with the whole dataset were carried out and the best recognition rate was 69.7%. When applying the method on the leaf dataset the recognition rate was 93.6%. Expanding the dataset with images which do not contain leaves should evaluate if the system can detect these. This is done by introducing an additional threshold. The classification rate for the leaf dataset drop to 84.4%. Two third of the images showing no leaf were recognized and the misclassified leaves of the first experiment where also assigned to the “no category” class.

1.5 Structure

In Chapter 2 the state of the art for the identification of tree species is presented. The first part describes the automated identification from images of the leaves, the second part from images of the bark.

In Chapter 3 the proposed method for the automated identification is presented. For the identification from images of leaves local features are used which will then used to form a bag of words. The second part of the chapter presented three possibilities to identify the tree species from photos of the bark. The last part of the chapter deals with images of the needles.

Chapter 4 details experiments and results of the proposed method. First the datasets used are presented and results of experiments which has been done with experts are presented. Afterwards the spacial distribution of the local features on leave images are evaluated. Next, the proposed methods were applied to these datasets and the results were evaluated. The last parts of this chapter are the evaluation of the multi-class support vector machine and the evaluation of the results.

The last Chapter 5 concludes the thesis. Advantages and disadvantages of the proposed system are discussed and an outlook is given.

Chapter 2

Related Work

In this chapter an overview of the current state of the art is given. In Section 2.1 the procedure when identifying a tree species with the help of a book is described. Section 2.2 reviews the current methods for the identification of tree species from images of the leaves whereas Section 2.3 describes current methods for the identification from images of the bark. To the best knowledge of the author no work has been published about the identification of tree species using images of needles.

2.1 Identification using books

To identify plant species manually a book like Godet [God07] is used. These books contain a diagnostic key where the user has to make various decisions which describes the leaf better in each step. Figure 2.1 shows the first two steps of a simple diagnostic polytomous key for broadleaf trees. Polytomous key means that the user has to decide between more than two possibilities. The users have to follow this tree step by step. First they have to decide whether the leaf is complete or compound. If it is complete they have to determine if the leaf is lobed, entire or dentate. If the leaf is compound then they have to decide whether it is digitated, paripinnate or imparipinnate. These books consist of many such steps, in Godet [God07] the identification of a leaf requires up to 6 steps each with up to 4 possibilities to chose.

The process of the identification can take several minutes since it includes scrolling through the book since most of the decisions lead to another page. Also the users have to be familiar with the vocabulary or have to compare the leaf with the illustrations in the book. Problems can occur when the users are unable to make a decision e.g. only a leaf is available and it has to be decided whether the grow alternate or opposite on the branch.

Since the bark can not be described as easily as leaves or needles, the user has to scroll through the book and has to look for the corresponding bark.

2.2 Identification of leaf images

The current state of the art of the identification of tree species from photos of the leaves can be divided into two parts. The first part, which deals with geometric and morphological

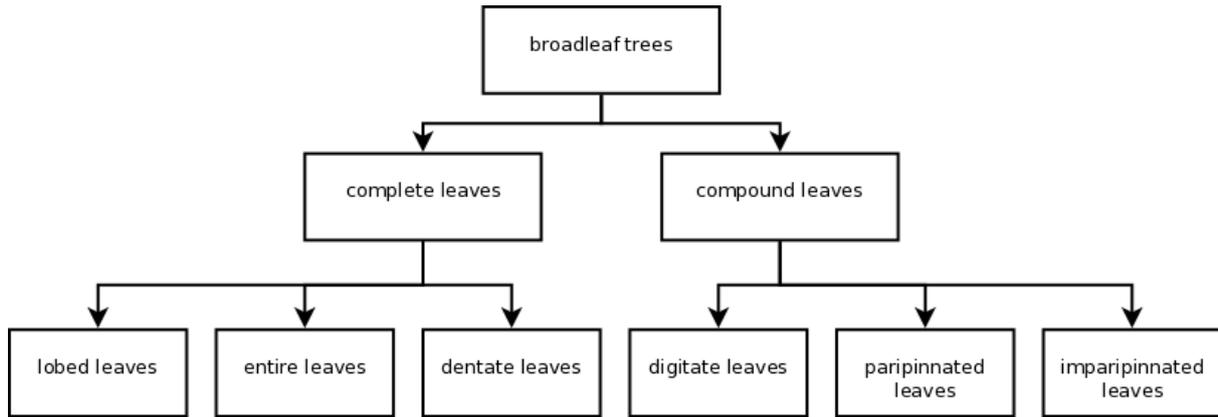


Figure 2.1: First steps of a simple diagnostic polytomous key to identify broadleaf trees from their leaves [God07].

features of the leaves will be described in Section 2.2.1. Section 2.2.2 describes the methods which are using the contour for the identification of the species. Methods which are using features of both approaches will be presented in the section which covers the main part. Due to the fact that every approach was tested on a dataset of different tree species from all over the world the results can not be compared.

2.2.1 Methods with geometric and morphological features

These methods are using geometric and morphological features for the identification. These features describe the properties of the different leaves. The geometric features are extracted directly from a binarized image whereas the morphological are derived from the geometric features. First a short overview of the geometric and morphological features found in the literature (see Table 2.1) will be given. The geometric features are illustrated in Figure 2.2. Afterwards the current state of the art methods will be described.

Geometric features are extracted directly from the binarized image I where $I(x, y) = 0$ is the background and $I(x, y) = 1$ is the leaf L . The border of the leaf ∂L is the subset of pixels of L where in the 8-connected neighborhood at least one pixel of the background occurs. The geometric features are:

Convex hull

The Convex Hull CH is the minimal convex polygon which contains the boundary of the leaf ∂L .

Convex hull area

The Convex Hull area $A(CH(L))$ is the area obtained by the convex hull.

Convex hull perimeter

The convex hull perimeter P_{CH} is the perimeter of the border of the convex hull.

	Circularity	Convexity perimeter	Eccentricity	Form factor	Rectangularity	Solidity	Sphericity
Ye et al. [YCL ⁺ 04]			x				
Wang et al. [WDZ05]	x	x	x	x	x	x	x
Wu et al. [WBX ⁺ 07]				x	x		
Du et al. [DWZ07]	x	x	x	x	x		x
de Zeeuw et al. [dZRP07]			x	x		x	
Lin and Peng [LP08]	x	x	x			x	x
Pauwels et al. [PdZR09]			x	x		x	

Table 2.1: An overview which of the methods are using which morphological feature sorted by the year of publication.

Leaf Area

The area of the leaf A is the sum of all pixels inside or on the boundary ∂L .

$$A(L) = \sum I(x, y)$$

Leaf Perimeter

The perimeter of the leaf P is the sum of all pixels on the boundary ∂L .

$$P = |\partial L|$$

Length

The length l is the longest distance between two points which are lying on the boundary of the leaf ∂L .

$$l = \max_{p, q \in \partial L} |p, q|^2$$

Radius of circumscribed circle

The radius of the circumscribed circle $R_{circumscribed}$ is the minimal radius of a circle that circumscribes the leaf.

Radius of inscribed circle

The radius of the inscribed circle $R_{inscribed}$ is the maximal radius of a circle that can be inscribed into the leaf.

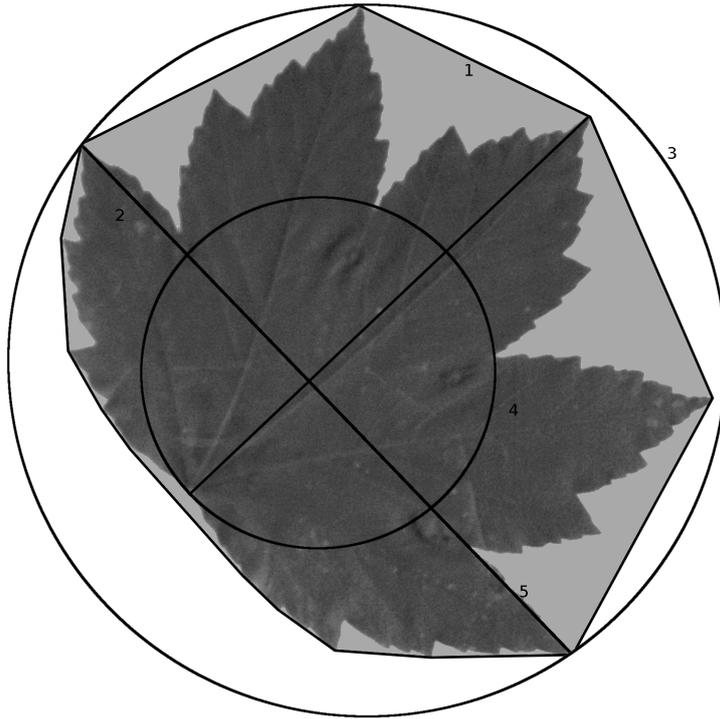


Figure 2.2: The geometric features of a sycamore maple leaf. 1: convex hull 2: length 3: circumscribed circle 4: inscribed circle 5: width. The gray region is the convex hull area.

Width

The width w is the longest distance between two points on the boundary of the leaf which lie orthogonal to l .

$$w = \max_{p,q \in \partial L, (p,q) \perp l} |p, q|$$

Since these features are not scale invariant they are used to calculate morphological features which achieve this property. Furthermore these features maintain the properties of the geometric features which are invariant against rotation and translation. These morphological features are:

Circularity

The circularity describes how similar the leaf is to a circle. The value is from 0 to 1, 1 meaning that the leaf is a circle.

$$C = \frac{4 * \pi * A}{P^2}$$

Convexity Perimeter

Is the ratio of the perimeter of the leaf and the perimeter of the convex hull. The value is 1 if the leaf is not lobbed.

$$CP = \frac{P_{leaf}}{P_{convexhull}}$$

Eccentricity

Eccentricity E is the ratio of the width and the length. It describes an ellipse with the identical second moment as the leaf. The value is from 0 to 1, where 1 is a perfect circle.

$$E = \frac{W}{L}$$

Form factor

The form factor describes the difference between the leaf and a circle. The value is from 0 to 1, where 1 is a perfect circle.

$$FF = \frac{4 * \pi * A_{leaf}}{P^2}$$

Rectangularity

The rectangularity describes how similar the leaf is to a rectangle. The value is from 0 to 1, where 1 is a rectangle.

$$R = \frac{A_{leaf}}{A_{boundingbox}}$$

Solidity

The solidity determine how lobbed the leaf is. It is the ration between the leaf area and the convex hull area. The value is from 0 to 1. 1 means that the leaf is not lobbed at all.

$$S = \frac{A(L)}{A(CH(L))}$$

Sphericity

Sphericity defines how round a leaf is. It is the ratio between the inscribed and circumscribed circle. It is 1 if the leaf is perfectly round.

$$Sp = \frac{R_{inscribed}}{R_{circumscribed}}$$

To calculate these geometric and morphological features it is necessary that the leaf has to be segmented. A wrong segmentation, which can occur due to shadows, overlapping leaves, or pinnate leaves will lead to wrong features. If a damaged leaf is segmented the features differ from those of a complete leaf. Also the stems have to be removed from the

image because they can cause differences in the geometric features since the stems can have a variable length or may not exist at all if a leaf has been found on the floor.

Ye et al. [YCL⁺04] proposed a web-based computerized plant species recognition system. First the leaves are preprocessed by rotating the leaves into the right position. Afterwards the image gets binarized. Then the direction and the length of the leaf stem is calculated. The features used are the eccentricity and the leaf apex and the leaf base angle. These angles are defined as the angle from the apex respectively the base to the points of a line perpendicular to the mid-vein at a quarter of the length of the lamina. For the matching the weighted distance for each of these 3 features are calculated and added up. The recall rate is 71.4% when the user has to choose the right result out of the five best matches. The database contains 130 leaves of a simple type.

Wang et al. [WDZ05] proposed a method of recognizing leaf images based on shape features. First the image is segmented by using an iterative threshold if the background of the image is simple or by using a marker-controlled watershed segmentation if the background is complicated. Afterwards the leafstems are removed to keep the precision of the shape features by performing an opening operation several times. So the leafstems can be removed successfully while preserving the main shape characteristics. Then the shape features, namely aspect ratio, rectangularity, solidity, convexity perimeter, sphericity, circularity, eccentricity, and form factor, are calculated. The aspect ratio is the ratio of the width and length of the bounding box of the leaf. Additionally the seven moment Invariants of Hu [Hu62] are used as features. The classification is done with a moving median center hypersphere classification which regards all data points of one class as a series of hyperspheres. These hyperspheres are then enlarged and the centers are moved to minimize the amount of hyperspheres. To classify a new data point the closest hypersphere is chosen. The average recognition rate is 92.2% on a dataset of 800 images of 20 classes.

In Wu et al. [WBX⁺07] the shape of the leaf is determined by transforming the image into grayscale and enhancing the contrast between the leaf and the background by weighting different color channels at the same time. These weights were gained by comparing the images of 3000 leaves [WBX⁺07]. This image is then transformed into a binary image and the shape of the leaf is gained by applying a Laplacian filter. Now the user has to mark the endpoints of the main leaf vein to get the physiological length and the physiological width. The features which are used for the leaf recognition are: smooth factor, ratio of the physiological length and width, form factor, rectangularity, ratio of physiological length and the length, the ratio between the perimeter and the length, the ratio between the perimeter and the sum of the physiological length and width and the vein features. The smooth factor is the ratio of the area of the image when smoothed by an averaging filter with different kernel size. The vein features are 5 features which can be calculated by performing a morphological opening on the grayscale image with different disk sizes. To reduce the input vector a principal component analysis is applied and the reduced vector is used to train a probabilistic neural network. The dataset used contains 1800 images with 32 classes. This approach has an accuracy of 90.3%.

Du et al. [DWZ07] also uses the moment invariants by Hu additional to eccentricity, rectangularity, solidity, convexity perimeter, sphericity, circularity, and form factor. The shape of the leaf is determined by using adaptive thresholding. With a moving median hypersphere classifier a classification rate of 91% is achieved. The dataset included 20

species of different plants. Each species includes 20 sample images.

De Zeeuw et al. [dZRP07] proposes a web-based tree taxonomy searchlabel by image query. They use a marker controlled watershed transformation to segment the leaves from the background. Afterwards the features for the shape are calculated. They are using solidity, form factor, eccentricity, and the moment invariants by Hu. The classification is done by apply a nearest neighbor classification. The success rate is approximately 85% when displaying the 10 best matches and let the user choose. The dataset used contains 146 images in total of 23 species.

Lin and Peng [LP08] proposes to use two leaf venation features besides the shape features. First the image is transfered to grayscale using different weights for each color channel. In this image the vein features are calculated. The first features is the nervation type which are labels for the different types of the distribution of the key vein and the side veins and the second features being the fractal dimension of the veins. The grayscale image is then converted to a binary image by thresholding. The shape features which are calculated are rectangularity, circularity, sphericity, eccentricity, axis ratio of the bounding box, solidity, and convexity perimeter. A probabilistic neural network is used as classifier and it achieves a classification rate of 98.3% on a dataset of 3000 images with 30 classes.

Pauwels et al.[PdZR09] proposes another web-based service for the automatic taxonomy of leaves. As shape features eccentricity, aspect ratio, elongation, solidity, stochastic convexity, form factor, maximal indentation depth, indentation spectrum, and lobedness are used. Elongation is the ratio of the radius of the largest inscribed circle to the smallest circumscribed circle. The stochastic convexity is the probability that a line-segment between to randomly chosen points is contained by the leaf. The maximial indentation depth is calculated by finding the maximal distance between the convex hull and the leaf contour normalized by the perimeter. For the indentation spectrum the Fourier spectrum of the indentation function is calculated and the smallest frequency at which the cumulated energy exceeds 80% of the total energy is taken and the lobedness is a feature which also describes the degree to which a leaf is lobbed. Additional to these shape features the moment invariants by Hu are used.

2.2.2 Methods with contour features

Abbasi et al. [AMK97] introduced a semi-automatic classification based on the Curvature Scale Space (CSS) image representation. The leaf is segmented from the background with adaptive thresholding and the boundary of the object is described using CSS which calculates the derivative of the tangent in the points of the boundary (see Figure 2.3). To get an equal amount of CSS representations for each boundary the boundary is re-sampled before calculating the derivatives. Now the maxima of the CSS representation are searched, to avoid noise in the representation only the maxima which are higher than 20% of the largest CSS contour are taken [AMK97]. When matching two CSS contours the first step is to look for possible orientation changes and then the Euclidean distance between the maximas is taken. Additional eccentricity, circularity, and aspect ratio of the CSS image are used as features. With a nearest neighbor classifier the correct answer of 95.25% input images has been among the 5 best varieties. The dataset contained 400 image.

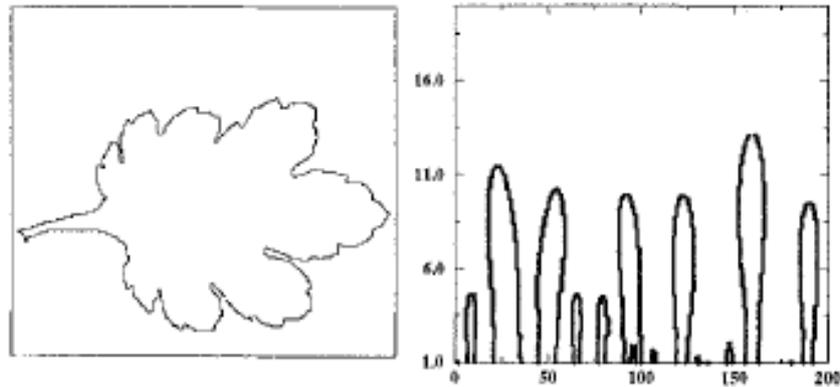


Figure 2.3: CSS representation of a leaf, taken from [AMK97]

Wang et al. [WCFW00] proposed a two step approach which uses the eccentricity and Centroid-Contour Distance (CCD) curve. The distance between the object centroid and a contour point is termed as the centroid-contour distance. Figure 2.4 shows a CCD representation of a leaf. This distance is already invariant to translation, to achieve invariance to scale the contour points are sampled and the distance is normalized. A thinning-based start point locating method, which takes the closest points of the contour to the centroid as possible start points, is proposed to make the CCD curve rotation invariant. The similarity measurement the euclidean distance between the centroid-contour distances for every possible start points is calculated and the minimum is taken. To make the leaf image retrieval more efficient the proposed to use the eccentricity as additional feature.

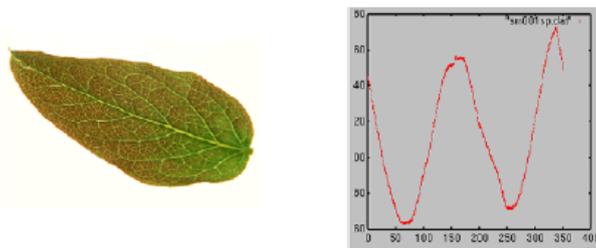


Figure 2.4: CCD representation of a leaf, taken from [WCFW00]

This method is extended in Wang et al. [WCF03] with an Angle Code Histogram (ACH). The ACH is introduced because the CCD may lose contour details such as ripples. The angle is computed for each point where two lines meet. These angles are than mapped with a code to a histogram (e.g. if the degree is from -5 to 5 degree the code is 0). The difference between two ACHs is sum of the differences of each bin of the histogram. With a nearest neighbor classification an average recall rate of 75.6% is achieved. The dataset contained 1400 leaves from 140 plants.

Liu et al. [LHWL09] used Maximum Numeric Sequences (MNS) to describe the shape of the leaves. The angle and the distance of the centroid to each contour point is calculated and normalized to gain scale invariance. The MNS representation has also the property of translation and rotation invariance. The matching is done with an modified KMP (Knuth-Morris-Pratt) fuzzy matching algorithm. The dataset contained 340 images of 92

different plant species. With an nearest neighbor classification a classification rate of 73% is reached.

Shape polygonal approximations and invariant attributes sequence representation are introduced by Du et al. [DWG05]. The shape of the leaf is approximated with polygons and it can then be represented as a sequence of vertices. With the invariant attributes sequence representation, which consists of four attributes (length, relative angle, area, and convexity) rotation, scale, and translation invariance can be achieved [DWG05]. The similarity of two leaves is calculated with a genetic algorithm for shape matching. Using a nearest neighbor classification the average recognition rate is up to 83.2%. In Du [Du06] the similarity is calculated with a modified dynamic programming algorithm and the average recognition rate is up to 92.3%. The dataset consisted of 2170 from 25 plant species.

Backes et al. [BCB09] introduced a small-work complex network to characterize shape boundaries, with which also tests with a leaves database, which contains 30 classes with 20 samples each, has been done. The success rate is 83.67%.

2.3 Identification of bark images

The automated identification of plant species from photos of the bark is done with a texture matching algorithm. Wan et al. [WDH⁺04] made a comparative study based on statistical texture features. They are using gray level run-length method, Gray Level Co-occurrence Matrices (GLCM) method, histogram method, and auto-correlation method. The gray level run-length method searches for the number how often a run of the same gray level occurs in the image in a given direction. From this occurrences five texture features can be calculated (short runs emphasis, long runs emphasis, gray level non-uniformity, and run percentage). The co-occurrence matrices method is based on the occurrence of gray levels with a defined spatial relationship (distance and angle between the two pixels). For each gray level 10 statistical measures can be extracted as texture features (entropy, angular second moment, contrast, inverse different moment, cluster tendency, cluster shade, correlation, maximum probability, and two correlation information measures). The histogram method uses the histogram of the image to generate 6 features. The auto-correlation method evaluates the linear spatial relationships between primitives. The best results were achieved with the co-occurrence matrices with an average recognition rate of 77% when using a nearest neighbor classification. The auto-correlation method had 72%, the run-length method 69%, and the histogram method 65%. To improve the classification rates they added the color information. Each of the three RGB channels were processed separately resulting in three times more features. The average recognition rate rises to 89% for the co-occurrence matrices, 84% for the run-length method, and 80% for the histogram method. The dataset used contained 160 preselected images of 9 classes.

Song et al. [SCLF04] proposed to use a combination of gray scale features, namely the GLCM, binary texture features, and the long connection length emphasis. With help of the wavelet transform the edge detection is done on which the binary features are calculated. For each foreground pixel, the pixel connection length is generated, which

is the maximum number of pixels which are connected to this pixel in a line at a given direction (0, 45, 90, and 135 degree). The long connection length emphasis is the normalized squared sum of these lengths for each direction. As co-occurrence features the energy and the angular second moment are used. The classification rate with a nearest neighbor classifier is 87.5% on a dataset of 180 images of 18 classes.

In Huang et al. [HZDW06] fractal dimension features are used additional to the GLCM features. The fractal dimension describe the complexity and self-similarity of texture at different scales. Homogeneity, contrast, correlation, energy, and entropy are used as GLCM features. For the classification a three layer artificial neural network was used. The dataset used contained 360 preselected images of 24 classes. The best recognition rate achieved was 91.67%.

Huang [Hua06] combined color and textural information for bark image recognition. Both information were extracted using the multiresolution wavelets. For the textural features the energy of the wavelet transformed images have been used. The color features where gained by transforming the color from RGB values to the YCbCr color space and calculating the energy at depth 3 of the wavelet pyramid for each channel. When using a radial basis probabilistic neural network an average recognition rate of 84.68% is achieved. The dataset consisted of 300 preselected bark images.

Chi et al. [CHC03] proposed to use Gabor filter banks for the recognition due to its efficiency and accuracy. They introduced multiple narrowband signals model to overcome problems with textures with a lot of maximas. The recognition performance for this approach is 96%. The dataset contained 8 classes of plants and each class containing 25 samples.

Chapter 3

Methodology

In this chapter the methodology is presented. First, in Section 3.1, the method for the automated identification of the leaf images is presented. Afterwards, in Section 2.3, methods for the identification of bark images are presented and Section 3.3 describes methods for needle images.

3.1 Identification of leaf images

All the methods for the automated identification of tree species from images of the leaves presented in Section 2.2 have in common that the leaf has to be segmented before calculating the features. A wrong segmentation which can occur due to shadows, or overlapping leaves will lead to wrong features. Also damaged leaves will lead to wrong features. The assumption that the leaf is always in the middle of the image for a better segmentation, e.g. when using watershed with markers, can lead to problems when dealing with multiple leaves in one image or with pinnate leaves. Furthermore since the users are making the photo the conditions when making the photo can not be controlled. It can not be assumed that the background of the leaf is always monochrome which would improve the result of a segmentation. Therefore a method for the automated identification of the tree species from images will be presented which avoids a segmentation step. To achieve this the use of the Scale Invariant Feature Transform (SIFT), introduced by Lowe [Low99], is proposed.

Figure 3.1 illustrates the workflow of the proposed method. As a preprocessing step the image is transformed to a gray scale image and is then normalized (Figure 3.1 a) to b)). On this image keypoints are found and the neighborhood is described (Figure 3.1 b)). With this description of the neighborhood a “bag of words” method by searching the nearest cluster center in order to form a histogram (Figure 3.1 c)). The x-axis of the histogram in Figure 3.1 c) represents the cluster center whereas the y-axis is the amount of the nearest local features. The cluster centers are generated from the trainings set for each class. This histogram is used to generate a feature vector for each image. The classification of the images is then done by using a Support Vector Machine (SVM).

This section describes the scale invariant feature transform in Section 3.1.1. Afterwards the bag of words method is described in Section 3.1.2.

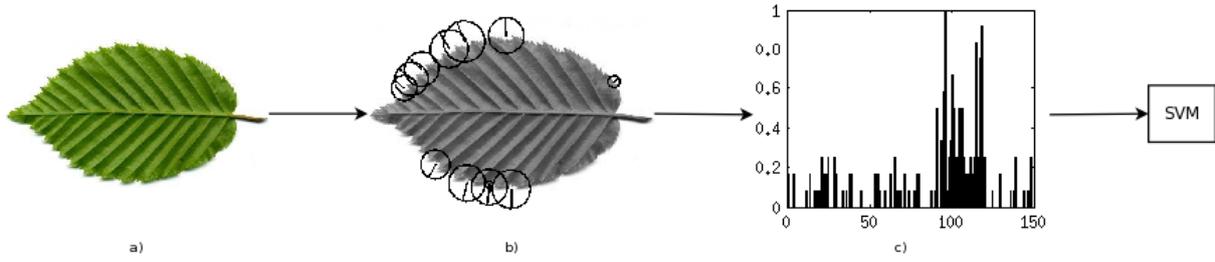


Figure 3.1: Workflow of the proposed methodology: a) Input image of a hornbeam b) Local Features (location and scale from 12 of 200 features are shown) c) Normalized histogram after searching the nearest cluster center.

3.1.1 Scale Invariant Feature Transform

The SIFT was proposed by Lowe [Low99] and improved in [Low04]. It was introduced in the field of object recognition by identifying objects using local features. These features are invariant to image translation, scaling, and rotation. Additionally they are partially invariant to illumination changes and affine or 3D projection [Low99].

To generate the set of image features 4 major stages of computation are used [Low04]:

1. Scale-space extrema detection
2. Keypoint localization
3. Orientation assignment
4. Keypoint descriptor

The first step is used to identify possible interest points over all scales and image locations. The second step selects the keypoints based on measures of stability followed by the assignment of the orientation for each keypoint. The last step is the transformation of these keypoints into a fixed representation. This section will explain the different steps in more detail.

Scale-space extrema detection

The scale space allows to extract structures of an image at different scales. The fine scale details are successively suppressed. Lindeberg [Lin94] showed that “the Gaussian kernel and its derivatives are singled out as the only possible smoothing kernels” [Lin94]. The scale space of an image is defined as a function $L(x, y, \sigma)$ which is calculated from the convolution of the input image $I(x, y)$ with Gaussians $G(x, y, \sigma)$ having variable scale:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

where $*$ denotes a convolution in x and y direction, σ being the scale parameter, and

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}.$$

To detect stable keypoints in the scale space efficiently Lowe [Low99] proposed to use the difference-of-Gaussian (DoG) function convolved with the image, $D(x, y, \sigma)$. It can be calculated by taking the difference of two nearby scales separated by a constant multiplicative factor k :

$$\begin{aligned} D(x, y, \sigma) &= ((G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y)) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned}$$

This function has the advantage that it is computational efficient since the smoothed images L has to be computed in any case for scale space feature description. Hence D can be computed by a simple image subtraction. Additionally this function is an approximation to the scale-normalized Laplacian of Gaussian, $\sigma^2 \nabla^2 G$. Mikolajczyk [Mik02] showed in experiments that the maxima and minima of $\sigma^2 \nabla^2 G$ produce the most stable features. In Lindeberg [Lin94] it is shown that the normalization of the Laplacian with the factor σ^2 is required for true scale invariance.

Figure 3.2 shows the pyramid representation of a Gaussian scale-space and its corresponding DoG scale-space. On the left column the initial image is incrementally convolved with a Gaussian. After doubling of σ the image gets resampled to speed up the computation time. The resampling is done by taking every second pixel in each row and column. The images of the same sample rate are called octave. Neighboring images are used to calculate the DoG images, which are shown in the right column, by image subtraction. Lowe [Low04] uses three scale samples per octave. Using more scale samples would lead to more extrema, but these are on average less stable. Additionally the cost of computation raises when using more scale samples.

The detection of the local extrema in the scale-space $D(x, y, \sigma)$ each sample point is done by comparing each sample point to its eight neighbors in the current image and its nine neighbors in the image of the scale above or below. The sample point is only selected if it is larger or smaller than all of its 26 neighbors.

To gain more stable keypoints Lowe [Low04] proposed to expand the input image to create more sample points. This avoids that the highest spatial frequencies are discarded when the image is pre-smoothed before the detection of the extrema. He proposed to double the image which increases the number of stable keypoints by almost a factor of 4.

Keypoint localization

In the previous section the keypoint candidates were found by comparing a pixel to its neighbors. The next step is to decide whether to reject or to take the pixel as keypoints. Pixels with low contrast or which are poorly localized along an edge are skipped.

The initial implementation of Lowe [Low99] proposed simply to use the located keypoints at the location and scale of the central sample point. Brown and Lowe [BL02] developed a method to determine the interpolated location of the maximum. In experiments it is shown that this method improves the stability. The interpolation spatially and in scale is done with search for the maximum of a Taylor expansion of the scale-space function. The function value in the extremum can be used to reject unstable keypoints with low contrast.

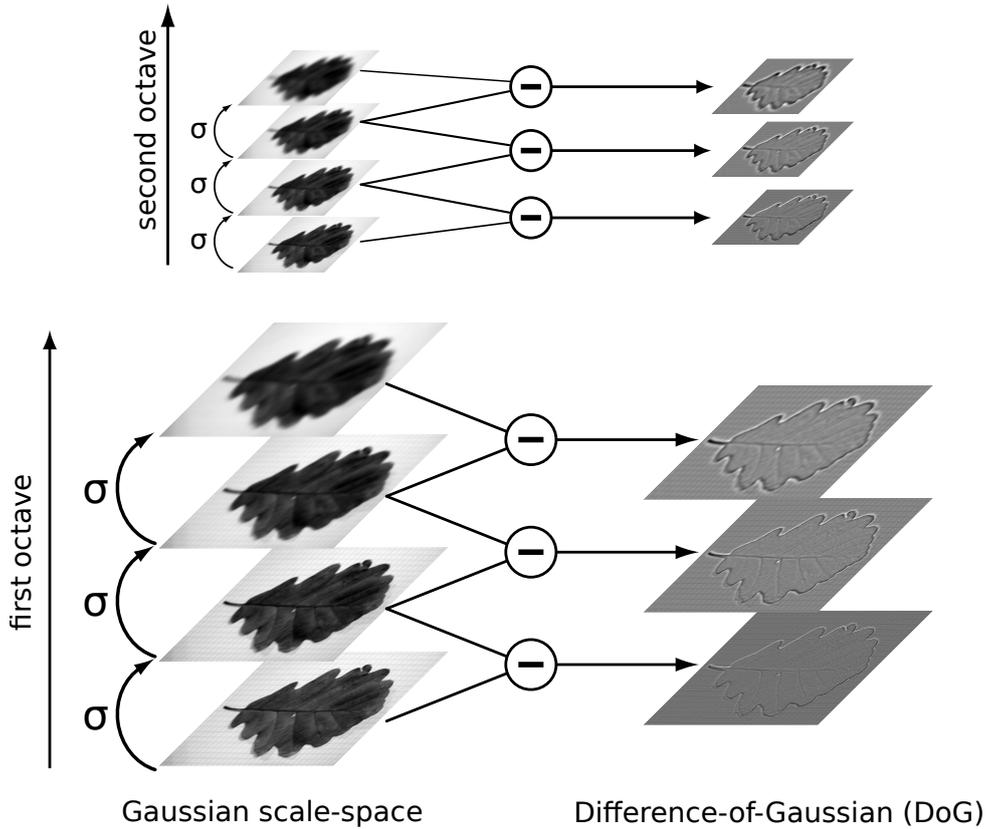


Figure 3.2: The pyramid representation of the scale-space. On the left side two octaves of Gaussian images with an increasing σ . On the right side the corresponding DoG images where edges and corners are enhanced and flat regions are gray.

The function value at the extremum is useful for rejecting unstable points with low contrast, so a threshold is introduced.

Additional the keypoints which are poorly localized along edges has to be rejected. These keypoints have been taken since the DoG function has strong response along edges, even if the location along the edge is poorly determined. To detect these points a 2×2 Hessian matrix, \mathbf{H} , is computed at their location and scale. The Hessian Matrix is defined as follows:

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

where D_{ij} denotes the second partial derivatives by x and y respectively. The derivatives are estimated by taking differences of neighboring sample points. The idea is to determine whether the principal curvature is large across the edge and is small in the perpendicular direction, which is characteristic for poorly defined peaks in the DoG function. Lowe [Low04] avoided to calculate the eigenvalues because only their ratio is important. A threshold r is introduced where it is only needed to check:

$$\frac{Tr(\mathbf{H})^2}{Det(\mathbf{H})} < \frac{(r+1)^2}{r}$$

where $Det(\mathbf{H})$ is the determinate of the Hessian matrix and $Tr(\mathbf{H})$ is defined as:

$$Tr(\mathbf{H}) = D_{xx} + D_{yy}$$

The threshold proposed by Lowe for r is 10.

Orientation assignment

To achieve rotation invariance the keypoint descriptor can be represented relative to the orientation of the keypoint. Thus, for every keypoint the main orientation has to be calculated. These are calculated on the Gaussian smoothed image, $L(x, y)$, which has the closest scale to the particular keypoint to maintain scale invariance. For the sample points within a region around the keypoint the gradient magnitude, $m(x, y)$, and orientation, $\theta(x, y)$ is calculated using pixel differences [Low04]:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$

$$\theta(x, y) = \tan^{-1}((L(x, y+1) - L(x, y-1)) / (L(x+1, y) - L(x-1, y)))$$

The orientations $\theta(x, y)$ are used to form an orientation histogram which consists of 36 bins, covering the 360 degree range of orientations. Every orientation added is weighted by its magnitude $m(x, y)$. To increase the descriptor robustness with respect to affine distortions and small variations of poorly localized keypoints, they are also weighted by a Gaussian window having a σ which is 1.5 times that of the scale of the keypoint.

The highest bin in the histogram indicates the dominant orientation of the given region. When another peak is within 80% of the highest it is used to create an additional keypoint with this orientation. Thus, multiple keypoints can exist at one location and scale but each of them having a different orientation. To accurately map the dominant orientation a parabola is fit to the 3 histogram values closest to the highest peak.

Figure 3.3 shows an image of a mountain oak with 6 keypoints. The scale of the keypoint is the radius of the circles whereas the lines are the main orientation of these keypoints.

Keypoint descriptor

The previous sections described how to find keypoints in an image and assign them location, scale, and orientation. Furthermore a 2D coordinate system has been created in which the local image region can be described and which is invariant to these parameters. The keypoint descriptor is generated with the magnitude $m(x, y)$ and the orientation $\theta(x, y)$ of the neighborhood of a keypoint at a given scale.

First the coordinate system of a local region is rotated in the orientation of the keypoint. To reduce the effect that changes of the gradients at the border of the region have a big effect to the descriptor the gradients are weighted with a Gaussian function which has a σ that is one half of the scale of the keypoint. Then the gradient magnitudes of a local region are trilinearly interpolated in order to avoid boundary effects. Finally the feature vector is normalized to unit length to gain invariance to affine illumination change. By thresholding the large gradient magnitudes and a renormalization the feature vector also

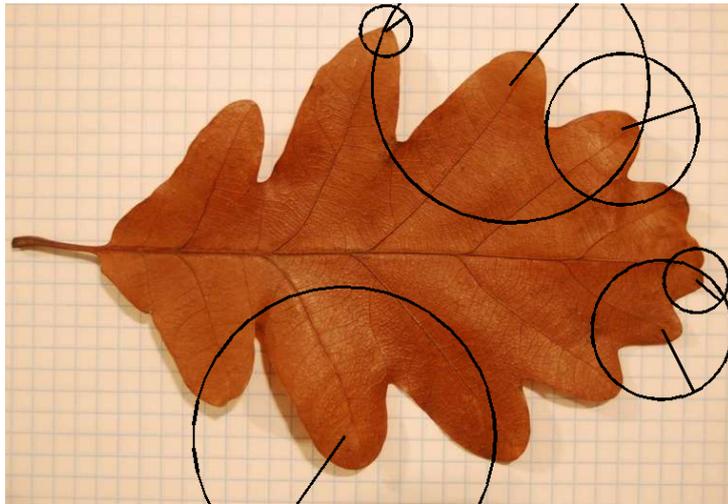


Figure 3.3: Six keypoints with their scale and orientation of a mountain oak leaf.

is invariant to non-linear changes of the illumination. Global changes in brightness does not alter the gradient values since they are computed from pixel differences.

Each descriptor consists of 4×4 orientation histograms with 8 bins, leading to a 128 feature vector for each keypoint. The 8 bins of the orientation histogram correspond to the gradient orientations ($0^\circ, 45^\circ, 90^\circ, \dots, 315^\circ$). Figure 3.4 illustrates how a 2×2 keypoint descriptor is created. On the left side are the gradient magnitudes and orientations of 8×8 sample array. These magnitudes are then weighted by a Gaussian function windows, which is indicated by the circle. For each subregion an orientation histogram is accumulated which is shown on the right side.

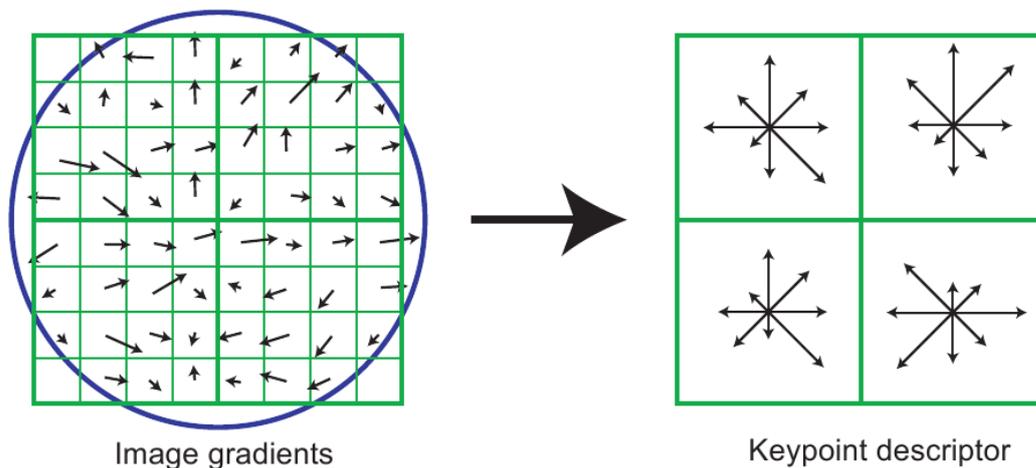


Figure 3.4: The construction of a 2×2 keypoint descriptor (taken from [Low04]). The left side shows a 8×8 sample array with the magnitudes and orientations of the gradients. The magnitudes are the weighted by a Gaussian window (circle) and then accumulated to a orientation histogram with 8 bins (right side).

3.1.2 Bag of Words

This section describes the bag of words method which is used to represent the image. The bag of words approach was introduced in the field of natural language processing and text classification [Joa98]. First the use of the bag of words approach in natural language processing is described, afterwards it will be explained how the bag of words model is used in computer vision. Subsequently the clustering of the SIFT features, which is necessary to form the bag of words, and how the feature vectors are presented. The last part is a description of the classification.

Bag of Words in natural language processing

In this model a text is considered as an unordered collection of words, regardless of grammar, or even word order. Words like articles, filler words, pronouns, determiners, and some verbs are skipped since they do not describe the content. To analyze texts, dictionaries have to be defined first which represent the classes. Then the occurrences of the dictionary words are counted in the text. The simplest way to classify a text will be to take the class of the dictionary with the highest occurrences. But if the classes do not differ very much it is possible that one word of the wrong dictionary appears often so that the wrong class has the highest percentage. So the idea is to build a histogram of the occurrences of the elements of the dictionaries and compare these histograms to others texts from all classes.

To show a simple example of text classification with a bag of words model the following short text should be classified:

This is a short text about leaf classification. The identification of tree species can be done using a computer. Often geometric features calculated from the shape of the leaf are used for the classification. Damaged leaves will lead to wrong features.

For this text it should be determined whether it is about automated leaf classification, automated text classification, or geometry. First dictionaries for all three classes have to be defined:

$$\begin{aligned} D_{leaf} &= \{ \text{"leaf"}, \text{"classification"}, \text{"computer"}, \text{"shape"}, \text{"identification"} \} \\ D_{text} &= \{ \text{"text"}, \text{"classification"}, \text{"computer"}, \text{"word"}, \text{"grammar"} \} \\ D_{geometry} &= \{ \text{"geometry"}, \text{"shape"}, \text{"triangle"}, \text{"rectangle"}, \text{"angle"} \} \end{aligned}$$

For simplicity reasons assume that each word also represents the plural and also the adjective. Note that “classification”, “computer”, and “shape” occur in two different dictionaries. The next step is to delete the articles, filler words, pronouns, and determiners from the text:

*Text leaf classification identification tree species computer geometric features
shape leaf classification damaged leaves wrong features*

We can now count the occurrences of each word in the dictionaries in the text. Some words are not part of any dictionary and will be skipped. Words of the dictionary D_{leaf}

appear 8 times in the text, words of D_{text} four times and words of $D_{geometry}$ occur twice. So the possibility that this text is about the classification of leaves is the highest with 57,1%. A normalized histogram of these occurrences in the text is shown in Figure 3.5.

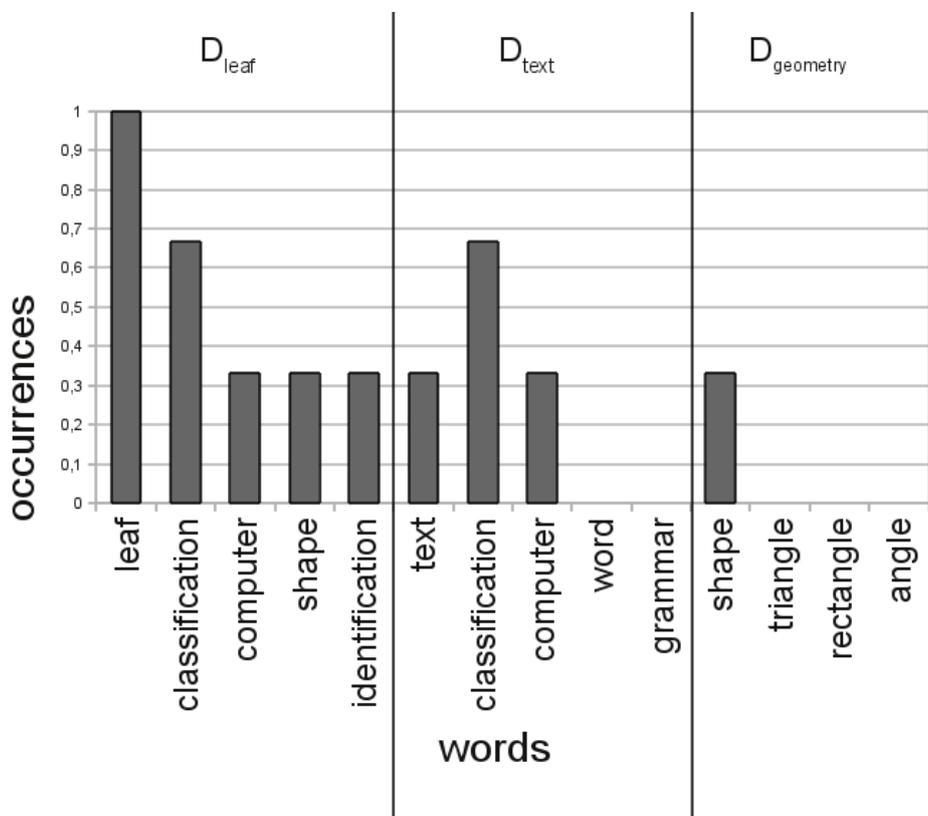


Figure 3.5: The normalized histogram of the occurrences of the words in the dictionaries in the text.

Bag of visual Words

When using the bag of words model in computer vision, image patterns take over the place of the words in the dictionary. These patterns do not have an actual meaning like “tree” or “leaf”. Csurka et al. [CDF⁺04] proposed to cluster the SIFT features to generate a dictionary of visual features from which a histogram of its appearances in an image can be generated.

The first step is to find the image patches in the set of training images. This is done as described in Section 3.1.1. All SIFT features from all the images of the trainings set are then clustered. This is done to reduce the computation time, because one naive approach would be to compare each feature vector from the test set with every feature vector of the trainings set. Csurka et al. [CDF⁺04] are using k-means for clustering and it is also used in the proposed method. So the training can be outlined in following steps:

1. Generate all SIFT features for the trainings set
2. For each class cluster the feature vectors

3. Generate the feature vectors for the trainings set
4. Train the classifier with the feature vectors

The cluster step can be skipped when classifying a new image since the old cluster centers are used to generate the feature vector. Since the cluster centers are calculated for the trainings set and never changes online learning is not possible with this method. For classifying a new image following steps have to be done:

1. Generate the SIFT features for the image
2. Calculate the feature vector for the image with the cluster centers from the trainings set
3. Classify the feature vector

The clustering of the trainings set can be seen as analogy to the vocabularies in the bag of words model for natural languages. The dictionaries are the cluster centers. The appearance of a vocabulary in the text can be considered as finding the nearest cluster center of a SIFT feature.

k-means clustering

Clustering divides points, which all have the same dimension, into groups by minimizing a given function. This is done by successively minimizing a distance function by assigning a point to another group. K-means, which was first mentioned by Lloyd [Llo82], uses as distance function the sum of all square-errors of the point and the cluster center within a cluster. The cluster center is defined as the mean of all points which are in this cluster. The distance function d of k-means is defined as:

$$d = \sum_{i=1}^k \sum_{x_j \in S_i} \|x_j - \mu_i\|^2$$

where k is the number of clusters, the set S are the points belonging to a cluster, and μ_i the mean of points in S_i . Because the minimization of this function is NP-hard, the k-mean method does not guarantee to find the perfect solution. The starting points of the cluster center are chosen randomly. Because no cluster center can be dropped or introduced the amount of cluster centers has to be predefined. The heuristic algorithm consists of two steps which are repeated consecutively until convergence or a maximum number of iterations is reached:

1. Initialize the centers μ_i for $i = 1 \dots k$,
2. Assign S_i all samples that are closer to μ_i than to μ_j for $j \neq i$
3. Recalculate the cluster centers μ_i
4. Repeat step 2 and 3 until convergence or a maximal number of iterations is reached

Figure 3.6 shows a demonstration of the k-means algorithm. The dots are data points, the filled squares are cluster centers whereas the white squares are the new centers. First randomly three data points are chosen as start points for the clustering, the data points are assigned to the nearest cluster center, and then the new centers are calculated. The data points are assigned again to the new centers and these are recalculated. This steps are repeated until convergence.

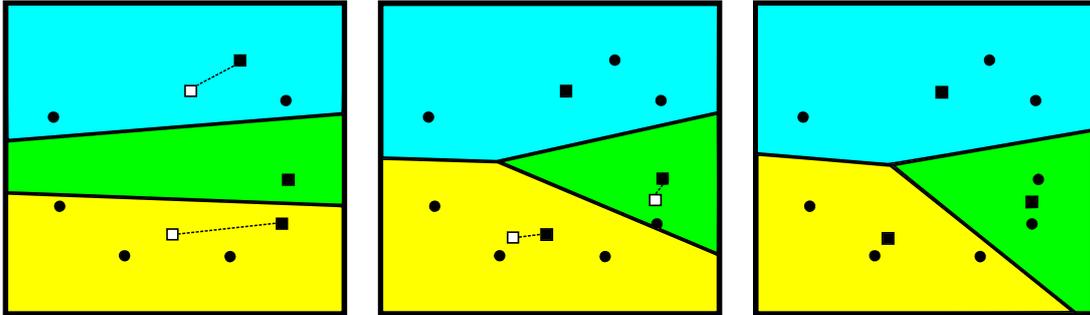


Figure 3.6: Demonstration of the k-means algorithm. Dots are data points, black squares are old cluster centers whereas white squares are new cluster centers. Random data points are chosen as starting points, all data points are assigned to the nearest cluster center and the new center is calculated (left). The clusters are visualized using a Voronoi diagram. The data points are assigned to the new cluster centers and the centers are calculated again (middle). The algorithm converged, every data point is assigned to the nearest cluster center and they are no longer moving (right).

Feature vector generation

The feature vector is generated by searching the nearest cluster center and generating a histogram of the occurrences of these centers. For each SIFT feature f the cluster S has to be found which has the nearest cluster center μ :

$$S = \min_{i=1\dots k} (||f - \mu_i||)$$

After generating the histogram it has to be normalized. Figure 3.7 shows a normalized histogram of a hornbeam image. It can clearly be seen that the cluster centers that are generated from the hornbeam images in the trainings set are from 90 to 120, since the peaks are within this interval. Only some features have a nearest cluster center which is not generated by the hornbeam trainings class. The reason for this is that some features from the hornbeam class are similar to ones from other classes or that features have been found in the test image which are not represented by the cluster centers of the hornbeam class. As long as there are only a minority of these features they are not influencing the classification.

Classification

The classification can be done using various classifier like nearest neighbor, neural networks, or support vector machines. Csurka et al. [CDF⁺04] made experiments with a

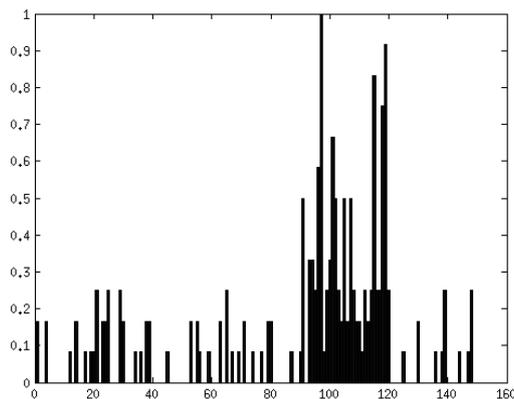


Figure 3.7: Normalized histogram of a hornbeam image.

naive Bayes classifier and a support vector machine (SVM). The SVM outperforms the naive Bayes classifier. So the proposed method also uses a SVM for classification, which was introduced by Vapnik and Chervonenkis [VC74], because it considers the difference between the empirical risk and the true overall risk.

Support Vector Machine

The SVM classifier was introduced to classify two classes which are linear separable. The goal is to find a hyperplane which is dividing the two classes and maximizes the margin. The margin is the distance from the nearest data point to the hyperplane. The nearest data points are called support vectors. Figure 3.8 shows linear separable classes and a hypersphere with maximal margin which divides the two classes. Only the feature vectors which are nearest to the hyperplane are used as support vectors.

The margin is defined as $1/||w||$ and should be maximized to get the best hyperplane. This leads to a dual optimization problem which can be solved by means of Lagrange multipliers. For the optimization problem only a small subset of the input data has to be considered, namely the support vectors.

The SVM as described above has been extended by Boser et al. [BGV92] to a non-linear classifier. Instead of the dot product in the criterion function non-linear kernels functions are used. The hyperplane is no longer calculated in the feature space, but in a space of higher dimension. The feature space is transformed in this high-dimensional space and there a linear hyperplane can be calculated. This hyperplane is then transformed back into the feature plan. There it does not have to be linear or even linked. By means of the kernel trick, which was first published by Aizerman [ABR64], this can be done by even not calculating the higher dimensional space explicitly. This is desirable since the high-dimensional space can be infinite. Here the radial basis function (RBF) kernel is used which is defined as:

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

where x_i and x_j are the features vectors and γ being a parameter which has to be deter-

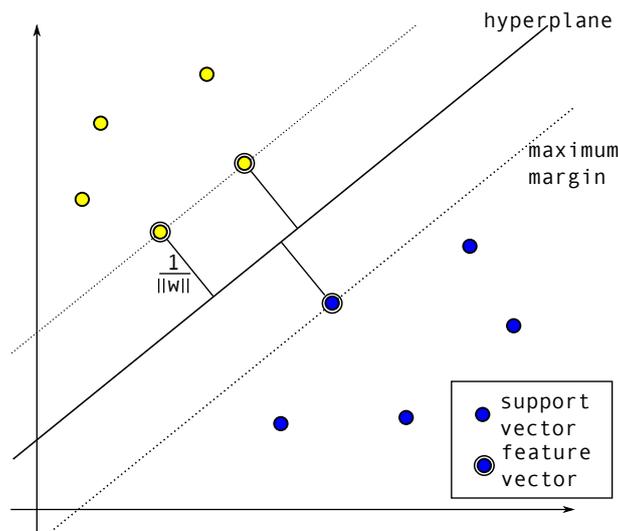


Figure 3.8: Linear separable classes which are divided by a hyperplane (black line) and the maximal margin (dotted line).

mined empirically.

Multi-class SVM

As described above the SVM is a classifier which can only handle two classes. To overcome this, two approaches have been introduced, both reduce the multi-class problem into multiple binary classification problems. These two approaches are:

- one-vs-one approach
- one-vs-all approach

The one-vs-one method generates multiple SVMs which classify between every pair of class. So for M classes $\frac{M*(M-1)}{2}$ SVMs are generated. The classification is done by a max-wins voting strategy. Every SVM increases the vote of one class depending on its output function. The class with the highest number of votes wins and is assigned to the multi-class SVM.

The one-vs-all method generates one SVM for each class. Each SVM being a classifier for the data points of one class against all other data points. The classification is done by a winner-takes-all strategy, meaning that the classifier with the highest output function is assigned the class.

Figure 3.9 shows how the different classes are classified to each other. Each square represents one class. The red respectively the blue squares are the classes which are examined by the SVM. The white squares are classes which are skipped. On the left side the first SVMs of the one-vs-one approach is illustrated where every class is combined with every other class. The right side shows the one-vs-all approach where a SVM is constructed with one class against all others.



Figure 3.9: Schematic diagram of a multi-class SVM. Pos respectively Neg are the two classes which are classified. On the right side the first 4 SVMs of the one-vs-one approach, the class with the most votes wins. On the left side the one-vs-all approach where every class is compared with the rest of the datasets. The one with the highest output function is taken.

In the proposed method the one-vs-all approach is used because each class has one result of the output function. Thus, a threshold can be introduced to eliminate image which are not part of a class. The one-vs-one approach is also evaluated in Section 4.7

3.2 Identification of bark images

Classification of the bark is done by using texture analysis methods. In Chen et al. [CPW00] texture is defined as repetitive patterns that occur in a region. The bark of trees does not have exact periodical and identical patterns due to natural growth. Natural cover of the bark, like moss and lichens, distort these patterns or the repetitive occurrence. Due to different lighting conditions the gray values of the patterns are changing and influence the recognition of the patterns.

One of the defining qualities of texture is the spatial distribution of gray values. This distribution can be described using statistical texture analysis methods like the GLCM. Other techniques rely on signal processing like wavelets. Section 3.2.1 describes the use of the SIFT features in texture analysis. In Section 3.2.2 the GLCM is described and in Section 3.2.3 wavelets are described.

3.2.1 Scale Invariant Feature Transform

Zhang et al. [ZMLS06] showed that SIFT features, which were already presented in Section 3.1.1, can keep up with common texture classification methods. The SIFT features are used to describe the texture of the region. The advantage of this method is that it does not rely on periodical patterns but on patterns which occur frequently in the image. With the bag of words approach these patterns do not have to be identical since the nearest cluster center is search which represents similar regions. So the method from Section 3.1 which is used for the identification of leave images is also applied for the automated identification of tree species from images of the bark.

3.2.2 Gray Level Co-occurrence Matrix

Haralick et al. [HSD73] proposed the GLCM for texture analysis. It describes the spatial distribution of the gray values in an image. Each co-occurrence matrix C contains the relative occurrences of combinations of two gray values i and j with a given spatial relationship. The number of gray values is G , thus C is of size $G \times G$. The spatial relationship is defined by two parameters, the orientation θ and the distance d .

The table on the left side of Figure 3.10 shows a small sample image with $G = 3$. The absolute values of the GLCM with the parameters $\theta = 0^\circ$ and $d = 1$ of this image is shown in the right table. It shows the occurrences of a pixel with the gray value which equals the row in C to a pixel with the gray value which equals the column in C in the given direction (here 0° , which means horizontally) in the given distance d .

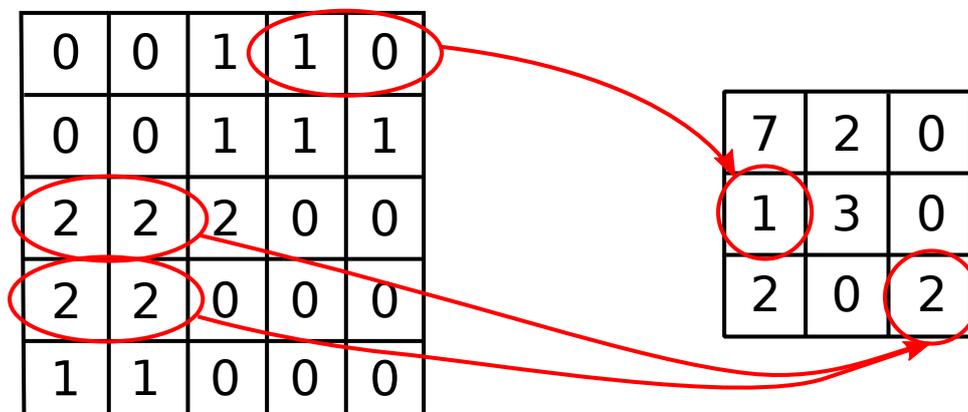


Figure 3.10: A small sample image with $G = 3$ on the left side and the corresponding GLCM with parameters $\theta = 0^\circ$ and $d = 1$ on the right side.

To get features for the classification statistical informations which are calculated from C are used. In this thesis these features are:

- **Contrast** This measures the local variations in the GLCM.
- **Correlation** This measures the joint probability occurrence of the specified pixel pairs.
- **Homogeneity** This is the sum of the squared elements in the GLCM.
- **Energy** This measures the closeness of the distribution of the elements in the GLCM to its diagonal.

3.2.3 Wavelet transform

The wavelet transform was introduced to multi-resolution signal decomposition by Mallat [Mal89]. It allows decomposition of a signal using a series of elemental functions, called wavelets, which are created by scalings and translation of a base function. This base function is called mother wavelet. Thus, wavelets provide spatial and frequency information

at different scales. Figure 3.11 shows a Daubechies mother wavelet. According to Chang et al. [CK93] the decomposition can be seen as passing a signal through a pair of filters and downsampling the filtered signals by two. These pair of filters correspond the the halfband lowpass and highpass filters.

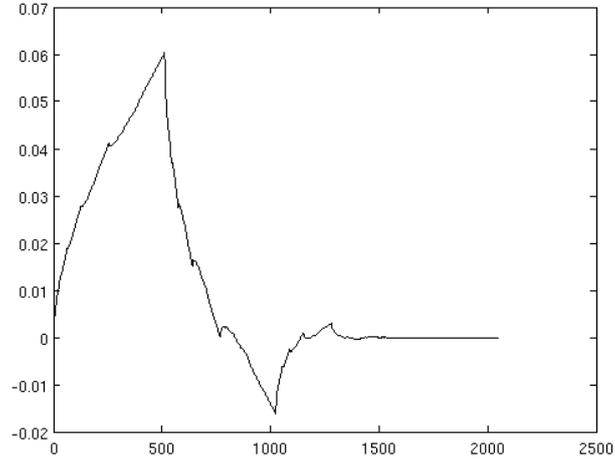


Figure 3.11: A Daubechies mother wavlet.

The 2-D wavelet basis function can be expressed by the product of two 1-D wavelet basis function, one in the horizontal and the other in the vertical direction. An image is composed into four sub-bands by applying the Discrete Wavelet Transform (DWT), see the left side of Figure 3.12. Where L and H denotes the lowpass and highpass filtering characteristics in the x - and y -directions. A wavelet packet can be build up recursively by decomposing subsignals in the low frequency domain. Figure 3.12 illustrates the first step, the $LL1$ subband is decomposed again and $LH2$, $HH2$, $HL2$, and $LL2$ are generated. This recursion is repeated until a given depth of the pyramid is achieved or if a threshold is reached.

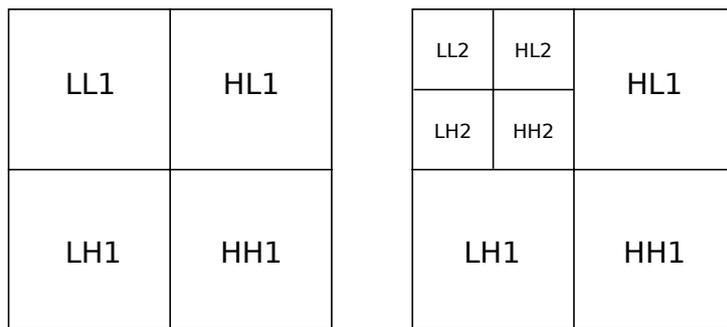


Figure 3.12: The decomposition into four sub-bands and the next step of the recursive decomposition to build up a wavelet packet.

A simple feature which can be extracted from the wavelet coefficients is the average energy of each detail in an image. The average energy is defined as the sum of squares normalized by the total number of coefficients in the image. More complex features are not used in this thesis since experiments have shown that they do not improve the results.

3.3 Identification of needle images

As already mentioned in Chapter 2, to the best knowledge of the author no published work has been found for the identification of tree species from images of their needles. The dataset contains 6 tree species which can be divided into two classes. The first class containing the trees on which one needle grow separately on the branch (see Figure 3.13) and the second class being containing the trees on which the needles grow in clusters on the branch (see Figure 3.16). Since the needles are too small the SIFT features which are used for the classification of leaves and bark can not be used. The keypoints which are found do not describe the needles. Rather they are located at the branches and the scale is so high that they do not have useful information for the classification of the tree species. So the two classes of the conifers will be treated separately.

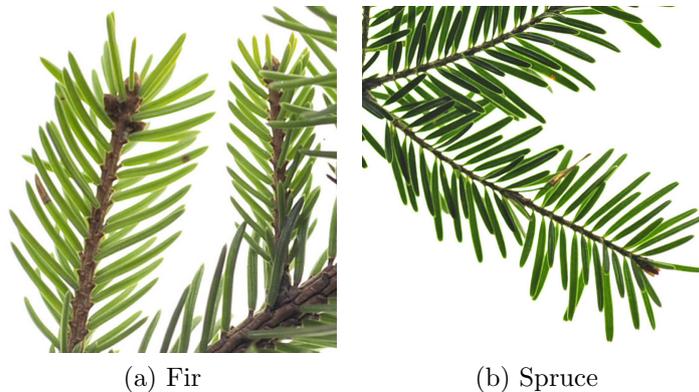


Figure 3.13: Images of fir and spruce on which the needles grow separate.

The easiest way to distinguish between a fir and a spruce needle is the backside of the needle. The spruce needle has two white stripes on it which do not exist on a fir needle. Since it can not be assumed that every image shows the backside of the needles, this characteristics can not be used. Another way to differentiate the fir and the spruce is by looking at the endings of the needles and the adjustment of the needles on the branch. The spruce needles are blunt and they grow only in one plane on the branch whereas the fir needles are pointed and they can grow in every direction. Since the image is taken by a user and the needles of the spruce can overlap the grow direction can not be used to distinguish those two species. To analyze the ending of the needles first the endpoints have to be found. This can be done using a segmentation algorithm, which separates the branch and the needles from the background, see Figure 3.14 (a). Afterwards the skeleton of the branch and the needles is calculated and so the endpoints can be found. This is illustrated in Figure 3.14 (b) and (c). The endpoints which are lying at the border of the image or which are lying too close to each other are skipped.

These endings can now be analyzed by calculating features like the eccentricity, solidity, curvature features, and the improved moment invariants by Chen [Che93] In Figure 3.15 it can be seen that the endings of the needles do not differ much, even on an image with a monochrome background and perfect lighting settings.

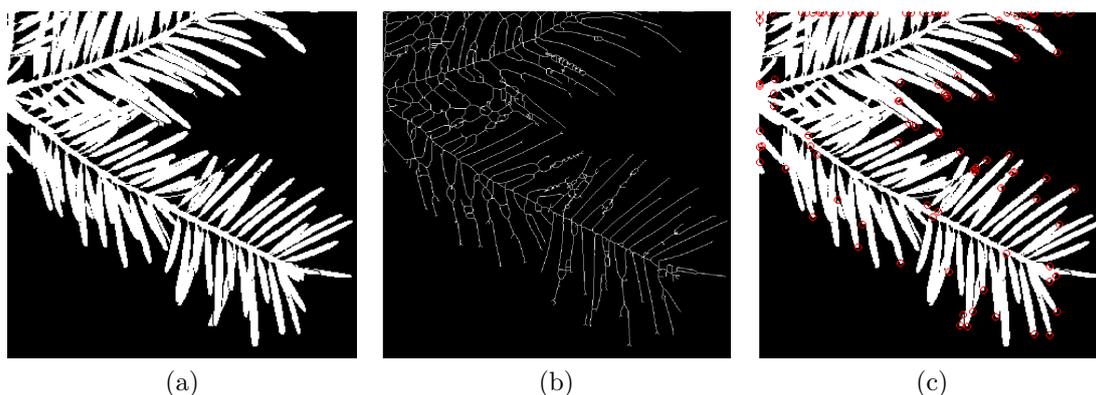


Figure 3.14: Finding the endings of the needles of an image of a spruce. The segmentation of the branch and the needles from the background is shown in (a). In the middle is the skeleton of the first image and with this skeleton the endpoints of the needles can be found (c).

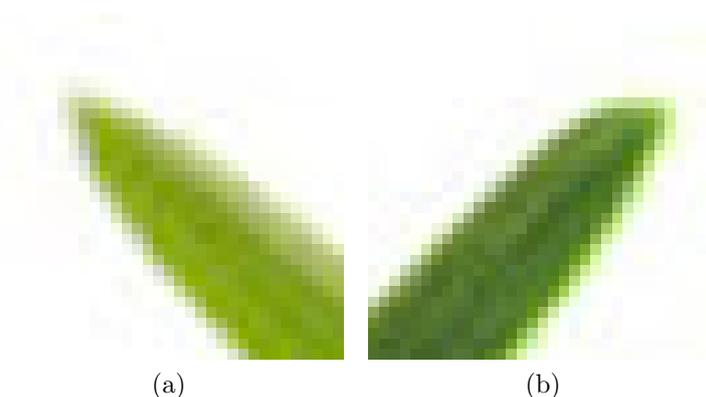


Figure 3.15: Needle endings of a Fir (a) and Spruce (b).

The conifers on which the needles grow in clusters on the branch the number of needles in the cluster is the most important characteristic. While the larch has a high number of needles in such a cluster (see Figure 3.16 (a)), the black pine and the scots pine the needles are always pairwise (see Figure 3.16 (b) and (c)), the swiss stone pine has usually five needles in each cluster.

With the same methods as described above the endings of the needles can be found. Figure 3.17 is showing the results of the segmentation, the skeleton and the endings of the needles. It can be seen that even though the endings have been found it is not possible to determine the number of needles of one cluster. The reason for this are overlapping needles, more than one cluster at nearly the same location, and the needles of one cluster are so close to each other that they can not be counted.



(a) Larch



(b) Black pine



(c) Scots pine

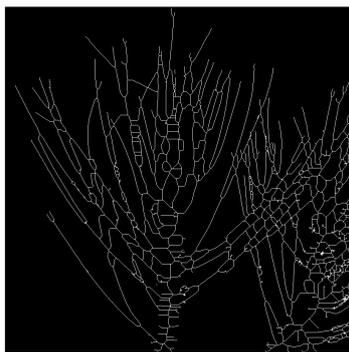


(d) Swiss stone pine

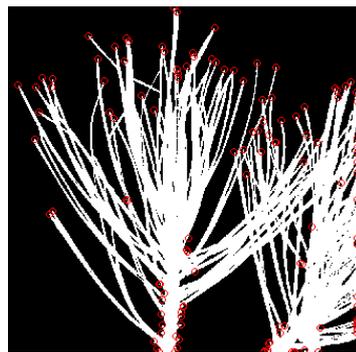
Figure 3.16: Images of the conifers on which the needles grow in clusters.



(a)



(b)



(c)

Figure 3.17: Finding the endings of the needles of an image of a spruce. The segmentation of the branch and the needles from the background is shown in (a). In the middle is the skeleton of the first image and with this skeleton the endpoints of the needles can be found (c).

Chapter 4

Experiments and Results

In this chapter the experiments and the results are presented. These experiments were done on a set of images which was provided by the “Österreichische Bundesforste AG”. This datasets are presented in the Section 4.1. Section 4.2 evaluates the distribution of the SIFT features on leaves. Afterwards the results of an experiment done with experts is presented in Section 4.3. This experts attended a forester and a botanist and can be considered as a kind of ground truth of experts to our datasets. In Section 4.4 the results of the experiments of the leaf dataset are presented. Section 4.5 deals with the results of the bark dataset. In Section 4.6 experiments on the needle dataset are presented and the last Section 4.7 evaluates the one-vs-one multi-class SVM.

4.1 Datasets

This section gives an overview of the used datasets. The first dataset are leaves of the most common Austrian broad leaf trees and is presented in Section 4.1.1. The second dataset, which is presented in Section 4.1.2, are the bark of the most common Austrian trees. The needles of the most common Austrian conifers are the third dataset which is presented in Section 4.1.3. These datasets were gathered by employees of the “Österreichische Bundesforste AG“ in autumn 2009 and spring 2010.

4.1.1 Leaf dataset

The leaf dataset consists of 134 images of five Austrian broad leaf trees which were scaled to either 800 pixel height or 600 pixel width. Every class has 25 to 34 images. Image of the leaves of these trees are shown in Figure 4.1, in braces are the amount of images for this class:

- a) Ash (25)
- b) Beech (30)
- c) Hornbeam (34)
- d) Mountain oak (22)

e) Sycamore maple (23)

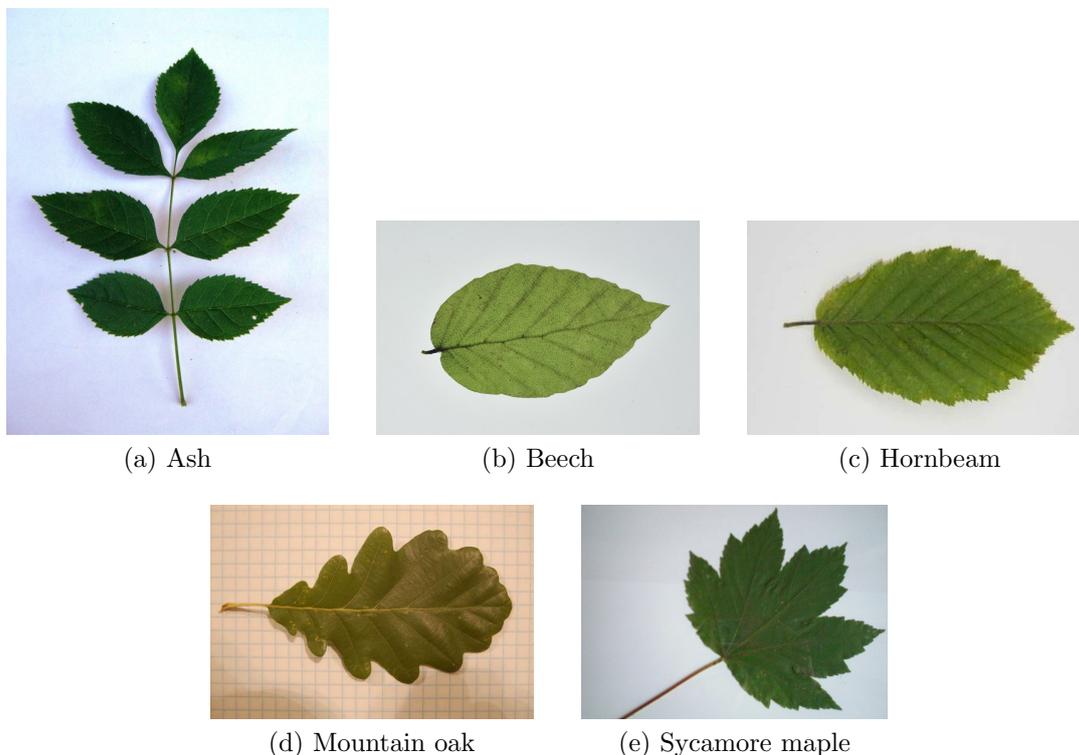


Figure 4.1: One sample images of a leaf of all tree species in the dataset.

While the beech, hornbeam, mountain oak and sycamore maple are complete leaves, the ash is compound, more precise a pinnate leaf. As mentioned in Chapter 3 this can lead to problems if images which contains a leaf of an ash will be segmented. The leaf does not have to be in the middle of the image, which can support segmentation algorithms like watershed with markers, and that there is not only one leaf in the image. Furthermore it can be seen in Figure 4.1 that a part of the leaf of the ash can be similar to a hornbeam leaf. Both leaves are toothed and can have approximately the same shape.

4.1.2 Bark dataset

The dataset of bark images contains 1183 images of eleven Austrian trees. Every class has 16 to 213 images. These images were also scaled to a size of either 800 pixel height or 600 pixel width. An image of the bark from every tree species in the bark dataset can be seen in Figure 4.2, in braces is the amount of images for this class:

- a) Ash (34)
- b) Beech (16)
- c) Black pine (166)
- d) Fir (127)

- e) Hornbeam (42)
- f) Larch (200)
- g) Mountain oak (77)
- h) Scots pine(190)
- i) Spruce (213)
- j) Swiss stone pine (96)
- k) Sycamore maple (22)



Figure 4.2: One sample images of a bark of all tree species in the dataset.

The dataset of the black pine, fir, larch, scots pine and spruce are divided in 3 sub-classes. The first containing images of the trees when they are younger than 60, in the second one images of trees with an age of 60 to 80, and the last one with images of trees which are older than 80. These separation is necessary because especially the bark of these trees differs at different ages. In the results these subclasses will be combined again.

Furthermore the barks of trees have a high intraclass variance which is often bigger then the difference to an other class. Figure 4.3 shows three images of a black pine older than 80 years and in contrast two images of a scots pine, which are older than 80, and an image of a larch younger than 40. The difference between the three black pines is higher then the difference to one of the other images.

Additional the texture of a bark can be adulterated by various things like knotholes, plant cover (like moss, and lichens) and shadows.



Figure 4.3: Sample images to show the intraclass difference. First row: black pine older than 80. Second row: two images of a scots pine older than 80 and one image of a larch younger than 40.

4.1.3 Needles dataset

The dataset of the needle images contains 275 of 6 Austrian conifers. Each class contains 10 to 114 images. Sample images for each tree species are shown in Figure 4.4, in braces are the number of images for each class:

- a) Black pine (107)
- b) Fir (10)
- c) Larch (114)

- d) Scots pine (10)
- e) Spruce (13)
- f) Swiss stone pine (21)

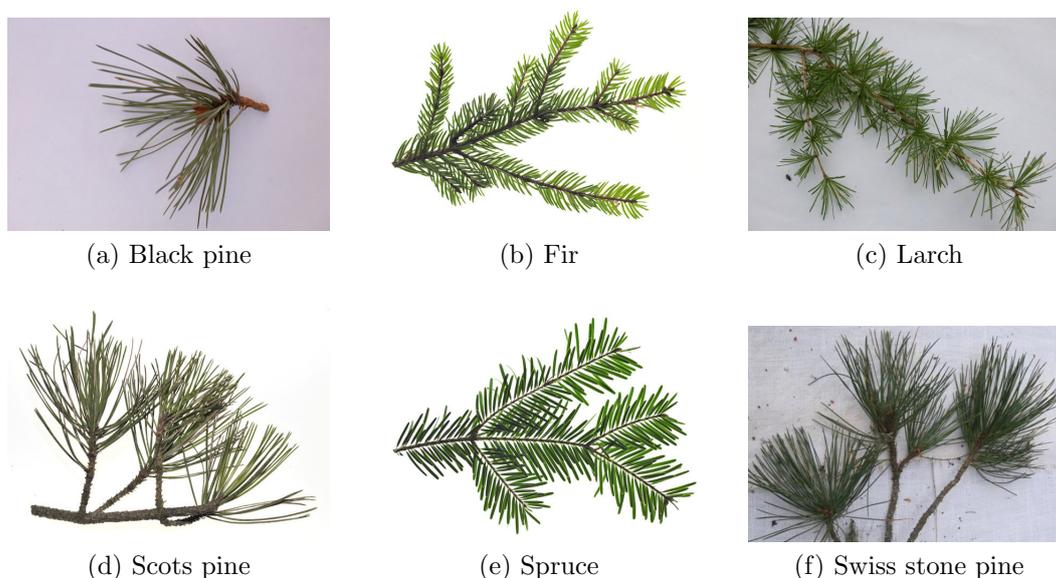


Figure 4.4: One sample images of the needles of all tree species in the dataset.

It has been already mentioned in Section 3.3 that the 6 conifers can be divided into two classes. The first class are the fir and the spruce on which the needles grow separate on the branch and the second class are the species on which the needles grow in clusters. It can be seen that the fir, scots pine, and spruce images have been made with perfect lighting conditions, whereas the other images have been photographed in the nature.

4.2 Distribution of the SIFT features

This section gives a short overview of the distribution of the SIFT features, described in Section 3.1.1, on images of leaves. The distribution is not evaluated on bark images, since the bark covers the whole image. The test were made on the trainings set of the leaf dataset which will be used in Section 4.4, which contains 40 images. Each class has 8 images in this set.

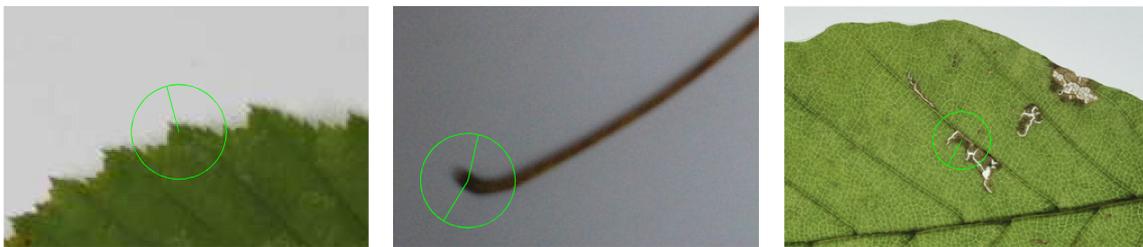
Table 4.1 shows the distribution of the SIFT features on these images. In the first column are the five tree species of this dataset and total means the sum over images in the dataset. Number of features is the sum of the features for each image. "Touching the leaf" means that one part of the SIFT descriptor describes the inside of the leaf and the other part the background, so the sift descriptor describes the border of the leaf if the scale is not high (see Figure 4.5 (a)). "Not touching the leaf" are the SIFT descriptors which only describes the background of the image or the stem of the leaf (see Figure 4.5

(b)). Since the stem of the leaf is similar for every leaf these features are not good for classification. SIFT descriptors on the background can occur due to lighting conditions of the image, so that the gradients of the shadow is high enough to be a stable keypoint. “Withing the leaf” are the SIFT descriptors which only describes the leaf. These can occur due to damaged leaves (see Figure 4.5 (c)) or due to the venation of the leaves.

In total there are 9170 sift features, which makes an average of 229.25 per image. Ash leaves have the most features (402 in average), since they consist of multiple parts and therefore each part generates features. The mountain oak images have an average of 204 features and the sycamore maple an average of 223.5 features. This is because both leaves are lobed respectively sinuate so the perimeter of the leaf is longer and more features are generated there. The beech and the hornbeam images have a similar amount of features with 157.6 respectively 159.1. This is because the border of the beech leaf is simple and less features are generated at the borders but more features are generated within the leaf due to venation and small damages. The serrated leaf of the hornbeam has more features on the border but still the indentations are too small to generate as much features as the mountain oak or the sycamore maple. In total most of the features (90.5%) are located at the border of the leaves which is an accurate description of the leaf shape.

	number of features	touching the leaf	not touching the leaf	within the leaf
Ash	3216	3077	139	17
Beech	1261	877	54	330
Hornbeam	1273	1192	44	37
Mountain oak	1632	1460	94	78
Sycamore maple	1788	1692	40	56
Total	9170	8298	354	518

Table 4.1: Distribution of the SIFT features on images of leaves. Most of the features are located at the border of the leaves.



(a) A SIFT feature located at the border of a hornbeam leaf. (b) Two SIFT features at the stem of a sycamore maple leaf. (c) A SIFT feature located in the inside of a beech leaf due to small damages.

Figure 4.5: Sample images which are showing the different locations for the SIFT features. The circle corresponds to the scale of the features whereas the line shows the orientation of the feature.

Figure 4.6 shows an image of an ash with all 385 keypoint locations. 371 of them are touching the images, which means that a part of the leaf and a part of the background is described. 14 of these features are only describing the background which means they are located on the background or on the stem and the scale is so small that they do not touch the leaf. None of them are within the leaf. Note that some keypoints are located within the leaf but the scale is at least so big that also parts of the background are in the neighborhood of this keypoint.

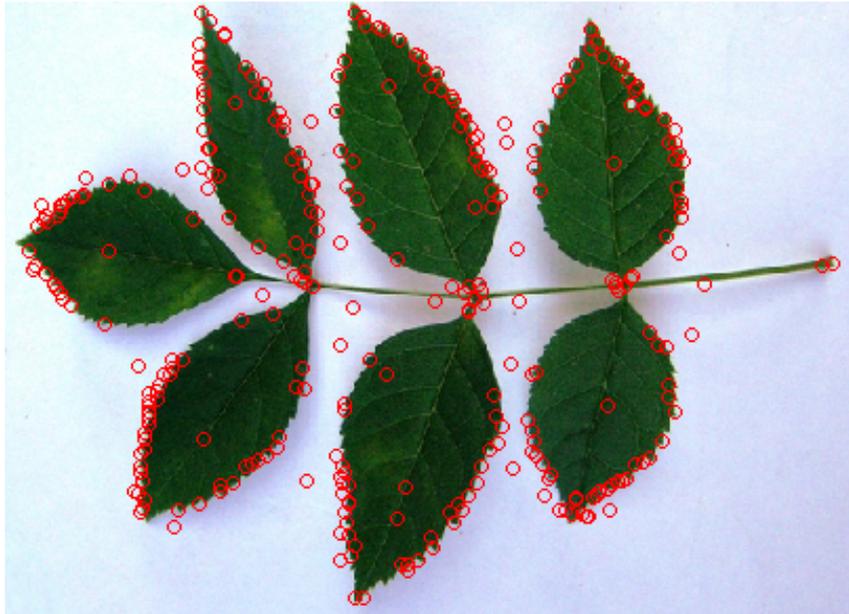


Figure 4.6: An image of an ash showing all 385 keypoint locations. 14 of them are only describing the background or the stem.

4.3 Experiments with experts

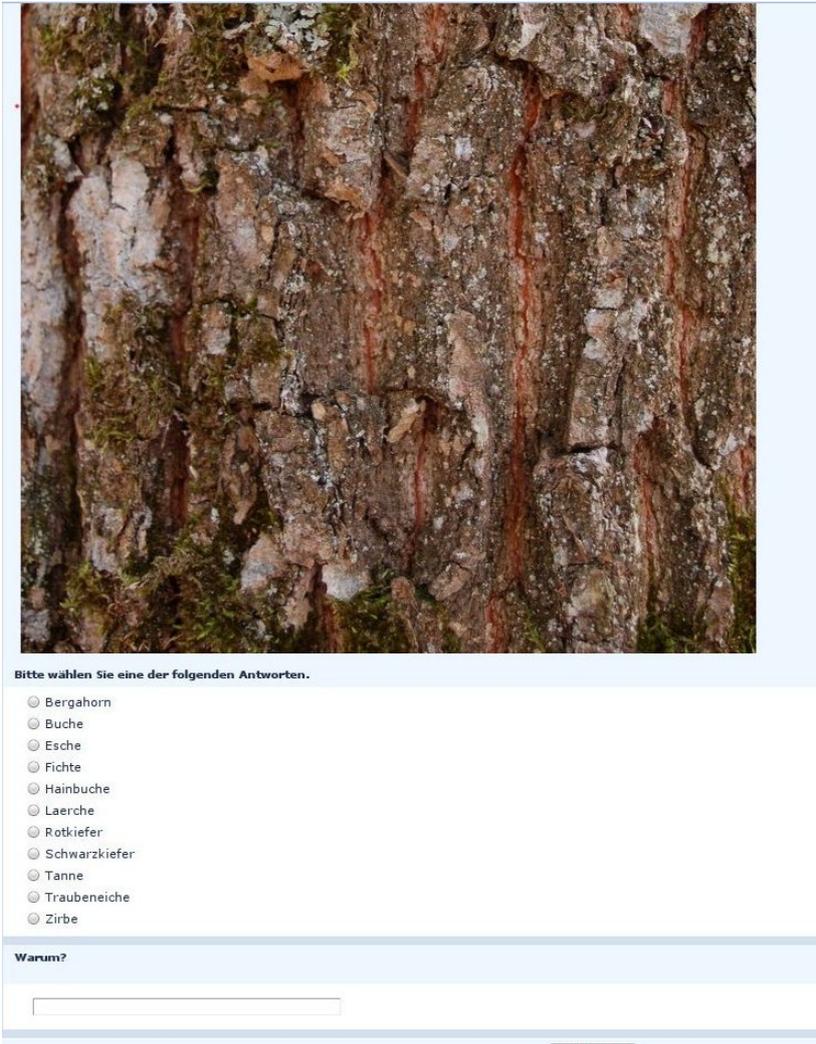
To check if there is enough information in the images to identify the tree species two experiments with experts has been done. Two employees of the "Österreichische Bundesforste AG" attended these experiments. One was a biologist, who studied at the University of Natural Resources and Life Sciences in Vienna and is now working in the natural resource management department. The second attendee was a forest ranger whose forest district lies in Lower Austria and got practical experience for more than 15 years. In the first experiment the experts had to assign bark images to the correct tree species, in the second experiment needle images had to be assigned.

Bark dataset

For this experiment a new dataset was created which is a subset of the bark dataset. 99 images were taken, 9 of each class. From classes which are grouped in subclasses three

images of each subclass were taken. The aim of this experiment was also to check if the visual information in the images (only a small section of the bark) is sufficient enough for a identification of the tree species or if other information are used. These information can be the location of the tree, leaves or needles, the habit of the tree, the surrounding area of the tree, the buds on the branches, or the haptics of the tree.

For this purpose an online survey has been created, see Figure 4.7. All images were presented consecutively and there was no possibility to switch back to an older image and change a choice. For every image all tree species, which occurred in the survey, were presented as possible answers. They were always sorted alphabetically. Additional there was a possibility to enter comments, which were only used for analysis why the attendee made his choice.



Bitte wählen Sie eine der folgenden Antworten.

- Bergahorn
- Buche
- Esche
- Fichte
- Hainbuche
- Laerche
- Rotkiefer
- Schwarzkiefer
- Tanne
- Traubeneiche
- Zirbe

Warum?

Warten

Figure 4.7: A screenshot of the survey with which the experiment was made. An image of a bark is shown (here only a part of it) and the attendees had to chose the appropriate tree species and could add comments.

Table 4.2 shows the results of the first expert. The tree names on the left side are the

true classes, whereas the tree names on the top are the classes the user assigned to an image. The classification rate is 56.6%. All images of the beech were classified correctly, because it is the only tree species in this dataset with a smooth bark. The sycamore maple and the fir had also a good classification rate, 8 from 9 images were identified. The ash and the scots pine 7 respectively 6 images were classified correctly and also 5 images from the hornbeam class. The black pine is the first class with a recognition rate below 50%. Only one third of the images of the swiss stone pine were identified and from the larch, mountain oak, and the spruce only 2 image of 9 were identified.

	Ash	Beech	Black pine	Fir	Hornbeam	Larch	Mountain oak	Scots pine	Spruce	Swiss stone pine	Sycamore maple
Ash	7						1				1
Beech		9									
Black pine			4			2		2	1		
Fir				8				1			
Hornbeam	1	1	1		5						1
Larch			2			2		4		1	
Mountain oak	2		1			1	2	1			2
Scots pine			1			2		6			
Spruce	1	1					2		2		3
Swiss stone pine	1							1	1	3	3
Sycamore maple										1	8

Table 4.2: The confusion matrix of the first expert. The tree names on the left side are the true classes, the names on the top the estimated classes. The overall recognition rate is 56.6%.

Table 4.3 shows the results of the second expert. The recognition rate is 77.8%. All images of the beech, fir, hornbeam, and spruce were classified correctly. Only 2 images of the ash, black pine, and sycamore maple were identified wrong. The larch, mountain oak, and the swiss stone pine have a classification rate of 66%. The worst result has the scots pine where only 2 images were identified correctly.

Both experts said at the end of the experiment that they had the biggest problem by distinguishing the three pine species and the larch. These four tree species have a similar bark, which was already shown in Figure 4.3. All of them have large bark flakes with similar form. According to the experts especially these trees can be recognized by the habit of the tree.

The lower recognition rate for expert one is because in university the distinction of different tree species is not taught from only the bark. Combined characteristics are used. An important part is the adjustment of leaves on a branch. Conifers were not treated during the studies. The recognition rate of expert two of the swiss stone pine was

	Ash	Beech	Black pine	Fir	Hornbeam	Larch	Mountain oak	Scots pine	Spruce	Swiss stone pine	Sycamore maple
Ash	7	1					1				
Beech		9									
Black pine			7					1		1	
Fir				9							
Hornbeam					9						
Larch						6		1		2	
Mountain oak							6				3
Scots pine			3			4		2			
Spruce									9		
Swiss stone pine				1			1	1		6	
Sycamore maple									1	1	7

Table 4.3: The confusion matrix of the second expert. The tree names on the left side are the true classes, the names on the top the estimated classes. The overall recognition rate is 77.8%.

explained, that this tree species does not occur in his district.

The proposed method was also applied to the same dataset which were presented to the experts. The results will be shown in Section 4.5.

Needles dataset

The second experiment with the experts has been done on a subset of the needles dataset. This time only 7 images of each class were taken and presented using the online survey. As answer possibilities all 6 tree species of the dataset were given.

Table 4.4 shows the result of the first expert. The classification rate was 69%. All images of the larch were identified correctly. Only images of the spruce and the swiss stone pine images were classified wrong, the spruce as fir and the swiss stone pine as black pine. Only two images of the black pine and the scots pine were identified correctly.

Table 4.5 shows the result of the second expert. The classification rate was 88.1%. All images of the fir, larch, and spruce were identified correctly, whereas one image of the scots pine and the swiss stone pine were assigned to the wrong class. 4 images of the black pine were identified correctly.

It can be seen in both results that the identification of the larch was no problem for the experts. All fir and the spruce images were identified correctly by the second expert, whereas the first expert identified one fir image as spruce and vice versa. It has already been explained that the fir and the spruce are the trees of the dataset were the needles

	Black pine	Fir	Larch	Scots pine	Spruce	Swiss stone pine
Black pine	2			3		2
Fir		6			1	
Larch			7			
Scots pine	3	1		2		1
Spruce		1			6	
Swiss stone pine	1					6

Table 4.4: The confusion matrix of the first expert. The tree names on the left side are the true classes, the names on the top the estimated classes. The overall recognition rate is 69%.

	Black pine	Fir	Larch	Scots pine	Spruce	Swiss stone pine
Black pine	4			1		2
Fir		7				
Larch			7			
Scots pine				6		1
Spruce					7	
Swiss stone pine	1					6

Table 4.5: The confusion matrix of the second expert. The tree names on the left side are the true classes, the names on the top the estimated classes. The overall recognition rate is 88.1%.

grow separate on the branch which explains this good classification rate. The problem for the experts were the three pine trees, because on some images it was even hard for them to count the number of the needles in the cluster.

4.4 Experiments on the leaf dataset

In our experiment with the leaf dataset all 134 images were used. The training set contains 8 images per class whereas the test sets are of a size from 14 to 26. Experiments has shown that 30 cluster centers for each class led to the best results with our dataset.

In the first experiment the methodology as described in Chapter 3 was applied to the images. The images were assigned to the class with the highest output function of the SVMs. The results of this experiment is presented in Table 4.6. The tree names in the top row are the estimated classes whereas the the tree names in the first column are the true classes. The total classification rate is 93.6%. The best performance was achieved for the sycamore maple and the mountain oak leaves where all images are classified correctly. The hornbeam has a recognition rate of 96%, followed by the beech with 91%. The method performs poorest on the ash with a classification rate of 82%. The reason for this low rate is the high intraclass variance in the ash leaves dataset. Figure 4.8 shows three different images of an ash. It can be seen that the leaf can consists of a variable amount of parts and the form of them can also differ a lot.

	Ash	Beech	Hornbeam	Mountain oak	Sycamore maple
Ash	14	1		1	1
Beech		20		2	
Hornbeam	1		25		
Mountain oak				14	
Sycamore maple					15

Table 4.6: Confusion matrix of the first experiment on the leaf dataset. The tree names on the top are the estimated classes, the names on the left side the true classes.



Figure 4.8: Three images of ash leaves to show the high intra-class difference.

The next experiment shows the effect of introducing a threshold to the output function of the multi-class SVM as mentioned in Section 3.1.2. This threshold is introduced to determine whether a leaf is in the image or not. For this purpose a set of images is introduced which contains images where no leaves occur. 15 images has been taken from the "Caltech google-things" database [Cal04]. These images show various things of the daily life, like bicycle, bookshelf, face, car, and landscape. These images were chosen because they contain information that can possible be photographed by an user as input images for the method.

If the output function is lower than a given threshold the images are assigned to a new output class called "no category". The value of the threshold is a trade-off between classifying an image correctly and a wrong classification or being assigned to the "no category" class. The higher the threshold the more images are assigned to the "no category" class but also correct identified leaves of the first experiments are not longer recognized. If the threshold is to low images where no leaf occurs are put in a leaf class and also wrong classified leaves occurs. Experiments have shown that the best results can by achieved by applying a threshold of 60%, where not all things of the "google-things" database are recognized and assigned to a leaf class but no wrong classified leaves appear. Additional experiments with a threshold of 70% and 80% have been done and will also be presented.

Table 4.7 is showing the results for this experiments when a threshold of 60% is introduced to the output function of the SVM. The overall classification rate is 85.3% and the rate for the leave classes is 88.3%. All wrong classified images of the first experiment are put in the "no category" class but also correct classified images were put in this class, lowering the overall recognition rate. Equal to the first experiment all mountain oak leaves has been identified correctly but one image of the sycamore maple was assigned to the "no category" class lowering the classification rate for this class to 93%. 22 images of the hornbeam class are correctly assigned but 4 images were not identified. In the first experiment 25 hornbeam leaves were identified and one was misclassified. So the recognition rate for this class is only 85%. For the beech database the new recognition rate is 86%. Both images which were assigned to a wrong class in the first experiment are now put into the "no category" class but also one correct classified image. The threshold has the best effect on the ash class where all three misclassified images of the first experiment are now assigned to the "no category" class, leaving the classification rate at 82%. For the new introduced "google-things" dataset four images is classified as beech and one image is classified as mountain oak. 10 images were correctly assigned to the "no category" class. This is a classification rate of 66%.

	Ash	Beech	Hornbeam	Mountain oak	Sycamore maple	no category
Ash	14					3
Beech		19				3
Hornbeam			22			4
Mountain oak				14		
Sycamore maple					14	1
"google-things"		4		1		10

Table 4.7: Confusion matrix of the second experiment. The tree names on the top are the estimated classes, the names on the left side the true classes.

When raising the threshold to 70% the overall classification rate is still at 84.4% but the rate for only the leave classes drops to 84%. Since with a higher threshold more images will be assigned to the "no category" class only two images of the "google-things" dataset remains wrong classified. But also two images of the hornbeam and the ash class are no longer identified correctly which lowers the recognition rate for these classes to 71 respectively 76%.

When the threshold of the output function is raised even more all images of the "google-things" dataset are correctly assigned to the "no category" class. The classification rate of the leaves drops to 70% since more images are put in the "no category" class.

4.5 Experiments on the bark dataset

For the experiment with the bark dataset first all 1182 images are used. Since some classes have a small number of images (the beech set being the smallest with 16 images) the first trainings set contains only 10 images per class. The amount of cluster centers has been set to 30 per class which was evaluated experimentally. First the SIFT with bag of words approach was applied to this dataset, afterwards a combination of the GLCM and wavelet features are used as input to the SVM.

Table 4.8 shows the result for this experiment. The overall classification rate is 64.2%. The beech classification rate with 100% but since there is only one image remaining for the test set this rate is not representative. The second best recognition rate is achieved on the fir bark with 78%. 76 out of 97 images are classified correctly. 75% of the larch images were assigned to the correct class (116 out of 155 images), followed by the mountain oak with a classification rate of 73%. 45 out of 62 images are assigned to the correct class. The largest test set are spruce images with an number of 168, 70% of them were recognized correctly. The scots pine and the sycamore maple have the same recognition rate with 57% and contain 145 respectively 7 images. 6 out of 12 hornbeam images were classified correctly. And three classes are below 50%, namely the black pine with 49% (121 images in total), the swiss stone pine with 49% (the test set contains 51 images), and the ash with 39% (from 18 images).

As previously shown in Figure 4.3 the bark of the larch, black pine, scots pine, and swiss stone pine are similar, this can also be seen in this results. 27% of the black pine images (33 images) and 26% of the scots pine images (37 images) are classified as Larch. Also 18% of the scots pine (22 images) and 18% of the swiss stone pine (9 images) are assigned to the scots pine class. Furthermore the recognition rate of the ash is low with 39% this is because the ash trainings set is not divided into subclasses.

Since a trainings set size of 15 images is not sufficient to get a good description of the bark, the trainings set has been expanded to 30 images. Because some classes, namely beech, hornbeam, and sycamore maple, have less than 30 images in the whole dataset there are no more images of these classes in the test set. But all available images are staying in the trainings set to get results which are comparable with the previous experiment. The number of centers for each class remains at 30.

Table 4.9 shows the results for this experiment. The overall classification rate raises to 69.7%. The highest recognition rate has the Spruce with 82% (101 out of 123 images), followed by the fir with 76% (51 out of 67). 55 images (which are 72%) of the black pine images were assigned to the correct class. The larch and the swiss stone pine have a classification rate of 70 respectively 67% (77 out of 110 respectively 12 out of 18 images). 62% of the mountain oak image are assigned correctly which are 29 out of 47. The scots pine has a recognition rate of 53% (53 out of 100). The ash has the poorest result with 33% but since there are only three images remaining in the test set this result is not representative.

A combination of the GLCM features (contrast, correlation, energy, and homogeneity), which were presented in Section 3.2.2, and the average energy of the wavelets coefficients, which were presented in Section 3.2.3 were taken as input for a one-vs-all SVM. The GLCM features were calculated for 0, 45, 90, and 135 degrees with a distance of 1 and

	Ash	Beech	Black pine	Fir	Hornbeam	Larch	Mountain oak	Scots pine	Spruce	Swiss stone pine	Sycamore maple
Ash	7	2		3	3		1	1		1	
Beech		1									
Black pine			59		5	33		22			2
Fir	2			76	1		5		6	5	2
Hornbeam	3	2			6				1		
Larch			15	1	1	116		13	4	2	3
Mountain oak	2				1	5	45		1	5	3
Scots pine	2		8	1	2	37		82		7	6
Spruce	1	2		9	4	4	10	6	117	6	9
Swiss stone pine			5		2	7	3	9		24	1
Sycamore maple			2					1			4

Table 4.8: Confusion matrix of the experiment on the bark dataset with a trainings set size of 15 images. The tree names on the top are the estimated classes, the names on the left side the true classes.

	Ash	Beech	Black pine	Fir	Hornbeam	Larch	Mountain oak	Scots pine	Spruce	Swiss stone pine	Sycamore maple
Ash	1							1	1		
Black pine	1		55			10		10			
Fir			1	51		2	3	1	9		
Larch			13	1		77	1	11	2	3	2
Mountain oak	2		2		1	3	29			6	4
Scots pine	1		10	4	2	19		53	4	6	1
Spruce	2	1		4	2		7	1	101	1	4
Swiss stone pine					1	2	1	2		12	

Table 4.9: Confusion matrix of the experiment on the bark dataset with a trainings set size of 30 images. The tree names on the top are the estimated classes, the names on the left side the true classes. Tree species which no longer occurs in the test set are skipped on the left side.

5 pixels. The depth of the wavelet packet was 5. The image was first normalized and then divided into 8 gray values which were used to build the GLCM. The results for this experiments are shown in Table 4.10. The overall classification rate is 61.2%. All three remaining images in the test set are identified correctly. The classes with the second best recognition rate are the fir and the spruce with 67%, followed by the fir with 65%. 53% of the black pine were assigned to the correct class, whereas the method has the worst performance on the mountain oak and swiss stone pine dataset with 43 respectively 39%. 5 out of 18 swiss stone pine images were assigned to the scots pine class, which is 28% and also 20% of the black pine images are assigned to this class. 21% of the black pine images are identified as larch images. So this methods has the same problem with the three pine species and the larch. Since the SIFT method has a better recognition rate and can be applied to leaves and bark images the following experiments were only done with this method.

	Ash	Beech	Black pine	Fir	Hornbeam	Larch	Mountain oak	Scots pine	Spruce	Swiss stone pine	Sycamore maple
Ash	3										
Black pine			40			16		15	1	3	1
Fir				45		3		9	7	2	1
Larch			12	1		71		21		5	
Mountain oak	4		1			2	20	8	4	7	1
Scots pine			11	2		17		64	2	3	1
Spruce	4	3		6	12	4	2	3	83	2	4
Swiss stone pine			1	2		1		5	1	7	1

Table 4.10: Confusion matrix of the experiment with combined features of the GLCM and wavelets. The trainings set contained maximal 30 images per class. Classes where no images are left are skipped in the first column.

The next experiment on the bark dataset is the comparison with the results of the experiments with the experts in Section 4.3. As test set the same dataset like before is used, containing 9 images from each class. The trainings set are maximal 30 images. For those classes which have less then 39 images the rest of the dataset was used as trainings set. The number of centers per class for the bag of word method remains at 30.

Table 4.11 is showing the results for this dataset. The classification rate is 65.6%, which is approximately between the rate of the two experts. The ash, beech, black pine, fir, and spruce have a recognition rate of 88.8%. The hornbeam and the swiss stone pine have a recognition rate of 77.7%. 6 of the 9 images of the scots pine were classified correctly and 5 of the mountain oak images are assigned correctly. None of the larch or sycamore maple are identified. The reason why none of the sycamore maple is classified correctly is that in these images shadows occur and the trees are covered with moss and

lichens. 6 of the larch images are assigned to the black pine class, also three images of the scots pine, which confirms that the three pines and the larch are hard to identify. This was already shown in Figure 4.3 and also mentioned by the experts in Section 4.3.

	Ash	Beech	Black pine	Fir	Hornbeam	Larch	Mountain oak	Scots pine	Spruce	Swiss stone pine	Sycamore maple
Ash	8		1								
Beech		8							1		
Black pine			8							1	
Fir				8					1		
Hornbeam					7		2				
Larch			6			-		2	1		
Mountain oak	3						5	1			
Scots pine			3					6			
Spruce								1	8		
Swiss stone pine			1					1		7	
Sycamore maple	1		4	1					1	2	-

Table 4.11: Confusion matrix of the experiment on the bark dataset used for the experiments with the experts in Section 4.3. The tree names on the top are the estimated classes, the names on the left side the true classes.

4.6 Experiments on the needle dataset

In Section 3.3 the problems with identifying the tree species from photos of the needles were explained. When dealing with the tree species on which the needles grow separate on the trees only one characteristics for the identification remains. The needles of fir and spruce only differ on the backside, at the endings, and on the grow directions. The needles of the spruce are growing in one plane whereas the needles of the fir can grow in any direction. These characteristics can not be used for an identification since the photo can not be controlled and due to overlapping needles on spruce images the plane on which the needles grow can not be determined.

The first experiment was done using the fir and the spruce dataset. Four images of each class are used as trainings set and the rest (5 images of the fir, 9 images of the spruce) are used as test set. The endpoints are found by using the method described in Section 3.3. For each needle ending 11 features were calculated, namely the eccentricity, extent, equivalent diameter, solidity, 7 improved moment invariants by Chen [Che93], and the bending energy.

Table 4.12 shows the result of this experiment for every image. For each image between 33 and 158 endings are found. The features of these endings were used as input for a SVM.

Two spruce images were not classified correctly (the percentage is below 50%). In the first image 42 out of 93 endings were classified as spruce endings, in the second image only 48 out of 158. Figure 4.9 shows a zoomed section of these two images. Some needles of the spruce are rotated, thus the endings of these needles are pointed and identified as endings of a fir.

	Number of endings	Correctly classified	percentage
Fir 1	116	110	94,83
Fir 2	110	110	100,00
Fir 3	88	80	90,91
Fir 4	98	60	61,22
Fir 5	94	83	88,30
Spruce 1	113	90	79,65
Spruce 2	53	44	83,02
Spruce 3	93	42	45,16
Spruce 4	158	48	30,38
Spruce 5	52	38	73,08
Spruce 6	33	23	69,70
Spruce 7	64	52	81,25
Spruce 8	100	67	67,00
Spruce 9	56	46	82,14

Table 4.12: Results of the experiment on the fir and spruce images. In each image between 33 and 158 endings were found. Two images were classified not correctly.



Figure 4.9: Rotated needles in the two spruce images which were classified not correctly.

4.7 Evaluation of the multi-class SVM

This section evaluates the result when using a one-vs-one multi-class SVM described in Section 3.1.2. In contrast to the one-vs-all SVM where every dataset of one class is compared to all other results this approach compares each class to each other. So $\frac{M*(M-1)}{2}$ SVMs are generated where M is the amount of classes. Every SVM increases the vote of the class according to the output function. The class with the highest number of votes is assigned as output to the multi-class SVM.

Table 4.13 shows the results with a one-vs-one SVM on the bark dataset used for the experts in Section 4.3. The overall classification rate of the one-vs-all approach was shown in Section 4.5 and was 65.6%. With the one-vs-one approach the classification rate drops to 55.5%. Again, none of the larch images were identified correctly but two images of the sycamore maple are assigned to the appropriate class. The spruce and the hornbeam images have the same classification rate but the rates of all other classes are lower.

	Ash	Beech	Black pine	Fir	Hornbeam	Larch	Mountain oak	Scots pine	Spruce	Swiss stone pine	Sycamore maple
Ash	7		1					1			
Beech		6	1						2		
Black pine			7					2			
Fir				6					3		
Hornbeam	1				7				1		
Larch			5			-		2	1	1	
Mountain oak	4						3	1		1	
Scots pine			2					4	1	2	
Spruce									8	1	
Swiss stone pine			1		1			2		5	
Sycamore maple			2	1					4		2

Table 4.13: Confusion matrix of the experiment with a one-vs-one multi-class SVM on the bark dataset. The total classification rate is 55.5%. The tree names on the top are the estimated classes, the names on the left side the true classes.

Table 4.14 shows the result when using a one-vs-one multi-class SVM on the leaf dataset. The dataset is the same as in Section 4.4 where the classification rate was 93.6%. With a one-vs-one SVM the classification rate raises slightly to 95.7% which means that one leaf more is classified correctly. In contrast to the one-vs-all SVM two additional images of the ash and one additional image of the beech are assigned to the correct class whereas one more image of the hornbeam class is classified as an ash leaf. However the one-vs-one approach has the advantage that no threshold can be introduced to the output function of the multi-class SVM which can be used to identify images which are

misclassified or images with no leaves in it (see Section 4.4).

	Ash	Beech	Hornbeam	Mountain oak	Sycamore maple
Ash	16				1
Beech		21		1	
Hornbeam	2		24		
Mountain oak				14	
Sycamore maple					15

Table 4.14: Confusion matrix of the experiment with a one-vs-one multi-class SVM on the leaf dataset. The total classification rate is 95.7%. The tree names on the top are the estimated classes, the names on the left side the true classes.

4.8 Evaluation of the results

Since the proposed system is planned to be published on mobile devices the recognition rate has to be high enough that users do not lose the motivation to use such a system. The recognition rate of the leaf dataset with up to 95% is high enough for such a program. The identification rate of the bark images is only 69%, which means that almost one of three images of a bark is assigned to the wrong class. The experiments with the experts showed that even they can not assign all images of the bark to the right tree, which means that only a section of the bark is not sufficient for a perfect classification. They said that normally other characteristics of the tree would be necessary to make a classification for them easier, like the habit of the tree, the buds on the branches, and the location where the tree grows.

A solution to increase the overall recognition rate would be to let the users take two images. One image of a leaf and one image of the bark of the tree. For both images the percentage of the belonging to a class can be calculated and the one with the highest sum of percentage can be taken. Since the broad leaf trees do not carry leaves during the winter and in autumn the leaves lie on the ground and one single leaf can not be exactly assigned to a specific tree, it is not possible for the user to provide both images. Therefore and due to the good results of the experiments on the leaf dataset (classification rate of 93.6%, see Section 4.1.1) an experiment with the combination of leaf and bark images have not been carried out.

The identification of the conifers can not be supported by such a combination since the needles and the bark of the three pine species and the larch are similar. Since the experiments on the bark images showed that only 0.05% of the spruce images are identified as fir and 0.06% of the fir images are identified as spruce images (see Section 4.5, experiments on the whole dataset with a training set size of 15) experiments with the combination of bark and needle images have not been carried out.

Chapter 5

Conclusion

This thesis presented a method for an automated identification of tree species from photos of leaves and bark. The method described uses local descriptors calculated on gray scale images. This has been done to avoid segmentation of the leaves. Since local descriptors can keep up with texture classification methods this approach was also used for the identification of bark images.

The proposed method consisted of 3 steps. First the images were transformed into a normalized gray scale image. There the SIFT features were calculated by searching for keypoints in the images using DoG at different scales. For this keypoints the neighborhood was described using orientation histograms. These features were scale, rotation, translation invariant. Furthermore the were invariant against linear and non-linear changes of illumination. These features were then used for a bag of words method, where the features of the trainings set were clustered to form a codebook. With this codebook new images can be described as an histogram of occurrences where the nearest cluster center is determined for each feature. This histogram can then be classified using a one-vs-all SVM. It has been showed that in images with leaves most of the SIFT features are at the border of the leaf giving an accurate description of the shape.

For the automated classification of the bark the same method was used and compared to an approach of a combination of GLCM and wavelets which were also classified using a one-vs-all SVM.

No method has been found for the classification of the needle images. A method has been presented to distinguish between fir and spruce because the needles of these two trees differ. The images have to be in a good quality, which means that the endings of the needles must not be blurred. The difference between the needles of the other conifers are only the number of needles which grow in one cluster on the branch and due to overlapping needles these can not be counted.

Experiments and results have been presented for a dataset of leaves and bark images. To gain a reference for the results an experiment with two experts have been made where they have to classify the bark images. The experts had a classification rate of 56.6 respectively 77.8%. The proposed method was also applied to this dataset and a classification rate of 65.6% is achieved which is approximately in between. The next experiment presented was on the leaf dataset and also a test to show how images which are not containing a leaf can possibly be determined. The classification rate for the leaf dataset was

93.6%. When adding a threshold to determine whether a leaf is in image and by adding new images of various things the classification rate drops to 84.4% and one third of the additional images were categorized that no images occur in them. When applying the proposed method to the bark dataset a classification rate was 64.2% but a small training set was used. When increasing the trainings set the classification rate raises to 69.7%.

An experiment on the needles dataset has shown that it is possible to tell fir and spruce apart. Two out of 14 images were not classified correctly, the reason for this is the rotation of the needles on the branch so that the blunt needles of the spruce appear pointed as the endings of a fir needle.

The proposed method can not be compared with results of the related work since different datasets are used with different tree species from all over the world.

Disadvantages of the proposed method

The results on the leaf dataset look promising that this part of the assignment can be solved by the proposed method. In contrast to the results for the bark dataset which are not sufficient enough for a real world application. Since only 69% of the bark images are classified correctly it still remains that nearly every third image would be misclassified which is demotivating for a user to continue to use such a system. Even the recognition rate of the experts is not high enough for a real world application, since every fifth tree would be misclassified.

Additionally the calculation of the SIFT features, especially on the bark images where thousands of features are calculated, is computational intensive. Furthermore these features have all be compared to the cluster centers to get the occurrence histogram for the bag of word method. Current mobile devices do not have the computing power to take over all steps of the method. When broadband internet is provided in the region images can be sent to a server or only the SIFT features can be calculated on the devices and the descriptors are sent to a server. This can reduce the computation time on the mobile device but requires data connection.

Due to the clustering of the features in the bag of word method online learning is not possible with this method. Especially if the data is sent via data connection numerous test data could be collected which could lead to better results in the classification.

Advantages of the proposed method

The advantages of the proposed method is that no segmentation is required. Segmentation is used to get the boundaries of the leaves and describing the leaf using morphological features. Every feature described in Section 2.2.1 depends on the results of this segmentation. A wrong segmentation will lead to wrong features. It can occur due to overlapping leaves, shadows, or pinnate leaves. Additional if the leaf is damaged wrong features are calculated. The proposed method also can handle images where multiple leaves occur, damaged leaves, or images that do not contains a complete leaf. Nevertheless the background of the image still has to be monochrome or blurred. Otherwise a majority of the

SIFT features are located at the background which are influencing the classification. But these requirements are needed for segmentation too.

In addition to this the proposed method can be used to classify images of leaves and bark. Since the features are calculated in any case an additional step can be introduced to tell images of leaves and barks apart. With this addition step user input is not necessary.

A possible advantage of the proposed method is that it can be easily adopted to recognize other objects. Only the vocabulary of the bag of words approach has to be changed and various other things can be identified.

Outlook

Since the classification rate of the images is not high enough for a real world application an extension of the proposed method is to let the user take two photos of each tree. One from the bark and the other from the leaves respectively needles. This will increase the recognition rate but will slow down the program.

Additionally since the number of mobile devices with integrated GPS is raising this information could also be used to assist the method. Also the distribution of the tree species in the area can be taken into account when identifying the tree species. The sea leaves on which the photos was taken is also an indicator.

Bibliography

- [ABR64] M A Aizerman, E M Braverman, and L I Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control* 25, pages 821–837, 1964.
- [AMK97] Sadegh Abbasi, Farzin Mokhtarian, and Josef Kittler. Reliable Classification of Chrysanthemum Leaves through Curvature Scale Space. *Scale-Space Theory in Computer Vision, First International Conference, Scale-Space'97, Proceedings*, pages 284–295, 1997.
- [BCB09] André Ricardo Backes, Dalcimar Casanova, and Odemir Martinez Bruno. A complex network-based approach for boundary shape analysis. *Pattern Recognition*, 42(1):54–67, 2009.
- [BGV92] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. *COLT '92: Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.
- [BL02] Matthew Brown and David Lowe. Invariant Features from Interest Point Groups. In *British Machine Vision Conference*, pages 656–665, 2002.
- [Cal04] Caltech. "google-things database". 2004. <http://www.robots.ox.ac.uk/~vgg/data/data-cats.html>, accessed September 2010.
- [CDF⁺04] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cedric Bray. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.
- [CHC03] Zheru Chi, Li Houqiang, and Wang Chao. Plant species recognition based on bark patterns using novel Gabor filter banks. In *Proc. International Conference on Neural Networks and Signal Processing*, volume 2, pages 1035–1038, 2003.
- [Che93] Chaur-Chin Chen. Improved moment invariants for shape discrimination. *Pattern Recognition*, 26(5):683–686, 1993.
- [CK93] T. Chang and C.-C.J. Kuo. Texture analysis and classification with tree-structured wavelet transform. *IEEE Transactions on Image Processing*, 2(4):429–441, oct. 1993.

- [CPW00] C. H. Chen, L. F. Pau, and P. S. P. Wang, editors. *Handbook of Pattern Recognition and Computer Vision*. World Scientific Publishing Co., Inc., 2000.
- [Du06] Ji-Xiang Du. Computer-Aided Plant Species Identification (CAPSI) Based on Leaf Shape Matching Technique. *Transactions of the Institute of Measurement and Control*, 28 No. 3:275–285, 2006.
- [DWG05] Ji-Xiang Du, Xiaofeng Wang, and Xiao Gu. Shape Matching and Recognition Base on Genetic Algorithm and Application to Plant Species Identification. In *Advances in Intelligent Computing, International Conference on Intelligent Computing, Proceedings, Part I*, pages 282–290, 2005.
- [DWZ07] Ji-Xiang Du, Xiaofeng Wang, and Guo-Jun Zhang. Leaf shape based plant species recognition. *Applied Mathematics and Computation*, 185(2):883–893, 2007.
- [dZRP07] Paul M. de Zeeuw, Elena Ranguelova, and Eric J. Pauwels. Towards an online image-based tree taxonomy. In *ICDM'07: Proceedings of the 7th industrial conference on Advances in data mining*, pages 296–306, Berlin, Heidelberg, 2007. Springer-Verlag.
- [ea10] Amanda Lenhart et al. *Social Media and Young Adults*. Pew Research Center, 2010.
- [God07] Jean-Denis Godet. *Bäume und Sträucher bestimmen und nachschlagen*. Eugen Ulmer KG, 2007.
- [Göt05] Gerit Götzenbrucker. Jugend im Netz? Effekte mobiler Kommunikation im jugendlichen Alltag. Eine qualitative Studie im Ballungsraum Wien. *kommunikation@gesellschaft*, Jg. 6:Beitrag 3, 2005.
- [Hor09] John Horrigan. *Wireless-Internet-Usage*. Pew Research Center, 2009.
- [HSD73] Robert M. Haralick, K. Shanmugam, and Its'Hak Dinstein. Textural Features for Image Classification. *IEEE Transactions on Systems, Man and Cybernetics*, 3(6):610–621, nov. 1973.
- [Hu62] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187, February 1962.
- [Hua06] Zhi-Kai Huang. Bark Classification Using RBPNN Based on Both Color and Texture Feature. In *IJCSNS - International Journal of Computer Science and Network Security, Vol. 6 No. 10 pp. 100-103*, 2006.
- [HZDW06] Zhi-Kai Huang, Chun-Hou Zheng, Ji-Xiang Du, and Yuanyuan Wan. Bark Classification Based on Textural Features Using Artificial Neural Networks. *Third International Symposium on Neural Networks, 2006, Proceedings, Part II*, pages 355–360, 2006.

- [Joa98] Thorsten Joachims. Text categorization with support vector machines: learning with many relevant features. *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 137–142, 1998.
- [LHWL09] Xun Liu, Keyi Hou, Liliang Wang, and Ping Liu. Index and Recognition for the Shape Contour of Plant Leaves. In *Proc. WRI Global Congress on Intelligent Systems GCIS '09*, volume 4, pages 436–440, 2009.
- [Lin94] Tony Lindeberg. Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 21:224–270, 1994.
- [Llo82] Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–136, 1982.
- [Low99] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. Seventh IEEE International Conference on Computer Vision The*, volume 2, pages 1150–1157, September 20–27, 1999.
- [Low04] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [LP08] Huang Lin and He Peng. Machine Recognition for Broad-Leaved Trees Based on Synthetic Features of Leaves Using Probabilistic Neural Network. *International Conference on Computer Science and Software Engineering*, 4:871–877, 2008.
- [Mal89] S.G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, jul. 1989.
- [Mik02] K. Mikolajczyk. Detection of Local Features Invariant to Affine Transformations, Application to Matching and Recognition. *PhD Thesis, Institut National de Polytechniques de Grenoble, France*, 2002.
- [Nix10] Steve Nix. Tree Identification Using a Tree Leaf Key. 2010. http://forestry.about.com/od/treeidentification/tp/tree_key_id_start.htm, accessed August 2010.
- [PdZR09] Eric J. Pauwels, Paul M. de Zeeuw, and Elena B. Rangelova. Computer-assisted tree taxonomy by automated image recognition. *Eng. Appl. Artif. Intell.*, 22(1):26–31, 2009.
- [SCLF04] Jiatao Song, Zheru Chi, Jilin Liu, and Hong Fu. Bark classification by combining grayscale and binary texture features. In *Proc. International Symposium on Intelligent Multimedia, Video and Speech Processing*, pages 450–453, 2004.
- [VC74] V. N. Vapnik and A. Ya. Chervonenkis. *Theory of Pattern Recognition*. Nauka, USSR, 1974.

- [WBX⁺07] S.G. Wu, F.S. Bao, E.Y. Xu, Yu-Xuan Wang, Yi-Fan Chang, and Qiao-Liang Xiang. A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network. In *Proc. IEEE International Symposium on Signal Processing and Information Technology*, pages 11–16, 2007.
- [WCF03] Z. Wang, Z. Chi, and D. Feng. Shape based leaf image retrieval. *IEE Proceedings - Vision, Image and Signal Processing*, 150(1):34–43, 2003.
- [WCFW00] Zhiyong Wang, Zheru Chi, Dagan Feng, and Qing Wang. Leaf Image Retrieval with Shape Features. *VISUAL '00: Proceedings of the 4th International Conference on Advances in Visual Information Systems*, pages 477–487, 2000.
- [WDH⁺04] Yuan-Yuan Wan, Ji-Xiang Du, De-Shuang Huang, Zheru Chi, Yiu-Ming Cheung, Xiao-Feng Wang, and Guo-Jun Zhang. Bark texture feature extraction based on statistical texture analysis. In *Proc. International Symposium on Intelligent Multimedia, Video and Speech Processing*, pages 482–485, 2004.
- [WDZ05] Xiaofeng Wang, Ji-Xiang Du, and Guo-Jun Zhang. Recognition of Leaf Images Based on Shape Features Using a Hypersphere Classifier. *Advances in Intelligent Computing, International Conference on Intelligent Computing, 2005, Proceedings, Part I*, pages 87–96, 2005.
- [YCL⁺04] Yanhua Ye, Chun Chen, Chun-Tak Li, Hong Fu, and Zheru Chi. A computerized plant species recognition system. In *Proc. International Symposium on Intelligent Multimedia, Video and Speech Processing*, pages 723–726, 2004.
- [ZMLS06] Jianguo Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study. In *Proc. Conference on Computer Vision and Pattern Recognition Workshop*, page 13, June 17–22, 2006.