Czech Technical University in Prague

Faculty of Electrical Engineering

Diploma thesis

Image and Video-based Recognition of Natural Objects

Tomáš Sixta

Supervisor: Prof. Ing. Jiří Matas, Ph.D.

Study programme: Open Informatics

Specialisation: Artificial Intelligence

May, 2011

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Cybernetics

DIPLOMA THESIS ASSIGNMENT

Student: **Bc. Tomáš Sixta**

Study programme: Open Informatics

Specialisation: Artificial Intelligence

Title of Diploma Thesis: **Image and Video-based Recognition of Natural Objects**

Guidelines:

1. Familiarize yourself with the state-of-the-art in video and image-based recognition of natural objects
2. Propose a visual method for recognition of natural objects in near real-time. Consider both a passive and active methods
3. Implement the metod and evaluate its performance on publicity available data

Bibliography/Sources:

Forsyth, D.A.; Ponce, J.: Computer Vision: A Modern Approach. New Jersey, 2002.
Sonka, M.; Hlavac, R.: Image Processing, Analysis and Machine Vision. Toronto, 2008.

Diploma Thesis Supervisor: Prof.Ing. Jiří Matas, Ph.D.

Valid until: end of the summer semester of academic year 2011/2012

prof. Ing. Vladimír Mařík, DrSc.
Head of Department

L.S.

prof. Ing. Boris Šimák, CSc.
Dean

Prague, April 1, 2011

Acknowledgment

I would like to thank my supervisor prof. Jiří Matas for his uncountable advices and patient lead through the thesis development. I am also very grateful to all the people who helped me by any suggestion, advice or who simply supported my effort.

Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

V Praze dne..... .....

Podpis

Abstract

The topic of this work is the recognition of natural objects from images. Thanks to the recent development of mobile devices it is easy to take a picture of a natural object (e.g. plant, animal, fungi) but its identification is difficult and it might require expert knowledge even with a proper identification key. In this work, we address the problem of identification of plants from images of their leaves and bark. Unlike moving animals, they are easy and safe to photograph, they can be found everywhere and identification of a significant number of species does not require any special equipment like a microscope or a DNA sequencer.

To perform recognition from images of leaves, we make use of the Inner Distance Shape Context (Ling and Jacobs [10]) and recognition from images of bark utilises Multi-Block Local Binary Patterns (Liao et al. [44]). Both methods are efficient enough to be implemented as an application for a mobile phone. Recognition of one leaf does not take more than 2 seconds with a database with 954 items and one image of bark can be classified in 3 seconds with a database with 543 items. Classification accuracy was measured on two datasets: The Flavia dataset (leaves [19]) and the Österreichische Bundesforste AG dataset (bark [22]). On the Flavia dataset, top one, two and three candidates included correct class in 83.3%, 91.0% and 94.3% of cases respectively. On the Österreichische Bundesforste AG dataset, top one, two and three candidates included correct class in 70.1%, 87.8% and 93.9% of cases respectively.

Contents

1 Introduction

The topic of this work is the recognition of natural objects from images. According to [3] there is currently more than 1,700,000 species of animals, plants and fungi. Due to this enormous number it is impossible for any one to know more than tiny fraction of those species. Therefore it would be useful if a person could take a picture of a natural object by a camera and use some software to find out the correct species. Thanks to the recent development in mobile technology, mobile phones equipped with a camera and offering computing power sufficient for the recognition of natural objects from images are becoming ubiquitous.

Living organisms are divided into six taxonomic categories called kingdoms [7]. This work focuses on plants. We propose a method and implement an application for mobile phone for automatic identification of plants from pictures of their leaves and/or bark. Unlike moving animals plants are easy to photograph, they can be found everywhere and identification of significant number of species does not require any special equipment like a microscope or a DNA sequencer.

1.1 Motivation

Identification of natural things has been attracting people for centuries [4]. The oldest known attempt to rigorously describe the diversity of nature was made by an ancient Greek philosopher Theophrastus. The foundations of the modern binomial nomenclature and also of the modern taxonomy were laid by a Swedish botanist and zoologist Carl Linnaeus in the 18th century.

The traditional method of identification of living objects are identification keys. They describe each species by their most typical and significant features. The process requires several steps: at each the user specifies the object more precisely. It may take several minutes to find out the species, since the time consuming scrolling through the pages may be expected. Moreover the keys assume the knowledge of biological terminology, so the identification may be very difficult for non-expert users. Even professional biologists must sometimes do subjective decisions leading to uncertain

identification.

Despite the limitations of traditional keys there are lot of people interested in identification of living objects: tourists, teachers on primary and secondary schools and last but not least professional biologists and expert witnesses. All these groups might appreciate an application that would speed up the identification process.

1.2 Objective and requirements

The objective of this work is to design and implement an automated plant identification system for smart phones. The program should work real-time (or at least near real-time) and it should be easy to use even for non-expert users as well. In addition, it should be able to work offline, because availability of the internet connection still cannot be expected everywhere. Therefore all the computations must be performed directly on the device.

Expected functionality of the application is as follows: User takes the picture(s) of leaf or bark by an internal camera of the phone. As the images are taken in natural environment, no prior assumption about the background can be made. Despite that, the segmentation must be fast and user friendly. For example user may be asked to roughly mark several foreground and background pixels. The recognition should be fast as well and it would be also useful if the user could verify the results e.g. by browsing some additional description of the most likely species. Optionally, the user may provide additional information that make the recognition task easier and more accurate (specify the growth habit, dimensions of the plant etc.).

2 Related work

2.1 Identification keys

The traditional way of identification of plants are identification keys. They have usually the form of a decision tree. At each step, the user is supposed to choose from two (dichotomous keys) or more (polytomous keys) statements that describe the plant more and more precisely.

1 Plants without flowers (without perianth) and without fruits. Reproductive structures borne on the upper surface, lower surface or base of leaves, or in axillary or terminal cones. Ovules without carpels. (Ferns, horsetails, and gymnosperms and allied families.) **2**

1 Plants with flowers (with a perianth) and with fruits containing 1–many seeds. Ovules borne in carpels. (Flowering plants, or angiosperms.) **13**

2 Plants without a woody stem, not treelike in stature. No seeds present, only spores. (Lower vascular plants.) **3**

2 Plants with a woody stem, treelike in stature. Seed present, no spores. (Gymnosperms.) **10**

3 Stems usually with hollow internodes, simple or with whorled branches at the solid nodes. Stems segmented into internodes, can be readily pulled apart into segments, without leaves or needlelike scales. Stems hollow, fluted, or grooved but roundish in cross section. Leaves reduced to a whorl of minute, fused scales at each node. ***Horsetail Family*—EQUISETACEAE**

3 Stems not so segmented or fluted. Leaves various but not reduced to a whorl of fused scales at each node. **4**

Figure 2.1 Beginning of an identification key [6]

The identification process may take several minutes and it also expects the user is familiar with the botanical terminology.

Identification keys are not limited to printed books. United States Department of Agriculture provides an on-line identification key for selected groups of U.S. plants [5]. Unlike a traditional identification key these keys (the web pages contain several keys for each U.S. state) let user select multiple features simultaneously.

Another electronic guides can be found for example at http://www.botanicalkeys.co.uk/flora/ (Botanical Society of the British Isles), http://bioeco.free.fr/ (e.g. Clé de détermination de quelques familles à partir des fleurs), http://dbiodbs.univ.trieste.it/carso/

chiavi_pub21?sc=294 (Parcul dendrologic al Colegiului National Liviu Rebreanu din Bistrita, Romania.) and last but not least http://botanika.bf.jcu.cz/bryoweb/klic/ (Mechorosty České republiky).

2.2 Applications available at AppStore and Android market

Android market and AppStore provide plenty of applications such as plant encyclopedias, identifications keys and electronic textbooks.
The Landscaper's Companion is a mobile plant reference guide. It contains information about more than 2300 plants: common and scientific names, textual description, at least one high quality photo, cultivation tips and other. However it is not straightforward to use the application as an identification key, because user has no possibility to search the database by botanical features of the plants. There are similar applications for Android like Critter Browser (plants and animals), North American Birds, Animals Encyclopedia or Botanical Dictionary and also for phone: Landscapedia Garden Tour Guide, Audubon Wildflowers - A Field Guide to North American Wildflowers (this one even allows to search by botanical features) or Wildflowers.
Project Noah is an award-winning mobile application that helps nature lovers discover local wildlife and aspiring citizen scientists contribute to current research projects. The user is encouraged to create spottings: a picture, GPS coordinates and other information about the wildlife encounter. The spottings are uploaded to a server and shared with a community. Its members can make species suggestions and flag inaccurate or inappropriate content. Spottings can be freely viewed on the map allowing people to explore wildlife in their surroundings.
Tree ID is an application for phone. It is an implementation of a classic identification key. It allows to user identify almost any tree in Northern America based upon its name, scientific name, bark, leaf, fruit or location. Because the user may not know the proper botanical terminology, the application will walk him/her through the various methods of tree identification. Similar purpose have the Animal Kingdom - Top Animal and the Fungi Identification App for Android. The former focuses on animals whereas the latter on fungi (unfortunately it is no longer supported).

2.3 State-of-the-art of the automatic identification

2.3.1 Identification of plants from images of leaves

Wu et al. [19] employ probabilistic neural network to implement an automated leaf recognition for plant classification. Captured images are first converted into gray scale. Thanks to the assumption of white background the segmentation can be done by thresholding the histogram with a fixed threshold of 0.95. Leaves are described by 12 digital morphological features, derived from 5 basic features. The basic features are:

Diameter: The longest distance between any two points on the margin of the leaf. It is denoted as D.

Physiological Length: The distance between two of the main veins of the leaf. The terminals are marked manually. It is denoted as L_p.

Physiological Width: The length of the longest line orthogonal to the line joining two terminals marked by user. The length is defined as a distance between intersections of the line and leaf's boundary. It is denoted as W_p.

Leaf Area: Number of pixels belonging to the foreground. It is denoted as A.

Leaf Perimeter: Number of pixels consisting leaf margin. It is denoted as P.

The 12 features actually used to describe the leaf are as follows: Smooth factor (ratio between area of leaf images smoothed by 5×5 rectangular averaging filter and the one smoothed by 2×2 rectangular averaging filter), aspect ratio (L_p/W_p), form factor ($4\pi A/P^2$), rectangularity (L_pW_p/A), narrow factor (D/L_p), perimeter ratio of diameter (P/D), perimeter ratio of physiological length and physiological width ($P/(L_p+W_p)$ and five vein features ([19]). The dimension of the feature vector is further reduced by the PCA and the result is used as the input to the neural network classifier.

Zhang et al. [20] proposes to describe the leaves by geometry and texture features. Geometry features include ratio of the leaf length to leaf width and seven 2-D moments invariants [20]. In addition the Discrete Wavelet Transform is applied on the leaf image and statistical moments are used to extract the texture features at different scales [21]. The Kohonen SOM neural network is used as an classifier.

Wang et al. [13] addresses the problem of classification of leaf images with complicated

background. They propose to extract the leaf by the marker-controlled watershed segmentation method [15]. The initial markers are found automatically. First the Otsu's thresholding method [14] is used to obtain preliminary segmentation. Foreground pixels given by thresholding procedure usually correspond to target leaf and other interferents due to the complicated background. Therefore the erosion operation is applied to the binary image in order to separate area corresponding to target leaf from that corresponding to touching or covered interferents and each separated part can be regarded as the markers (the biggest part is considered as foreground whereas the others as background). The leaves are described by feature vectors containing seven Hu moments [16] and sixteen Zernike orthogonal moments [17]. The classification is done by the moving center hypersphere classifier [13].

Fiel [22] proposes to use SIFT ([23]) for leaf description. The image is first transformed into gray scale and normalized. The key-points are found and their neighbourhood described by the SIFT. The SIFT features are summarized by the bag-of-visual-words method [24] and multi-class SVM is used as an classifier (one-vs-all approach). Although images with simple background are expected the proposed method does not require any segmentation. In addition it is convenient for compound leaves. There is also no need for leafstalk removal or any other preprocession steps. The work also addresses problem of identification of plants from images of their bark and needles.

Unlike the previous approaches Rosatto et al. [18] classifies leaves according to their textural characteristics. They propose to describe a leaf using the volume fractal dimension. Feature vectors are then classified by a naive Bayes classifier. Their method unfortunately requires high-quality pictures (the leaves used in [18] were digitalized with a traditional scanner with a resolution of 1,200 dpi) which cannot be expected in case of internal mobile phones cameras.

To the best of our knowledge the most advanced system for automated recognition of plants is being developed by Belhumeur et al. [8]. Leaves are expected to be photographed on a plain white background, so the color-based EM algorithm [11] can be utilised for segmentation. In order to make the segmentation real-time only 5% of pixels are used. The shape of the leaf is encoded by the inner distance shape context ([9], [10], see also section 4.3). The best matching leaves are found by nearest neighbour algorithm. The search process is speeded up by utilisation of the approximating and

eliminating search algorithm (AESA [12]). Reportedly it reduces the time required to find the ten best matching species by a factor of 3. The whole system is designed to run on a mobile device like a tablet PC or even smaller ultra-mobile PC. It is currently in use by botanists at the Smithsonian Institution National Museum of Natural History to help catalogue and monitor plant species. Three datasets covering the flora of Plummers Island (an island in the Potomac River), all woody plants in published flora of the Baltimore-Washington, DC area and a nearly complete dataset of all the trees of Central Park in New York City have been available in 2008.

2.3.2 Identification of plants from images of bark

Zheru et al. [34] proposes to use two types of texture features for the bark classification task. The first is based on co-occurrence matrix ([2]) and the second is long connection length emphasis ([34]). The k-NN (k=1, 5, 8) classifiers are used for bark classification.

In [35] (Dahl et al.) the active appearance modelling (AAM) is used for bark classification. Three approaches are evaluated: in the first one an AAM based on the training images is built for each class. All models are matched to each of the test images giving model textures for all classes. The model texture is then compared to the original image by calculating the texture difference [35]. Classification is done by finding a model with the smallest texture difference. In the second approach only one AAM is built for all classes. The parameters from matching the model to a test image are then used for classification. The third approach is similar to the second one but here the alignment is used for extracting texture features.

2.3.3 Identification of plants from images of flowers

Saitoh et al. [25] uses four shape (the ratio of the average width over the average height, the number of petals, the roundness and the central moment) and six colour features to describe the flower. Before running the recognition algorithm the boundary of the flower is extracted by the intelligent scissors method [26]. It is an manual

method to draw a boundary based upon a number points on the visually identified boundary. In order to make the system more user-friendly the method is extended to an automatic one. Only in case the boundary is not extracted correctly the user can add several additional points to fix the error.

Pornpanomchai et al. [30] proposes to describe the flower by five features: actual flower ratio (ratio between number of foreground pixels over number of background pixels in the circumcircle), average values of the red, green and blue colour channels and number of edge pixels detected by Sobel edge detector. The species of the flowers are estimated by the nearest neighbour classifier.

In [31] (Cho et al.) the segmentation is done manually by the intelligent scissors method [26]. The flowers are described by 12 shape and 6 colour attributes. The shape feature extraction is based on the shape density distribution as well as the edge density distribution to extract the statistical attributes, such as mean, variance and entropy from the probability distribution functions. The color features are extracted from the HSV color model. They also propose to cluster the image database by the so-called virus infection clustering algorithm [31] to improve the searching efficiency.

Kim et al. [27] has created mobile-based flower recognition system. The user captures a picture, draws the boundary of the flower he/she is interested in and sends the image to the server. The contour features of the flower are extracted here: zero-crossing rate, the min distance, and contour line's length ([27]). The difference images are computed using pixel subtraction between the flower image and the normalized average flower images. Finally the flower's species is estimated using the minimum difference image entropy value and the results are sent back to the user.

Nilsback [28] adresses the problem of flower segmentation and classification. Segmentation process begins with the initial flower segmentation using general (non-class specific) prior foreground and background colour distributions. Given these preliminary results, a binary segmentation is obtained using the contrast dependent conditional random field (CRF) [29]. The generic flower shape model is then fitted to this initial segmentation in order to detect petals. The foreground (and similarly background) colour model is then updated by blending the image specific foreground model with the general foreground model and the CRF segmentation is repeated using the new colour models. Extracted flower is described by HSV values, MR8 filters, SIFT and

histogram of gradients (HOG). The features are represented using bag-of-visual-words method. The multi-class SVM is used as an classifier.

2.3.4 Recognition of other natural objects

A method for classification of animals is presented by Afkham et al. [32]. It utilises Markov random fields and bag-of-visual-words model to describe the animals (their texture) and SVM as an classifier. Another contribution of this paper is introduction of a novel image database containing 1239 images of animals captured at most possible natural conditions and variations.

Ramanan et al. [33] combines spatial and texture model for description of animals. The spatial model is learned from a video sequence.

3 Datasets

3.1 Flavia dataset

The Flavia dataset was introduced in [19] and contains 1907 images of leaves of 32 different plant species (50 to 77 images for each species). Each image contains exactly one leaflet: there are no compound or occluded leaves. All images have resolution 1600x1200 px. The dataset is publicly available at the following URL: http://flavia.sourceforge.net/.

	Common name	Images		Common name	Images
	Pubescent bamboo	59		Deodar	77
	Chinese horse chestnut	63		Ginkgo	62
	Anhui Barberry	65		Crape myrtle	61
	Chinese redbud	72		Oleander	66
	True indigo	73		Yew plum pine	60
	Japanese maple	56		Japanese Flowering Cherry	55
	Nanmu	62		Glossy Privet	55
	Castor aralia	52		Chinese Toon	65
	Chinese cinnamon	55		Peach	54

Table 3.1 The Flavia dataset (part 1)

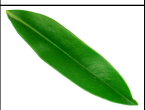
	Common name	Images		Common name	Images
	Goldenrain tree	59		Ford Woodlotus	52
	Big-fruited Holly	50		Trident maple	53
	Japanese cheesewood	63		Beale's barberry	55
	Wintersweet	51		Southern magnolia	57
	Camphor tree	64		Canadian poplar	64
	Japan Arrowwood	60		Chinese tulip tree	53
	Sweet osmanthus	56		Tangerine	56

Table 3.2 The Flavia dataset (part 2)

3.2 The Österreichische Bundesforste AG dataset

The dataset introduced in [22] was created by employees of the "Österreichische Bundesforste AG" in autumn 2009 and spring 2010. It consists of images of leaves, bark and needles. In this work only the bark dataset is used. It contains 1183 images of 11 tree species growing in Austria (7 to 204 images for each species). Images of the black pine, fir, hornbeam, larch, scots pine and spruce are further divided according to the age of the tree. All images have the same orientation but their scales differ.

	Common name	Images		Common name	Images
	Ash	24		Larch (> 80 years)	60
	Beech	7		Mountain oak	68
	Black pine (< 40 years)	54		Scots pine (< 40 years)	53
	Black pine (40 - 80 years)	53		Scots pine (40 - 80 years)	72
	Black pine (> 80 years)	50		Scots pine (> 80 years)	76
	Fir (< 60 years)	59		Spruce (< 40 years)	107
	Fir (> 60 years)	59		Spruce (40 - 80 years)	49
	Hornbeam (young)	4		Spruce (> 80 years)	48
	Hornbeam (middle)	18		Swiss stone pine (< 40 years)	15

Table 3.3 The Österreichische Bundesforste AG dataset (part 1)

	Common name	Images		Common name	Images
	Hornbeam (old)	11		Swiss stone pine (40 - 80 years)	38
	Larch (< 40 years)	50		Swiss stone pine (> 80 years)	34
	Larch (40 - 80 years)	81		Sycamore maple	12

Table 3.4 The Österreichische Bundesforste AG dataset (part 2)

4 Identification from leaves

4.1 Overview

The method proposed in this thesis is a four step process. First a picture of single leaflet is taken (by mobile phone's internal camera). Then the segmentation takes place. We use a semi-automatic approach based on the marker controlled watershed algorithm [36]. The user is asked to roughly mark several pixels from foreground and background and the leaf extracted from the background is shown instantly. If the user does not consider the result correct, additional markers can added until the segmentation is perfect. Extracted leaf is then represented by the inner distance shape context descriptor [10] and compared with the other leaves in the database. In the last step the most likely species are shown to the user.

4.2 Segmentation

The segmentation algorithm used in this work is based on a marker-controlled watershed algorithm [36]. A (gray-scale) image can be imagined as topographic surface: altitude of each point is determined by the gray level of corresponding pixel. Therefore dark areas in the image are considered valleys and catchment basins whereas the light areas mountains. Let us further imagine there are several holes (markers) in the surface and we take the surface and begin to slowly immerse it into a lake. When a hole reaches the water level the water will begin to flow through it and progressively fill up the catchment basin associated with the hole. At each pixel where the water coming from two different holes would merge we build a "dam". The catchment basins which do not contain a hole are filled up by overflowing of the neighboring catchment basin. When the water reaches the saddle point between both basins, it rushes through the pass and fills the so far empty basin. No dam is constructed between such basins. At the end of the immersion procedure the relief is divided into several regions each containing one hole.

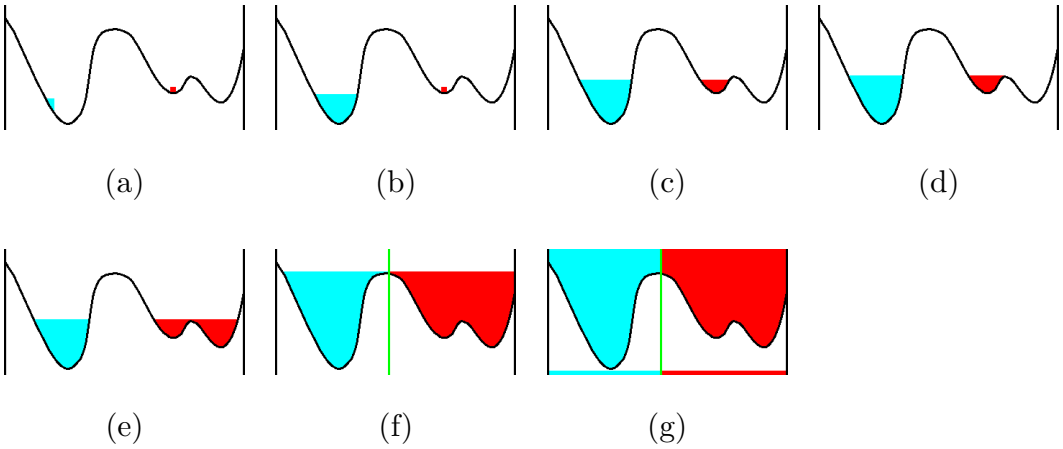


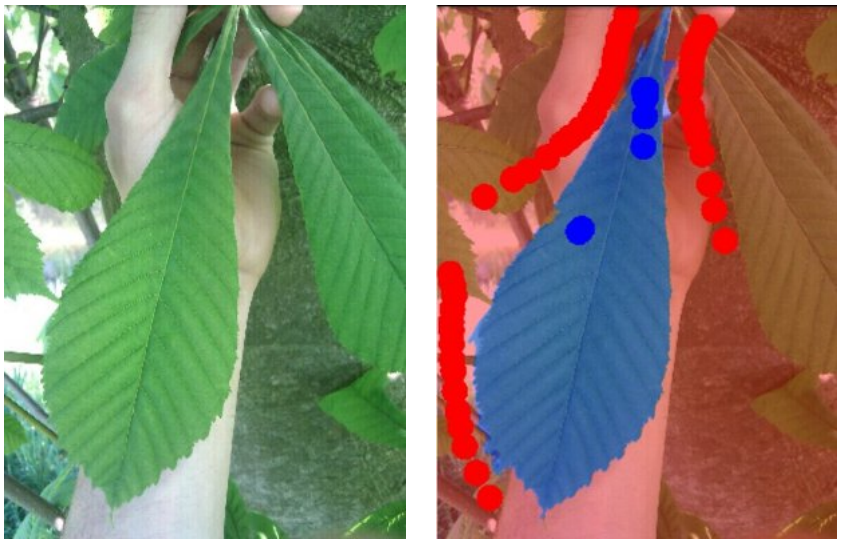
Figure 4.1 Example of the watershed segmentation (1D case). – a) Set of initial markers. b), c) Flooding catchment basins associated with the markers. d), e) Flooding the catchment basin neighbouring with the red region. f) Building a dam. g) Final segmentation.

Meyer [36] proposed the following watershed algorithm:

1. Compute gradient image of the input picture.
2. Chose markers. Marker is a set of pixels that have assigned the same label. Implementation of the algorithm used in this thesis (from openCV library [46]) puts no special requirements on the marker's shape, it may be even set of non-connected areas. There is no trivial rule for choosing the markers, so in this work they are selected manually by user.
3. The neighbouring pixels of each marked area are inserted into a priority queue. The pixels are ordered by their altitude (grey level in the gradient image): the lower altitude the higher priority they have.
4. A pixel with the highest priority is extracted from the queue. If it neighbours with pixels that are all marked by the same label, it is marked with that label as well. If it lies between two or more regions marked with different labels, it is marked as a border pixel between them (a dam). All non-marked neighbors of the extracted pixel that are not yet in the queue are put into the priority queue.
5. Redo step 4 until the priority queue is empty.

Figure 4.2 Marker-controlled watershed algorithm [36]

Cousty et al. [37] studied the watersheds in edge-weighted graphs and proposed a linear time algorithm (with respect to the number of vertices) to compute them.



(a) (b)

Figure 4.3 Watershed algorithm on real image (leaflet of Horse-chestnut). – a) Original image. b) Image after segmentation. The red (background) and blue (foreground) dots are provided manually by user.

4.3 Inner distance shape context (IDSC)

4.3.1 Shape context

Shape context descriptor was first introduced by Belongie et al. [9]. The object to be described is represented as a discrete set of points sampled from its internal and external contours. There is nothing special about these points, they do not correspond to key-points such as maxima of curvature or inflection points. The contours are rather sampled with uniform spacing. Each point can be now represented by a set of vectors originating from given point to all other sample points on the shape (for n sample points we get $n-1$ vectors for each point). However the full set of vectors would be too much detailed so Belongie proposed for each point p_i compute a coarse log-polar histogram h_i of the relative coordinates of the remaining $n-1$ points.

$$h_i(k) = \#\{p_j \neq p_i : (p_j - p_i) \in bin(k)\} \tag{4.1}$$

This histogram is defined to be the shape context of p_i. The descriptor is already invariant to translation. To achieve the scale invariance all radial distances are nor-

malized by the mean distance α between the n^2 point pairs in the shape. Rotation invariance can be achieved by treating the tangent vector at each point as the positive x-axis. In this way the reference frame turns with the tangent angle. Belongie tested the descriptor with 12 bins for angle, but in this work we use only 1. Such descriptor can not capture directions of vectors outgoing from a given point but at the same time it is rotationally invariant without necessity of computing the tangent vector. The resulting histogram is also smaller making the consecutive computations even faster.

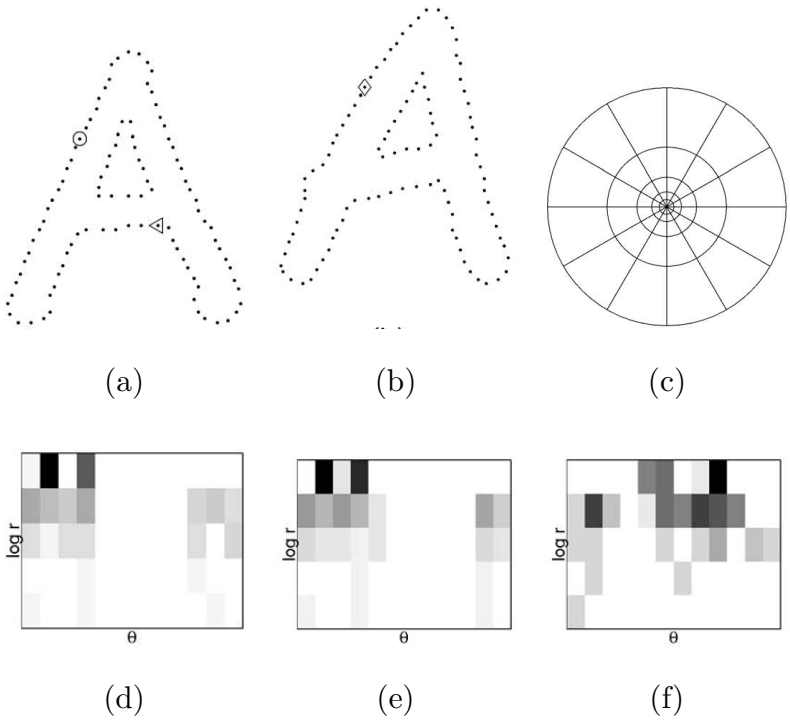


Figure 4.4 Example of shape context computation (picture taken from [9]). – a), b) Sampled edge points of two shapes. c) Diagram of log-polar histogram. d), e), f) Example shape contexts for reference samples marked by ∘, ⋄ and ◁.

4.3.2 Inner distance

Ling et al. [38] shown that some shapes are indistinguishable by histograms of Euclidean distances. Therefore he proposed to replace the distance function by so-called "inner distance". Inner distance between two points on contour of a shape is defined as "the length of the shortest path between these two points within the shape boundary"

[9]. It implies that the shape must consist of exactly one connected component which is not limiting for images of single leaflets.
Computation of the inner distance consists of two steps:

1. Build a graph with the sample points. All sample points are located on the outermost boundary of the shape and they are treated as vertices in the graph. For each pair of sample points p_1 and p_2, if the line segment connecting p_1 and p_2 falls entirely within the object, an edge between p_1 and p_2 is added to the graph with its weight equal to the Euclidean distance $||p_1 - p_2||$. The shape is approximated with a polygon formed by the points, hence neighboring boundary points are always connected.
2. Apply a shortest path algorithm to the graph. For example Floyd-Warshall algorithm finds the lengths of the shortest paths between all pairs of vertices in $O(n^3)$ time.

4.3.3 Shape matching

Let p_i be a point on the first shape, q_j on the second one and let $C_{ij} = C(p_i, q_j)$ denote the cost of matching these two points. As shape contexts are distributions represented as histograms, it is natural to use the χ^2 test statistic as a cost function:

$$C(p_i, q_j) = C_{i,j} = \frac{1}{2} \sum_{k=1}^{K} \frac{(h_i(k) - h_j(k))^2}{h_i(k) + h_j(k)}, \tag{4.2}$$

where K is number of bins and $h_i(k)$ and $h_j(k)$ denote k-th bin of the histogram associated with p_i and q_j respectively.
Given the set of costs C_{ij} between all pairs of points p_i on the first shape and q_j on the second shape, we want to minimize the total cost of matching:

$$H(\pi) = \sum_i C(p_i, q_{\pi(i)}) \tag{4.3}$$

subject to the constraint that the matching be one-to-one, i.e., π is a permutation. This would lead to weighted bipartite matching problem, which can be solved in $O(n^3)$ time. However, in our case we can put additional constraints on the permutation π

and solve the task more efficiently. Let us recall that our shapes consist of exactly one connected component and they are sampled only on their outermost boundary. Therefore the sample points from one shape can be treated as ordered sequence. In addition, orientation of both shapes can be determined (see section 4.3.4). Thus we assume the mapping of two shapes is approximately known and we just want to find "small" warping of one sequence that minimizes expression 4.3. This task can be solved by simplified Viterbi algorithm in $O(n)$ time.

$\mathcal{S}_1 := (p_0, p_1, p_2, \ldots, p_{n-1})$ // first sequence of points
$\mathcal{S}_2 := (q_0, q_1, q_2, \ldots, q_{n-1})$ // second sequence of points
$w :=$ maximum allowed warping (integer)
$best\_paths :=$ array$(n, 2w+1)$

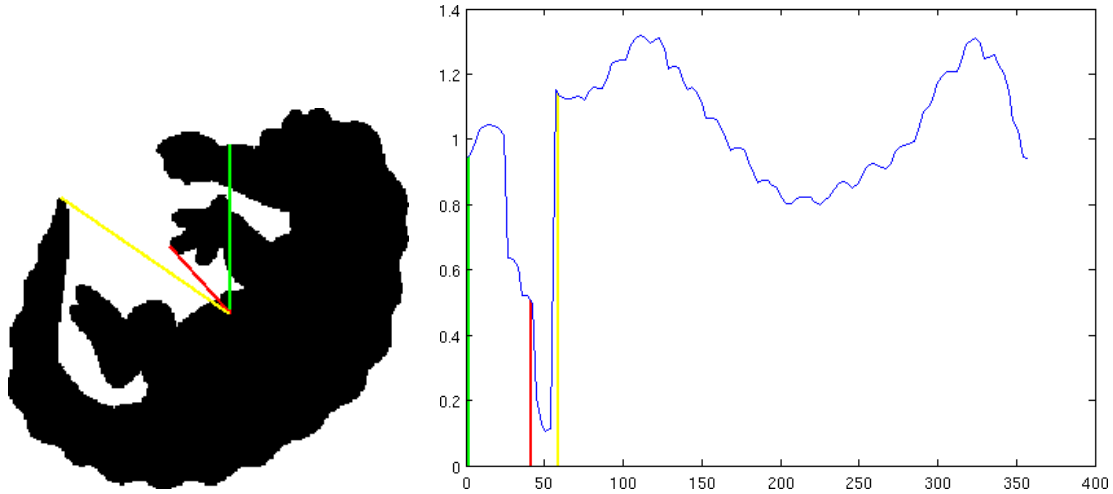
for $j = -w$ **to** w **do** $best\_paths(0, j+w) = C(p_0, q_{(0+j+n) \bmod n})$
for $i = 1$ **to** $n-1$ **do**
  for $j = -w$ **to** w **do**
    $best\_paths(i, j+w) = \min_{k \in \{-w..w\}} \{best\_paths(i-1, k) + C(p_i, q_{(i+k+n) \bmod n})\}$
  end
end
return $\min_k \{best\_paths(n-1, k)\}$ //total cost of matching

Figure 4.5 Simplified Viterbi algorithm

We got the best results for $w = 2$.

4.3.4 Finding shape orientation

Finding leaf orientation is important step in efficient shape matching. We have utilised centroid contour distance (CCD) curve [40] for that task. In this work, the centroid contour distance of the shape $\mathcal{S}$ at direction α is defined as distance between the center of gravity c (centroid) of $\mathcal{S}$ and a point p which is located on the outermost contour of $\mathcal{S}$ and angle between line cp and positive axis y is equal to α.



(a) (b)

Figure 4.6 Sample shape a) and its normalized ccd curve b) with three highlighted angles.

CCD curve $f(t)$ of each leaf is aligned with a pattern curve $g(t)$ in such way that their difference is minimal:

$$\alpha_{align} = \arg \min_{0 \leq \alpha < 2\pi} \int_0^{2\pi} (f(t+\alpha) - g(t))^2 dt \tag{4.4}$$

Let us note that both $f(t)$ and $g(t)$ are periodic functions so the expression inside the integral is always defined. In our case $g(t)$ corresponds to the general egg-shaped leaf (see section 6.1).

4.4 Speeding things up

Leaves of some species have such a unique shape that they can be distinguished from other species by even simpler method than IDSC. If we query the database with an unknown leaf reducing number of records that must be considered may significantly improve efficiency of the search.

Let X be a random variable which denotes distance of a pixel p from the center of gravity c and let $f(x)$ be its density. Binary image of a leaf can be treated as several (usually many) realizations of X. Therefore k-th raw moment of $f(x)$ can be estimated by appropriate sample moment

$$\mu_k = E(X^k) = \frac{1}{n}\sum_{i=1}^{n} ||x_i - c||^k \tag{4.5}$$

and similarly the k-th central moment:

$$\mu'_k = E((X - E(X))^k) = \frac{1}{n}\sum_{i=1}^{n} (||x_i - c|| - \mu_1)^k, \tag{4.6}$$

where n is overall number of pixels. The moments (both raw and central) are also random variables whereas sample moments are their realizations. Imagine for example we have a shape $\mathcal{S}$ generated by random variable X which is characterized by an arbitrary moment (for sake of clarity let us denote that moment Y). It follows from Chebyshev's inequality, that for any $\lambda > 0$ holds

$$Pr(|Y - E(Y)| \leq \lambda) \geq 1 - \frac{D(Y)}{\lambda^2} \tag{4.7}$$

Chebyshev's inequality can be utilised in the following way: Imagine a shape $\mathcal{S}$ for which we want to guess, whether it could be of class l (and whether class l should be considered during the classification). If so, one would expect, that its sample moments described above will fall into certain intervals associated with class l. If it holds for all moments, $\mathcal{S}$ might be of class l and otherwise if any of the moments falls outside of its expected interval, $\mathcal{S}$ is not of class l. If we denote $p = Pr(|Y - E(Y)| \leq \lambda)$ be the probability, that random variable Y will fall into the resulting interval, its bounds can be computed from Chebyshev's inequality as follows:

$$I = \left(E(Y) - \sqrt{\frac{D(Y)}{1-p}}, E(Y) + \sqrt{\frac{D(Y)}{1-p}}\right), \tag{4.8}$$

where the probability $p \in [0, 1)$ (common for all classes) is the only parameter to be chosen. Because mean values and variances are unknown, we replace them by their maximum likelihood estimations.

5 Identification from bark

5.1 Overview

Similarly to identification from leaves, identification from bark is also a four stage process. First the user takes a picture of the bark from the trunk. In addition, if the bark has oriented structure it is assumed the main furrows are approximately parallel with phone's y-axis. In the second step the user select a rectangle, where the bark is "good enough". If no rectangle is selected the whole image is used. The bark is then represented by uniform multiscale local binary patterns and compared with the other bark samples in the database. Finally the most likely species are shown to the user.

(a) (b) (c)

Figure 5.1 Sample images of bark). – a) Bark with oriented structure (European Alder). b), c) Bark with uncertain orientation (Common Beech, Siberian Larch)

5.2 Local binary patterns

Local binary patterns (LBP) [42] are widely used features in texture analysis. The basic version of the operator works with the eight-neighbours of a pixel, using the value of the center pixel as a threshold. An LBP code for a neighborhood is produced by multiplying the thresholded values with weights given to the corresponding pixels, and summing up the result (see figure 5.2). In this work we characterize texture by histogram of LBP codes in a given region of interest (whole image or a rectangle selected by user). It would lead to the 256-bin histogram with the basic LBP operator.

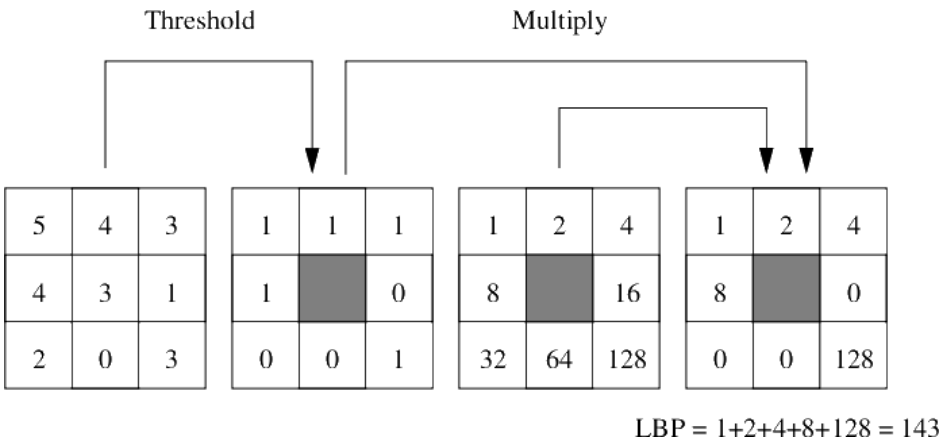


Figure 5.2 The original LBP operator (picture taken from [41])

Reportedly [42] vast majority of patterns that can be found at various textures have one property in common: they contain at most two one-to-zero or zero-to-one transitions in the circular binary code. These patterns are called "uniform". Because most of the information describing the texture is contained in the uniform patterns, the number of bins in the LBP histogram can be reduced to 59 (there are 58 distinct uniform patterns plus one bin for all non-uniform ones).

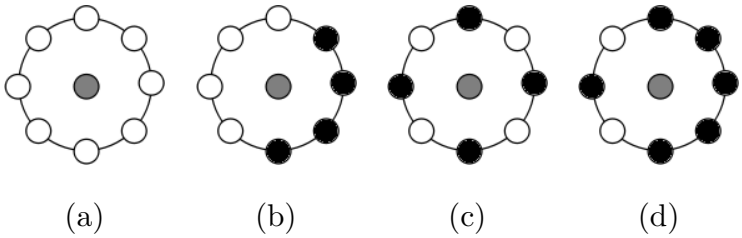
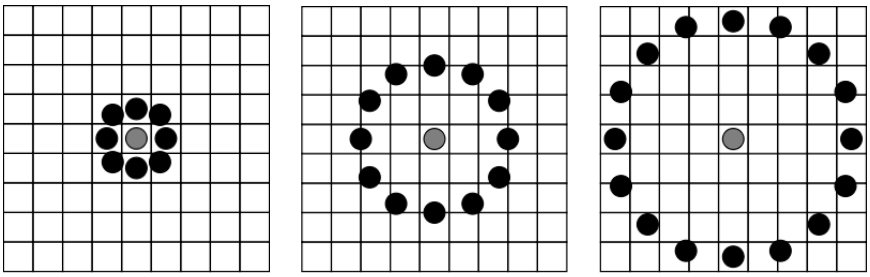


Figure 5.3 Example of uniform (a, b) and non-uniform patterns (c, d)).

In order to capture larger texture structures the operator can be further extended. Instead of 3x3 neighbourhood it may consider arbitrary circular neighbour sets [43]. The number of samples as well as the sampling radius can vary. For example figure 5.4 shows three different operators with various number of samples and radii of the neighbourhood. Samples that do not exactly fall on pixels are obtained with bilinear interpolation.



(a) (b) (c)

Figure 5.4 LBP operators with different number of samples and radii of the neighbourhood (picture taken from [43]). – a) 8 samples, radius = 1. b), 12 samples, radius = 2.5. c) 16 samples, radius = 4.

Due to aliasing effects and noise the LBP operator with large radius of the neighbourhood might not be an adequate representation of the texture. Therefore a low-pass Gaussian filter should be applied to the image in order to collect information from larger area (see [43]).

5.3 LBP on integral images

One of the advantages of the LBP operator is its low computational cost. However this is not true for LBP with large radius of neighbourhood. Due to the reasons we have mentioned in the previous section the image should be first filtered with a Gaussian kernel which is computationally intensive. One approach how to overcome this problem are so called multi-scale block LBP (MB-LBP)[44]. Unlike the original LBP the MB-LBP compares average gray values of square regions instead of single pixels. For example figure 5.5 shows the 9x9 MB-LBP operator: the average gray level of the central region is used as a threshold and the MB-LBP code is produced by multiplying the thresholded average gray levels of the remaining regions with corresponding weights and summing up the result.

Average gray level of arbitrarily large rectangular area can be computed in constant time using the integral image representation. Integral image has the same size as the original image and its value at any point (x, y) is just the sum of all the pixels above and to the left of (x, y), inclusive:

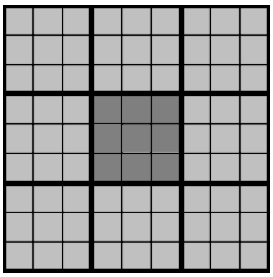


Figure 5.5 The 9x9 MB-LBP operator)

$$int(x, y) = \sum_{a \leq x, b \leq y} I(a, b) \tag{5.1}$$

where I is the original image. As shown in figure 5.6 computation of the average gray value of a rectangular area requires only 4 additions and one division. The whole MB-LBP thus requires $9 \cdot 4 = 36$ additions, 8 comparisons and if all the regions have not the same area, 9 divisions as well.

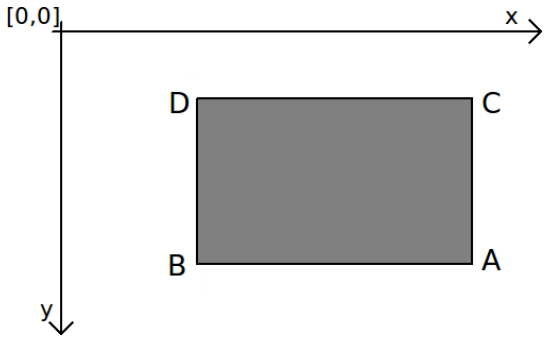


Figure 5.6 Integral image. Average gray value of the highlighted rectangle can be computed as $\frac{A-B-C+D}{\text{area}(ABCD)}$

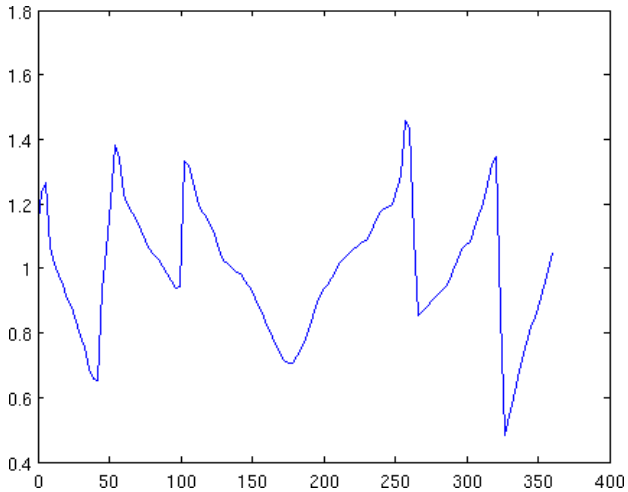
6 Results

6.1 Orientation of leaves

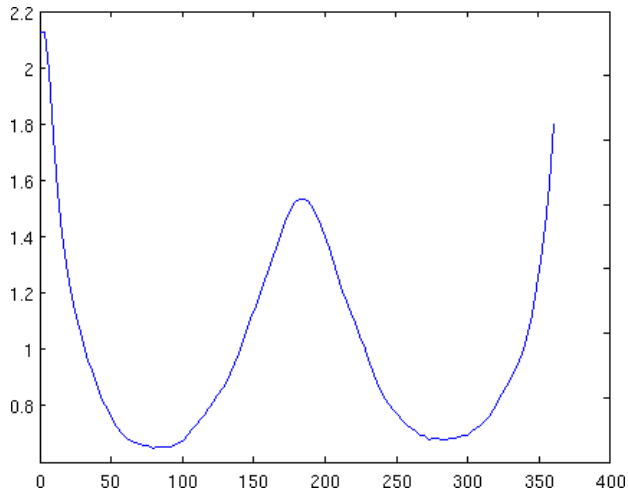
As described in section 4.3.2 comparison of two shapes can be done in $O(n)$ time (with respect to the number of sample points) if orientation of both shapes is known. Thus orientation of all the leaves in the database is determined using centroid contour distance curve method: CCD curve $f(t)$ of each leaf is aligned with a pattern CCD curve $g(t)$ such that their difference measured by L_2 norm is minimal (section 4.3.4). However it is not clear how to choose the $g(t)$. The chosen pattern curve should have the following properties:

1. There are only a few different orientations (preferably just a single one) assigned to the leaves from one class.
2. Small distortion or damage of the examined leaf will not cause significant change of its orientation.

We have tested two $g(t)$ corresponding to two main shapes of leaves (egg-shaped and maple-shaped):



(a)

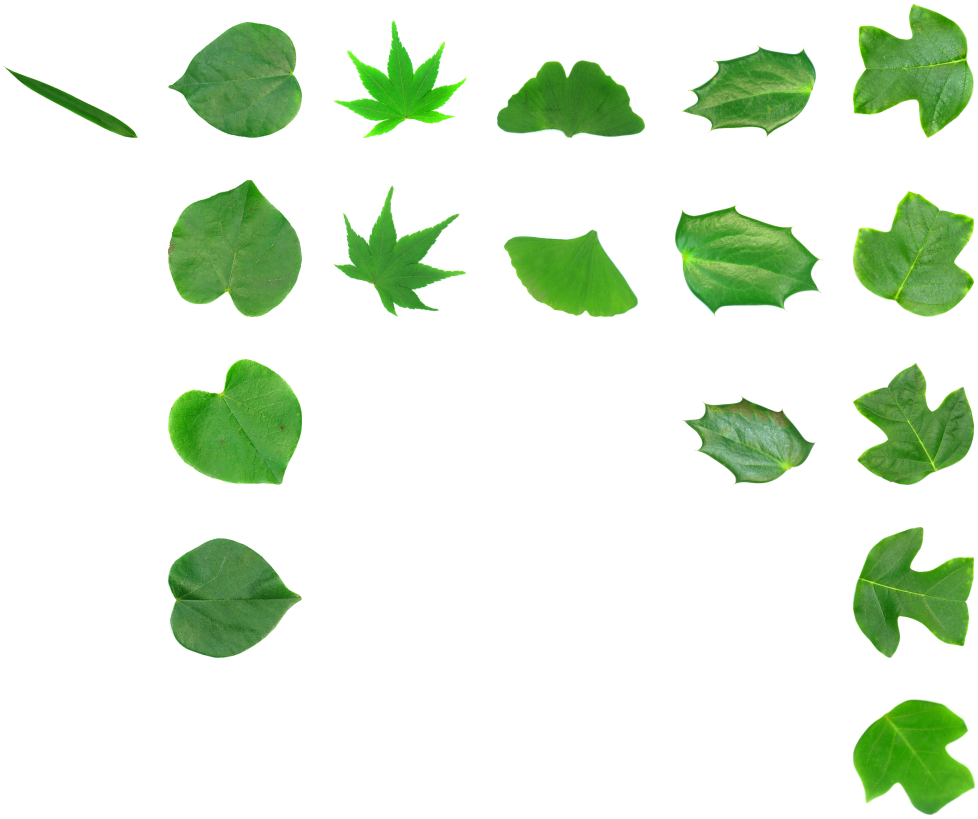


(b)

(c) (d)

Figure 6.1 Normalized pattern CCD curves and their corresponding leaves). a), c) Maple-shaped (Castor aralia). b), d) Egg-shaped (Japanese cheesewood).

Orientations assigned to several species from the Flavia dataset using the first $g(t)$ (6.1 a)):



(a) (b) (c) (d) (e) (f)

Figure 6.2 Orientations assigned to several species from the Flavia dataset using the first $g(t)$. a) Pubescent bamboo. b) Chinese redbud. c) Japanese maple. d) Ginkgo. e) Beale's barberry. f) Chinese tulip tree.

Orientations assigned to several species from the Flavia dataset using the second $g(t)$ (6.1 b)):

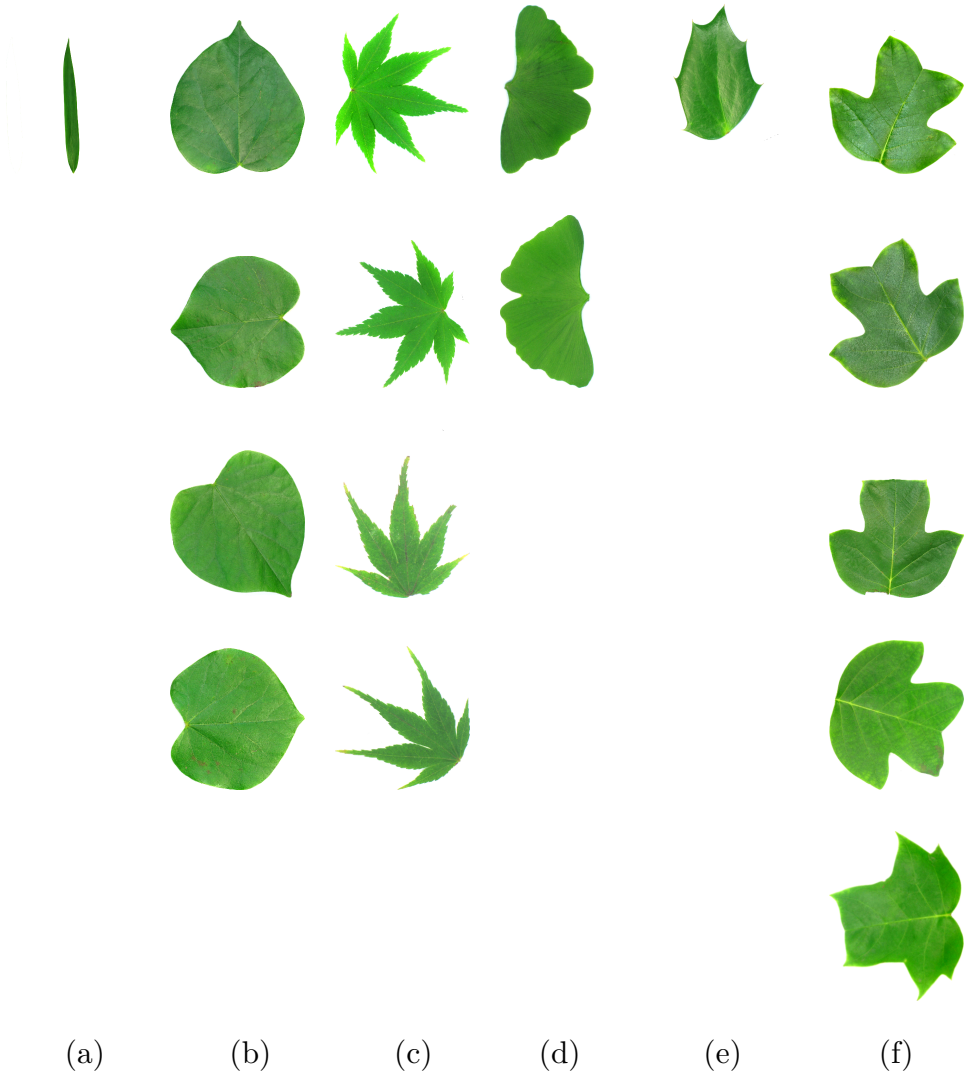


Figure 6.3 Orientations assigned to several species from the Flavia dataset using the second $g(t)$. a) Pubescent bamboo. b) Chinese redbud. c) Japanese maple. d) Ginkgo. e) Beale's barberry. f) Chinese tulip tree.

Surprisingly overall number of different orientations assigned to leaves of these six species is the same (although it differs within the species). We have further tested $g(t)$ corresponding to other species (Ginkgo, Japanese maple, etc.) but the results were the

same or even worse (more different orientations). The consequence for the database is as follows: if the classification has to be reliable, the database must contain each leaf rotated to several different angles given by local minima of the function measuring difference between $g(t)$ and $f(t)$ 4.4. Although it increases several times amount of data in the database (and consequently the time needed for the search), it is still more efficient than rotating the leaf to all possible angles during the matching.

6.2 Speeding up the recognition process

Time needed for recognition of a leaf may be reduced by the method described in section 4.4. It represents leaves by sample moments of their mass distribution function. A leaf might be of class l only if all its moments belong to certain intervals associated with the class l. These intervals are computed from Chebyshev's inequality 4.8: the lower p the smaller intervals but it is also more likely, that a distorted or damaged leaf will be marked as "certainly not of class l" even if l was its true class. In order to choose a reasonable p we have performed the following experiment:
The available data (Flavia dataset) was divided into two parts. The first set (training) was used to compute the intervals for 5 central and 5 raw sample moments. On the second set we have made two tests for each class: how many classes on average is excluded from further classification if the unknown leaf is of class l and how much percent of leaves of class l is marked as "certainly not class l". Despite the fact that p is probability we have evaluated values from range $[-5, 0.99]$ ($[0.01, 6]$ for $1-p$). The training set contained 953 leaves from 32 classes whereas the testing set contained 954 leaves (also from 32 classes).
The biggest $1-p$ for which no leaf from the testing set is misclassified is $1-p = 0.96$. For this value, on average 21.4% of classes is excluded from further classification. If the unknown leaf was of species Deodar, 81.9% of classes (more than 26 from 32 in absolute numbers) is excluded reducing the time required to find the correct species by factor of 5. Otherwise, for unknown leaf of species Japanese cheesewood, Camphor tree or Ginkgo only 5% of classes is not involved in further classification.
If misclassification of 1% of samples from the testing set was feasible (9 in absolute

numbers), on average 43.3% of classes would be excluded from further classification speeding up the process by factor of 1.76 ($1 - p = 2.61$).

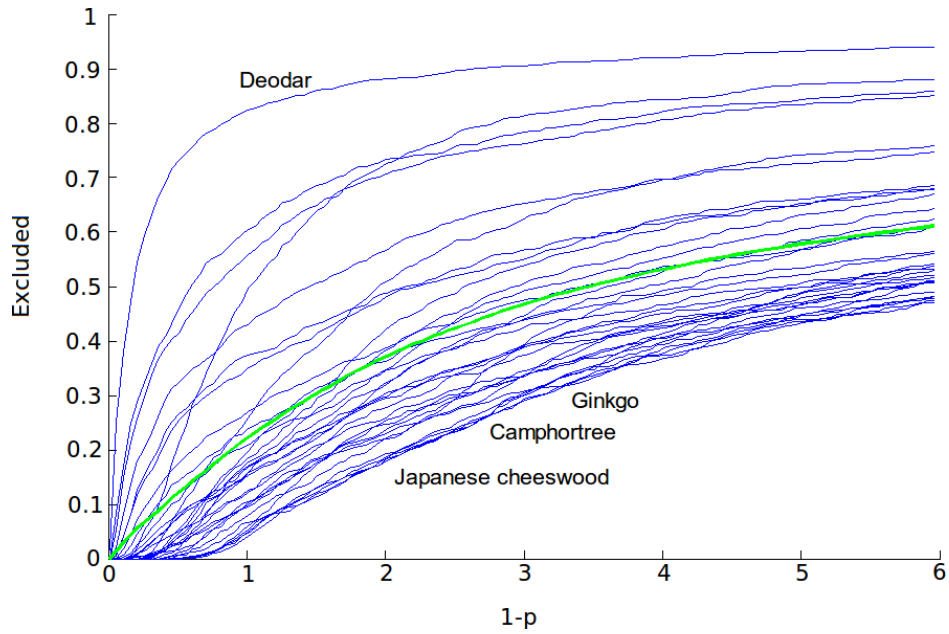


Figure 6.4 Average number of classes in percent excluded from further classification for each class (the thick green line is average over all classes)

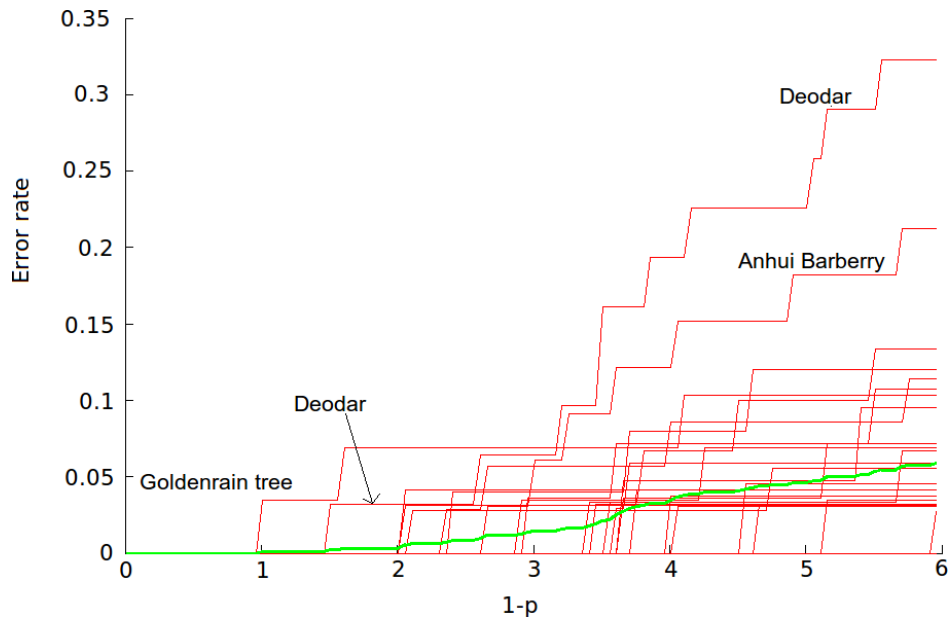


Figure 6.5 Number of misclassified samples in percent for each class (the thick green line is average over all classes)

6.3 Recognition from leaves

Classification accuracy on the Flavia dataset was tested with the following settings:

- The dataset was randomly split into two parts: training set (953 samples) and testing set (954 samples). Each set contained leaves from 32 classes.
- Egg-shaped pattern CCD curve (figure 6.1 b), d)) was used for determining orientation of the leaves. The CCD curves were sampled on 120 evenly spaced angles.
- In order to speed up the recognition process, 5 raw and 5 central sample moments of the mass distribution function were computed for each leaf. We have tested two values of $1 - p$: 0.96 and 2.61.
- Each leaf was described by 64 IDSC descriptors. The sample points were evenly spaced. The descriptors had 20 log-spaced bins for distance and 1 for angle (angles were not considered).
- The maximum allowed warping of two sequences was $w = 2$ (see section 4.3.2).

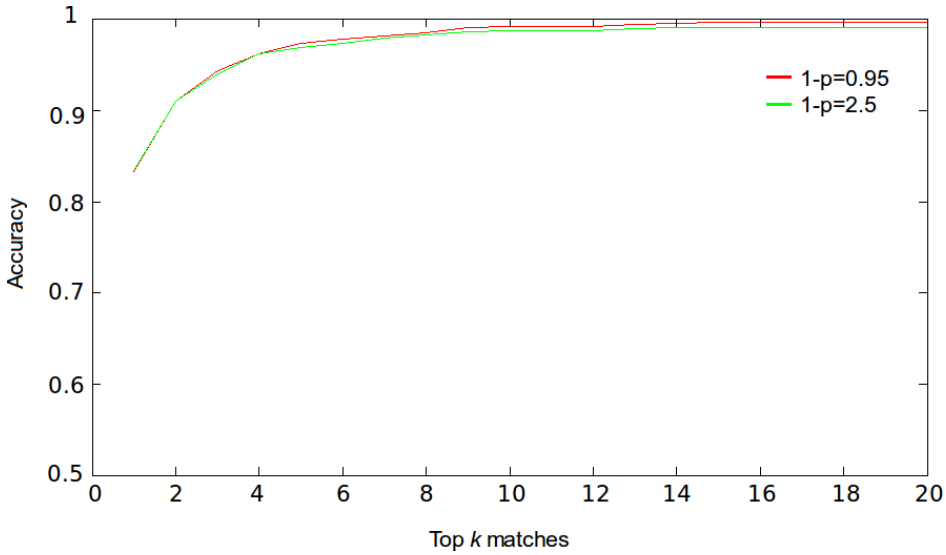


Figure 6.6 Classification accuracy for the Flavia dataset

If the correct answer was allowed to appear among the first one or two candidates, the recognition rates were surprisingly the same for both values of parameter $1 - p$: 83.3% and 91.0% respectively. The influence of different $1 - p$ is not apparent until 5 candidates are considered. In this case the recognition rate is 97.4% (929/954) for $1 - p = 0.96$ and 97.0% (925/954) for $1 - p = 2.61$. Therefore for practical purposes

might be feasible to set the parameter p to such value that for a small number of leaves their true class is excluded from the classification.

For $1-p = 0.96$ and only one permitted candidate, True indigo, Japanese maple, Castor aralia and Deodar are identified with 100% accuracy whereas the worst recognition rate had Chinese cinnamon (55.2%), Big-fruited Holly (61.9%), Camphor tree (61.1%) and Sweet osmanthus (56.3%). However if we consider 3 best candidates, their recognition rates become 96.6% (Big-fruited Holly), 90.5% (Chinese cinnamon), 77.8% (Camphor tree) and 87.6% (Sweet osmanthus). For example Sweet osmanthus is most often misclassified as Southern magnolia (6 samples from 32).

(a) (b)

Figure 6.7 An example of Sweet osmanthus a) misclassified as Southern magnolia b)

These two species are easily distinguishable by features that cannot be captured by 64 point IDSC: venation (and the texture at all), smoothness of the leaf margin and shape of the leaf blade tip. In addition they can be easily distinguished by bark: magnolia has gray smooth bark whereas sweet osmanthus often grows as a shrub with no bark[1].

[1] Sweet osmanthus can also grow as a small tree with gray smooth bark. In this case only the additional leaf features would be valid.

6.4 Recognition from bark

Classification accuracy on the Österreichische Bundesforste AG dataset was tested with the following settings:

- The dataset was randomly split into two parts: training set (542 samples) and testing set (539 samples). Each set contained images of bark from all available classes.
- As some species in the dataset are further divided according to the age of the tree we have tested the ability to recognize not only the species but also the age. In the first experiment each class in the dataset was treated independently (24 classes) whereas in the second experiment classes corresponding to one species were merged together (11 classes).
- Images from the testing set were described by three histograms of MB-LBP. Their dimensions were 9x9, 11.6x11.6 and 15x15 (dimensions increase by factor of 1.291). The second histogram was computed from the remaining ones by linear interpolation. The histograms were calculated only from uniform codes (therefore each histogram has 59 bins).
- Images from the training set were described by ten histograms of MB-LBP. Size of the first one was 9x9 and dimensions of each following MB-LBP increased by factor of 1.291. Histograms corresponding to MB-LBP with non-integer dimensions were computed by linear interpolation.
- Let h_{t_i} be the i-th histogram associated with an image T from the database and h_{q_j} be the j-th histogram associated with the query image Q. The distance between T and Q is defined as the best alignment of their histograms:

$$d(T,Q) = \min_{s \in \{0, n-m-1\}} \sum_{i=1}^{m} C(h_{t_{i+s}}, h_{q_i}), \tag{6.1}$$

  where n (m) is number of histograms associated with the image in the training set (with the query) and C is the χ^2 distance function:

$$C(h_{t_i}, h_{q_j}) = \frac{1}{2} \sum_{k=1}^{K} \frac{(h_{t_i}(k) - h_{q_j}(k))^2}{h_{t_i}(k) + h_{q_i}(k)}. \tag{6.2}$$

- The images were classified by nearest neighbour classifier.

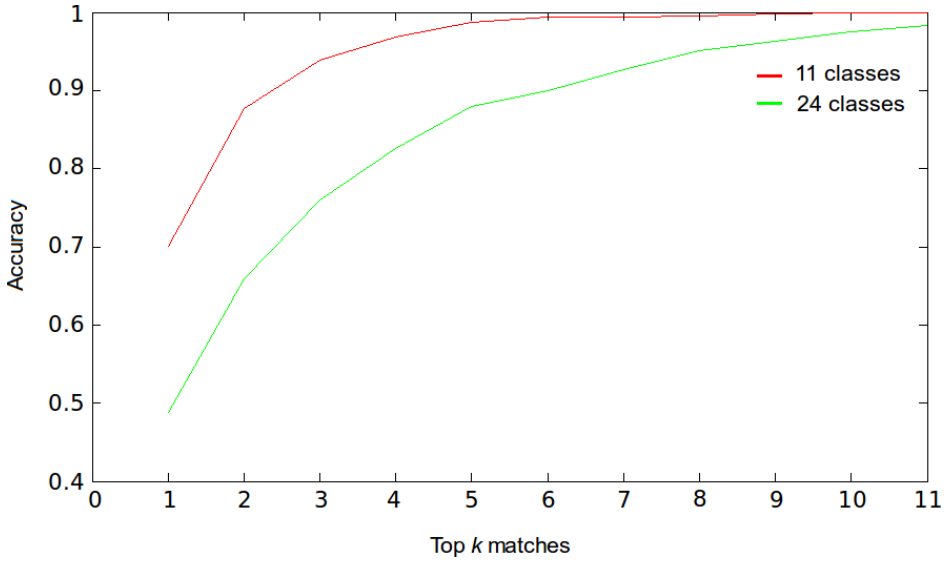


Figure 6.8 Classification accuracy for the Österreichische Bundesforste AG dataset

	Ash	Beech	Black pine	Fir	Hornbeam	Larch	Mountain oak	Scots pine	Spruce	S.s. pine	S. maple
Ash	**5**				1		5		1		
Beech		**3**									
Black pine			**61**			7		6		4	
Fir				**47**		7	1	1	3		
Hornbeam			1		**12**	1				2	
Larch			6			**71**		15	1	3	
Mountain oak					2	2	**26**	1	1	2	
Scots pine			6			31		**49**	1	4	
Spruce				2		2	3	6	**82**	6	
Swiss stone pine			2		2	7	5	6	2	**19**	
Sycamore maple									1	2	**3**

Table 6.1 Confusion matrix for 11 class experiment, rows are true classes, columns estimated

Recognition rates in 24 class experiment were 48.8%, 65.9% and 76.1% if the correct class was included in the top one, two or three candidates. Confusion matrix 6.2 shows

that identification of the species is more reliable than recognition of age. For example 61 images of Black pine from 78 in the testing set are correctly identified as Black pine whereas only for 27 the correct age is recognized as well. Recognition of age works better for Fir and Spruce: both age and species are correctly recognized at 36 from 58 images of Fir (62.1%) and at 66 from 101 images of Spruce (65.3%).

	Black pine			Fir		Horn-beam			Larch			Scots pine			Spruce			Swiss stone pine		
Black pine	**16**	4	2								2			1						
	7	**7**	7						2					2					2	
	3	11	**4**						3			1	2						2	
Fir				**19**	2				1	1	3			1			2			
				9	**17**						2				1					
Horn-beam						**6**	2				1									
			1				**3**											1		
						1		**0**											1	
Larch	1	1							**18**	2	5	2	2							
	2								1	**3**	11		3	2				2		
		2							3	10	**18**	1	5			1		1		
Scots pine			1						5	3	7	**11**		1						
		2							2	3	2	1	**12**	2		1			2	
	1	1	1							2	7	1	7	**14**					2	
Spruce															**46**	5	1			1
				1							1	1		2	2	**7**	4	1	2	2
				1							1	3			2	2	**13**			
Swiss stone pine		1	1			1			2	1	1			2	1			**1**	5	
							1		2		1		1	2				2	**8**	
														1		1				**3**

Table 6.2 Partial confusion matrix for 24 class experiment (trees with known age only), rows are true classes, columns estimated

Recognition rates in 11 class experiment were 70.1%, 87.8% and 93.9% if the correct class was included among the top one, two or three candidates. Beech was always correctly recognized however there were only three images of beech in the testing set so this recognition rate is not relevant. Tree with the best recognition recognition rate supported by sufficient number of images was Spruce: 81.2%. Otherwise, the worst

recognition rate was reported for Ash, only 47.7% of images were recognized correctly. There are several pairs of species with very similar bark. For example Larch is often misclassified as Scots pine and Ash as Mountain oak. Inter-class similarity of these species is demonstrated in figure 6.9.

(a) (b) (c) (d)

Figure 6.9 Trees with similar bark. a), b) Larch and Scots pine. c), d) Ash and Mountain oak

7 Application for mobile phone

Both methods for automatic identification of plants proposed in this thesis were implemented as an application for a mobile phone with the Android operation system. The program has been tested on the Nexus One smart-phone.

7.1 Available hardware

The Nexus One is equipped with a 1 GHz CPU with ARM architecture, it has 512 MB RAM and SSD card with 4 GB capacity. The operation system is the Android 2.2. The Nexus One is also equipped by a touch screen which helps user to mark the foreground and the background on the taken picture. The program is not limited to this particular device but can be run on any mobile device with the Android operation system.

Figure 7.1 The Nexus One

7.2 Android

Android is a software stack for mobile devices that includes an operating system, middleware and key applications [45]. The Android operating system is a multi-user Linux

system in which each application is a different user and live in its own security sandbox. By default, each application has access only to the components that it requires to do its work and no more.

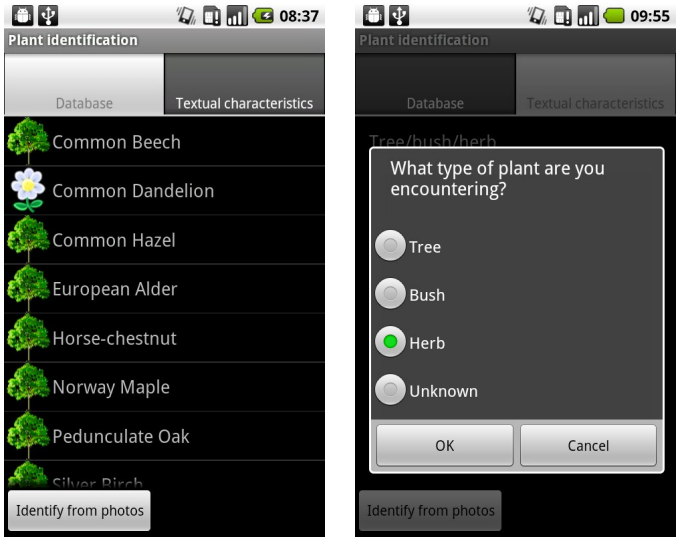
The Android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language. It allows the application to access camera, SSD card, GPS locator, network devices and provides tools to construct graphical user interface.

Android also allows to develop native applications. The Android NDK (native development kit) lets to build activities, handle user input, use hardware sensors, access application resources, etc., when programming in C/C++ programming language. However due to increased complexity developers are encouraged to write applications in Java if possible. In this work, the application logic and user interface are implemented in Java whereas the algorithms of computer vision in C++. We utilise openCV library [46] for some tasks (general image handling, segmentation). The native code is compiled by g++ to run on ARM architecture (cross-compilation). Java and native code are connected by Java native interface and SWIG (SWIG is a tool that generates the "glue code" required for the high-level languages to call into the C/C++ code).

7.3 User interface

The main window of the application contains two tabs: Database of the plants and Textual characteristics. Plants are displayed by their name and a small icon indicating its "form" (herb, bush or tree). If user taps on a plant name additional information are displayed on the screen (currently relevant web pages from Wikipedia).

Textual characteristics help user to reduce the number of displayed plants by answering questions that are easy for humans but difficult or even impossible for methods of computer vision (place of growth, shape of the stem etc.). In fact, this tab contains an interactive implementation of classic identification key. Only a few questions are displayed in the beginning. When the user answers some question, new questions (relevant for provided answer) appear in the list and simultaneously number of plants in the database decreases so only plants that fulfil provided answer remain in the list. Thus each question can be treated as a decision node in the identification key.



(a) (b)

Figure 7.2 Main window a), Database of plants. b), Sample question

Unlike the "paper" keys the application is revealing the questions dynamically and user see only relevant questions on the screen. In addition he/she can whenever change his/her decision for any question. Textual characteristics are not required for plant identification but they can make the process more accurate.

Button "Identify from photos" displays a screen where the user can select a feature to be photographed and consequently used for identification. Currently identification from leaves and bark is supported and we plan to add identification from flowers as well.

7.3.1 Identification from leaves

Identification from leaves works as follows: first, user taps on the button "Photo leaf" which invokes a Camera application[2]. When a photo is taken user is supposed to tap on the foreground and the background providing markers for the watershed segmentation algorithm. Segmentation is performed instantly and if the user is not satisfied with the result he/she can provide another markers until the segmentation is correct. User

[2] In this work we have utilised the default Camera application provided by Google Inc.

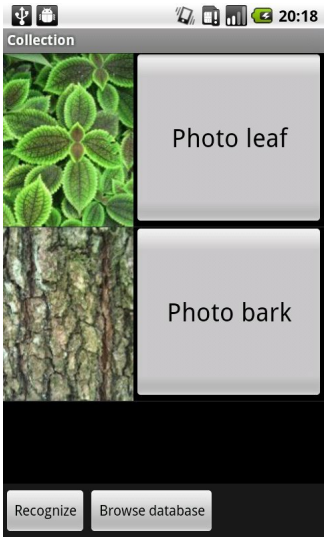


Figure 7.3 Features to be photographed (leaves and bark)

then pushes the "Confirm" button and at this point he/she has two options: he/she may continue in collecting photos (number of pictures of leaf or bark is not limited) or push the "Recognize" button. In this case the application uses all the taken and segmented photos to recognize plant the user is encountering. After a few seconds the reordered database of plants appears on the screen (the first item is the most likely plant).

7.3.2 Identification from bark

Identification from bark has the same work-flow as identification from leaves. The only difference is in the segmentation: instead of providing markers the user draws rectangle around region of the picture where the bark is "good enough".

7.4 Application performance

Recognition of one leaf takes with a database with 954 items (Flavia dataset) no more than 2 seconds. Computation of the leaf's description takes 0.5 seconds and the remaining time is spent on searching the database. Loading of the database into the memory takes 3 seconds but this task is done just once during the application's

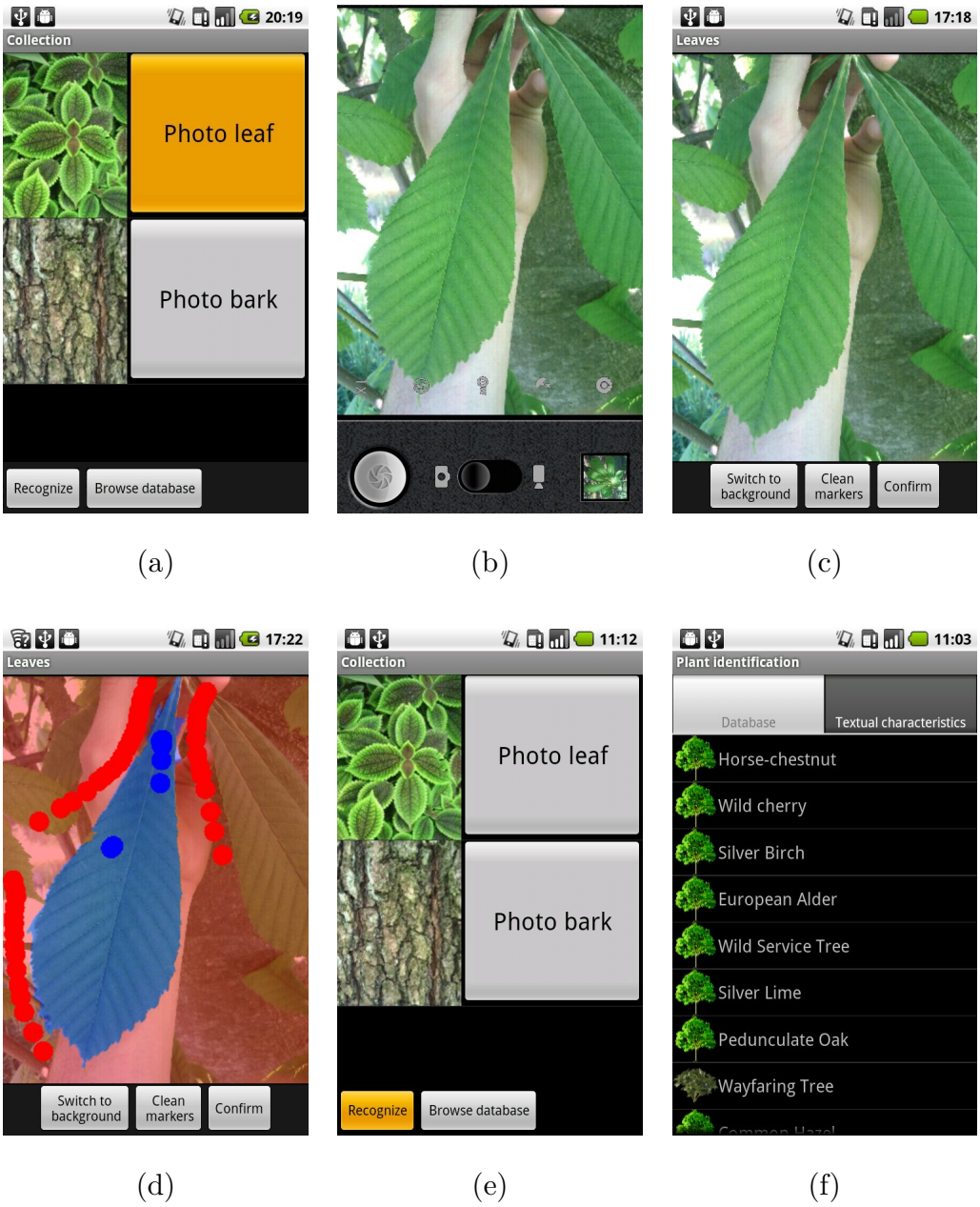
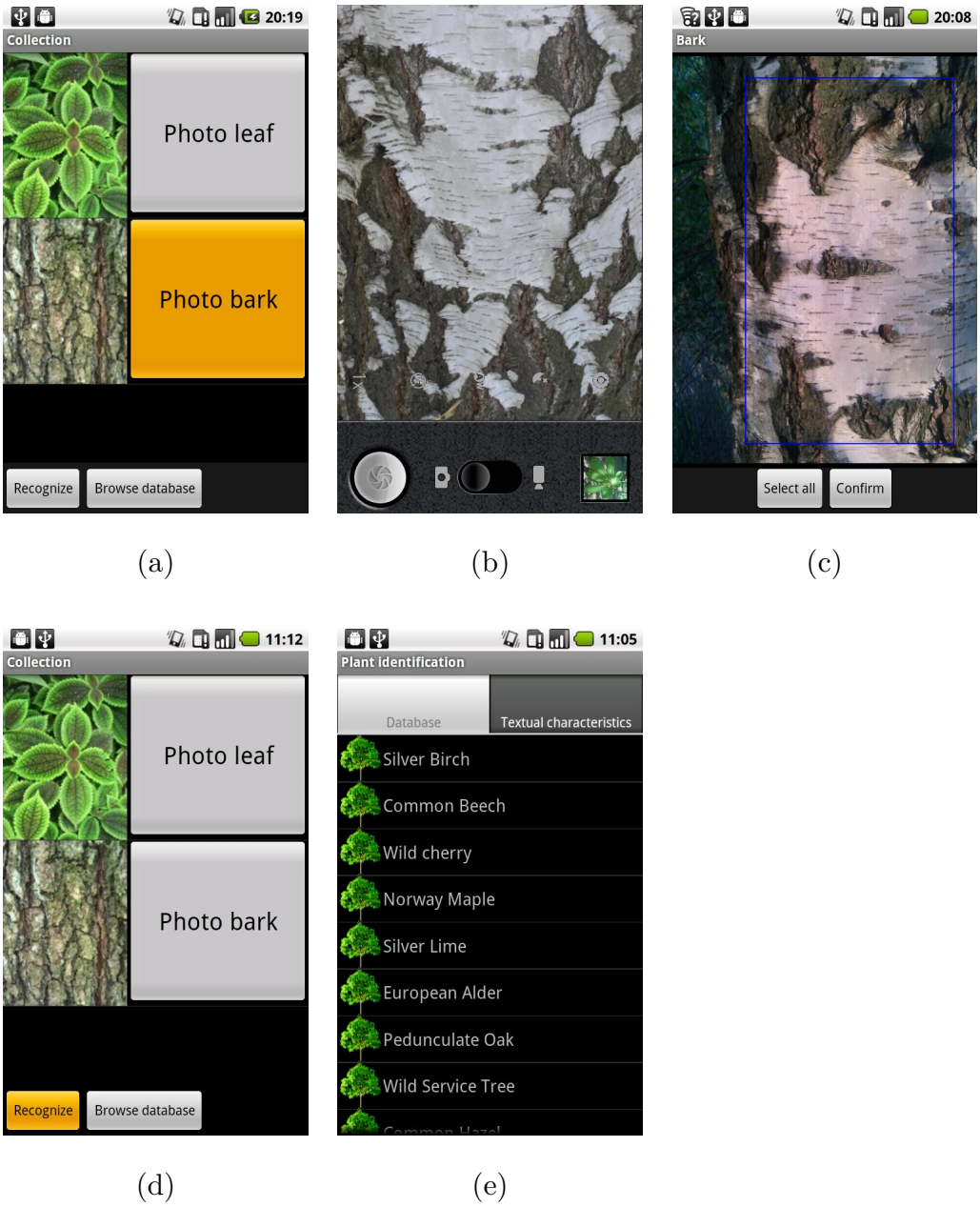


Figure 7.4 Identification from leaves - work-flow. a) Push "Photo leaf". b) Camera application. c) User acquired photo. d) After segmentation. e) Push "Recognize". f) Results.

start and it takes place in an independent thread so the user does not notice anything. The running time is further reduced by excluding some classes from the classification process (this method is described in section 4.4).

Recognition of one image of bark takes about 3 seconds with a database with 543 items (Österreichische Bundesforste AG dataset). However most of the time is spent on



(a) (b) (c)

(d) (e)

Figure 7.5 Identification from bark - work-flow. a) Push "Photo bark". b) Camera application. c) User acquired photo with drawn region of interest. d) Push "Recognize". e) Results.

computing the description of the query image so the database can be further extended without significant increase of run time.

7.5 Database of plants

We plan to create a database of trees, bush and herbs commonly growing in central Europe (subset of them). Until now we have collected pictures of leaves of 15 different species of trees and bush and pictures of bark of 18 species of trees. For 10 species of trees both leaves and bark was collected. However the database is currently incomplete and for some species it contains insufficient number of pictures. We are about to complete the database in the nearest future.

7.6 Typical users

7.6.1 Tourist, teacher on primary school

Some users require to identify a plant as quickly as possible. Therefore they quickly navigate to screen 7.3 and take a picture of leaf or bark. After the segmentation is finished they push the "Recognize" button (screen 7.3) and wait for the results. Because they are not very familiar with the botanical taxonomy they check out the additional information available for several most likely species in order to verify the application's output. The whole process (taking a picture, recognition and verification) does not take more than one minute.

7.6.2 Nature lover

This model user will appreciate both automatic identification and interactive key. Therefore the application is designed in such a way that switching between these two methods is fast. In addition taken pictures can be easily downloaded from SD card to user's computer. The photos are already categorized and for each photo the date, time and GPS position are known, so the user may whenever return to the place where he/she encountered the plant.

7.6.3 Professional botanist, expert witness

Users like professional botanists and expert witnesses do not identify plants as a hobby but it is their regular work they are responsible for. Therefore they will not rely on the automatic identification but use the interactive key (textual characteristics) instead. Any time during the navigation through the key they may switch to the database tab and check out the remaining plants.

8 Conclusions

In this thesis, we have presented two methods for automated identification of plants from pictures of leaves and from pictures of bark. Both methods are efficient enough to be performed on a mobile device. Recognition of one leaf does not take more than 2 seconds with database with 954 items and recognition of one image of bark takes about three seconds with database with 543 items. In addition time needed for recognition of leaf may be further reduced by excluding some classes from the classification.
Recognition of leaves works as follows: user takes a picture of a leaf and roughly marks foreground and the background. The markers are used as input to watershed segmentation algorithm. Segmentation is done instantly and if the user is not satisfied with its results he/she may provide another markers until the segmentation is correct. Orientation of the leaf is estimated by matching its centroid contour distance curve with a pattern CCD curve. This approach cannot determine the orientation unambiguously but it is able to reduce the number of possible ordinations to only a few angles. Leaves are represented by sequence of inner distance shape context histograms. Because their orientation is approximately known matching of two leaves can be done by dynamic programming in linear time.
In this work we impose two requirements on pictures of bark: they should be taken from trunk and they should preserve the natural bark orientation. If the whole image is filled with bark, no segmentation is needed, otherwise the user selects a rectangle in which the bark is "good enough". Bark is represented by three histograms of multi-block local binary patterns.

We have performed experiments on two datasets: The Flavia dataset (leaves [19]) and the Österreichische Bundesforste AG dataset (bark [22]). Recognition rates on the Flavia dataset were 83.3%, 91.0% and 94.3% if the correct class was included in the top one, two or three candidates. Recognition rates on the Österreichische Bundesforste AG dataset were 70.1%, 87.8% and 93.9% if the correct class was included in the top one, two or three candidates.
Described methods were implemented as an application for mobile phones with Android operation system. Although all the computations are performed directly on the

device the application is still real-time. Recognition of one leaf takes less than two seconds with database with 954 items and classification of one image of bark takes about three seconds with database with 543 items.

8.1 Future work

There are several possible directions of future work:

- The most often error made while using classic keys is identification of a plant which is not presented in the key. Therefore it would be helpful to show users confidence of the identification. If it was below a chosen threshold the application would warn user that the plant he/she is encountering is probably not in the database.
- Beside leaves and bark, plants may be also identified by another features like flowers, habitus etc. Therefore this work could be extended to handle these features. It would be also challenging to develop a general method for identification of plants from multiple features simultaneously. It should work properly when only one photo of one feature is available but it should work also when multiple pictures of multiple features are available and, in addition, each feature might be captured on whatever number of pictures.
- In terms of artificial intelligence, identification keys can be treated as decision trees. If proper source data was available (list of plants, each plant would have associated set of features) they could be generated automatically. Such keys could cover significant number of plants (e.g. all plants growing in Czech republic) which would make it useful even for professional botanists.
- Design and implement the graphical user interface that allows to view and manage taken photos, show position of encountered plant on the map, implement own Camera application etc.

9 References

[1] Forsyth D.A., Ponce J.. *Computer vision: A Modern Approach.* New Yersey, 2002.

[2] Šonka M., Hlaváč R.. *Image Processing, Analysis and Machine Vision.* Toronto, 2008.

[3] The World Conservation Union. *IUCN Red List of Threatened Species. Summary Statistics for Globally Threatened Species..*, 2010.

[4] Linneaus C.. *Systemae Naturae.* Impensis Direct. Laurentii Salvii, 1759.

[5] USDA, NRCS. *The PLANTS Database (http://plants.usda.gov, 19 April 2011).* National Plant Data Center, Baton Rouge, LA 70874-4490 USA, 2011.

[6] Becking W.R.. *Pocket flora of the redwood forest.* Island Press, 1982.

[7] Cavalier-Smith T.. *Only six kingdoms of life.* Proc. R. Soc. Lond. B 271 (1545): 1251–62, doi:10.1098/rspb.2004.2705, PMC 1691724, PMID 15306349, 2004.

[8] Belhumeur P.N., Chen D., Feiner S., Jacobs D.W., Kress W.J., Ling H., Lopez I., Ramamoorthi R., Sheorey S., White S., Zhang L.. *Searching the World's Herbaria: A System for Visual Identification of Plant Species.* In European Conf. on Computer Vision, LNCS vol. 5305: 116–129. Springer, 2008.

[9] Belongie S., Malik J., Puzicha J.. *Shape Matching and Object Recognition Using Shape Contexts.* IEEE Trans. on Patt. Anal. and Mach. Intell., 24(4): 509-522., 2002.

[10] Ling H., Jacobs D.. *Shape Classification Using the Inner-Distance.* IEEE Trans. on Patt. Anal. and Mach. Intell., 29(2): 286-299., 2009.

[11] Bilmes J.A.. *A Gentle Tutorial of the EM algorithm and its application to parameter estimation for Gaussian Mixed and Hidden Markov Models.* U.C. Berkely, California, April 1998, TR-97-021, 1998.

[12] Ruiz E.. *An algorithm for finding nearest neighbours in (approximately) constant average time..* Patt. Rec. Lett., 4(3): 145-157., 1986.

[13] Wang X.F., Huang D.S., Du J.X., Xu H., Heutte L.. *Classification of plant leaf images with complicated background.* Appl. Math. Comput. 205(2): 916–926, 2008.

[14] Otsu N.A.. *Threshold selection method from gray-level histograms.* IEEE Trans. System Man Cybernet. 9(1): 62–66., 1979.

[15] Beucher S. Meyer F.. *Morphological segmentation.* J. Visual Commun. Image Represent. 1(1): 21–46., 1990.

[16] Hu M.K.. *Visual pattern recognition by moment invariants.* IRE Trans. Inform. Theory 8: 179–187, 1962.

[17] Teague M.. *Image analysis via the general theory of moments.* J. Opt. Soc. Am. 70(8): 920–930., 1980.

[18] Rossatto D., Casanova D., Kolb R., Bruno O.. *Fractal analysis of leaf-texture properties as a tool for taxonomic and identification purposes: a case study with species from Neotropical Melastomataceae (Miconieae tribe).* Plant Systematics and Evolution 291(1): 103-116, 2011.

[19] Wu G.S., Bao F.S., Xu E.Y., Wang Y.X., Yi-Fan Chang Y.F., Xiang Q.L.. *A Leaf Recognition Algorithm for Plant classification Using Probabilistic Neural Network.* Signal Processing and Information Technology, 2007 IEEE International Symposium on.

[20] Zhang L., Kong J., Zeng X., Ren J.. *Plant Species Identification Based on Neural Network.* Natural Computation, 2008. ICNC '08. Fourth International Conference on.

[21] Arivazhagan S., Ganesan L.. *Texture classification using wavelet transform.* Pattern Recognition Letters, 24: 1513-1521, 2003.

[22] Fiel S., Sablatnig R.. *Automated identification of tree species from images of the bark, leaves or needles.* Technical Report, CVL-TR003, Computer Vision Lab, Institute of Computer Aided Automation, Vienna University of Technology, 2010.

[23] Lowe D.G.. *Object recognition from local scale-invariant features.* In Proc. Seventh IEEE International Conference on Computer Vision The, 2: 1150–1157, 1999.

[24] Csurka G., Dance C.R., Fan L., Willamowski J., Bray C.. *Visual categorization with bags of keypoints.* In Workshop on Statistical Learning in Computer Vision, ECCV, 1–22, 2004.

[25] Saitoh T., Aoki K., Kaneko T.. *Automatic Recognition of Blooming Flowers.* ICPR '04 Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04), 2004.

[26] Mortensen E.N., Barrett W.A.. *Intelligent scissors for image composition.* In Proc. of SIGGRAPH 1995, 191–198, 1995.

[27] Kim J.H., Huang R.G., Jin S.H., Hong K.S.. *Mobile-Based Flower Recognition System.* Intelligent Information Technology Application, 2009. IITA 2009. Third International Symposium on, 2009.

[28] Nilsback M.E.. *An automatic visual Flora – segmentation and classification of flower images (PhD thesis).* University of Oxford, United Kingdom, 2009.

[29] Boykov Y., Jolly M.P.. *Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images.* In Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada, 2: 105–112, 2001.

[30] Pornpanomchai C., Sakunreraratsame P., Wongsasirinart R., Youngtavichavhart N.. *Herb flower recognition system (HFRS).* Electronics and Information Engineering (ICEIE), 2010 International Conference On, 2010.

[31] Cho S.Y., Lim P.T.. *A novel Virus Infection Clustering for Flower Images Identification.* Pattern Recognition, 2006. ICPR 2006. 18th International Conference on.

[32] Afkham H.M., Targhi A.T., Eklundh J.-O., Pronobis A.. *Joint Visual Vocabulary For Animal Classification.* Pattern Recognition, 2008. ICPR 2008. 19th International Conference on.

[33] Ramanan D., Forsyth D.A., Barnard K.. *Detecting, Localizing and Recovering Kinematics of Textured Animals.* Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on.

[34] Zheru C., Li H., Wang C.. *Plant species recognition based on bark patterns using novel Gabor filter banks.* Neural Networks and Signal Processing, 2003. Proceedings of the 2003 International Conference on.

[35] Dahl A.B., Aanæs H., Larsen R., Ersbøll B.K.. *Classification of Biological Objects using Active Appearance Modelling and Color Cooccurrence Matrices.* SCIA'07 Proceedings of the 15th Scandinavian conference on Image analysis, 2007.

[36] Meyer F.. *Color image segmentation.* In Proceedings of the International Conference on Image Processing and its Applications, 303-306, 1992.

[37] Cousty J., Bertrand G., Najman L, Couprie M.. *Watershed Cuts: Minimum Spanning Forests and the Drop of Water Principle.* IEEE Transactions on Pattern Analysis and Machine Intelligence. 31(8): 1362–1374, 2009.

[38] Ling H., Jacobs D.. *Shape Classification Using the Inner-Distance.* IEEE Trans. on Patt. Anal. and Mach. Intell., 29(2):286-299., 2007.

[39] Saw J.G., Yang M.C.K., Mo T.C.. *Chebyshev inequality with estimated mean and variance.* Amer. Statistician, 38: 130-132, 1984.

[40] Wang Z., Chi W., Feng D.. *Shape based leaf image retrieval.* Vision, Image and Signal Processing, 150(1): 34-43, 2003.

[41] Mäenpää T.. *The local binary pattern approach to texture analysis — extensions and applications (PhD. thesis).* Infotech Oulu and Department of Electrical and Information Engineering,University of Oulu, 2003.

[42] Ojala T., Pietikäinen M., Harwood D.. *Performance evaluation of texture measures with classification based on Kullback discrimination of distributions.* Proceedings of the 12th IAPR International Conference on Pattern Recognition, 1: 582-585, 1994.

[43] Mäenpää T., Pietikäinen M.. *Multi-scale binary patterns for texture analysis.* SCIA2003.

[44] Liao S., Zhu X., Lei Z., Zhang L., S. S.. *Learning multi-scale block local binary patterns for face recognition.* in Advances in Biometrics, 828–837, 2007.

[45] Google Inc.. *Android developers.* http://developer.android.com, 2011.

[46] G. Bradski. *The OpenCV Library.* Dr. Dobb's Journal of Software Tools, 2000.