

Bachelor's Thesis

Recognition of Plants Based on Images of Fruit

Jan Kůrka



December 2013

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

Czech Technical University in Prague
Faculty of Electrical Engineering, Department of Cybernetics

Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Control Engineering

BACHELOR PROJECT ASSIGNMENT

Student: **Jan Kůrka**

Study programme: Cybernetics and robotics
Specialisation: Systems and Control

Title of Bachelor Project: **Recognition of plants based on images of fruit**

Guidelines:

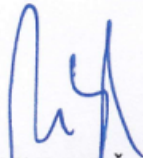
1. Familiarize yourself with the theses written by Tomáš Sixta and Milan Šulc as these will serve as your starting point.
2. Review the state-of-the-art in image-based plant recognition.
3. Choose a suitable method for fruit recognition.
4. Implement the chosen method and evaluate its quality, preferably on publicly available data.

Bibliography/Sources:


- [1] FORSYTH, David a Jean PONCE. Computer vision: a modern approach. New Jersey, 2002
- [2] ŠONKA, Milan, Václav HLAVÁČ a Roger BOYLE. Image processing, analysis, and machine vision, Toronto, 2008

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

Valid until the winter semester 2013/2014


prof. Ing. Michael Šebek, DrSc.
Head of Department




prof. Ing. Pavel Ripka, CSc.
Dean

Prague, December 20, 2012

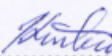
Acknowledgement

I would like to thank my supervisor, prof. Jiří Matas, who lead me through this project with great patience and encouragement. I am also grateful for the support of my family and friends.

Declaration

I declare that this theis is my own work and that I have listed all the literature and publications used in accordance with *Metodický pokyn č. 1/2009 - O dodržování etických principů při přípravě vysokoškolských závěrečných prací.*

Prague, December 31. 2013


.....

Abstract

Tato práce se zabývá poloautomatickým rozpoznáváním rostlin na základě fotografií plodů v přírodním prostředí. Prezentovaná metoda má dvě základní fáze: segmentaci a samotné rozpoznávání. Pro fázi segmentace byla navržena a zhodnocena poloautomatická metoda založená na kombinaci graph cut algoritmu a barevného histogramu. Ve fázi rozpoznávání je popředí zpracováno klasifikátorem na základě nejbližšího souseda. Jako příznaky jsou využity barevný histogram a poměr os elipsy vytvořené pomocí Direct Least Square Fitting metody [1].

Klíčová slova

rozpoznávání rostlin; rozpoznávání plodů; rozpoznávání na základě obrázku; segmentace; graph cut; barevný histogram

Abstract

This thesis deals with semi-automatic plant identification based on photographs of fruit in the natural environment. The method proposed in this thesis has two stages: segmentation and recognition. For the segmentation stage, a semi-automatic method based on a combination of graph cut and color histogram is presented and evaluated. In the recognition stage, the foreground is processed by a nearest neighbor classifier. As features, the color histogram and axes ratio of an ellipse fitted by Direct Least Square Fitting method [1] are used.

Keywords

plant recognition; fruit recognition; image-based recognition; segmentation; graph cut; color histogram

Contents

| | |
|--|-----------|
| 1. Introduction | 1 |
| 2. Segmentation | 3 |
| 2.1. State of the art | 3 |
| 2.1.1. Thresholding | 3 |
| 2.1.2. Marker-controlled watershed segmentation | 4 |
| 2.1.3. Graph cut | 4 |
| Energy function | 5 |
| Neighborhood links (n-links) | 5 |
| Terminal links (t-links) | 6 |
| Finding a minimum cut | 6 |
| 2.1.4. GrabCut | 6 |
| 2.2. The Proposed method | 7 |
| 2.2.1. Terminal links (t-links) | 7 |
| 2.2.2. Color space | 8 |
| 2.2.3. Histogram | 8 |
| 2.2.4. Constants | 8 |
| 2.3. The Experiments | 9 |
| 2.3.1. RGB histogram | 10 |
| 2.3.2. L*a*b* histogram | 12 |
| 2.3.3. k-Means histogram | 15 |
| 2.3.4. T-links | 18 |
| 2.4. Results | 19 |
| 3. Dataset | 20 |
| 4. Recognition | 24 |
| 4.1. State of the Art | 24 |
| 4.2. Recognition based on the color and shape features | 26 |
| 4.2.1. Metrics for histogram comparison | 27 |
| 4.3. Results | 28 |
| 4.3.1. Recognition rates for color feature | 28 |
| 4.3.2. Recognition rates for color and shape features | 30 |
| 5. Conclusion | 31 |
| 5.1. Future work | 31 |
| Appendices | |
| A. Contents of the attached DVD | 32 |
| Bibliography | 33 |

1. Introduction

People have always tried to label and identify plants that are all around us. Nowadays, we have detailed botanical literature, but using it to identify a plant could be very difficult and time consuming task for a non-expert. Even experts often have to rely on the literature, since there are over 300 000 plant species on Earth and they can know only a small portion of them.

These days, with smartphones and tablets being more common and affordable, we can use them for recognition tasks. Such recognition programs could help ordinary people quickly identify a plant or fruit. The experts could benefit from geotagging or other features of a mobile application, which could increase effectiveness greatly. Sixta [2] proposed a plant recognition based on leaves and bark images, and implemented it as an application for a smartphone running on Android operating system. Šulc [3] extended Sixta's work with coniferous branches identification and also developed Plant Ident application for Android operating system. Plant Ident is a user-friendly application allowing plant identification based on images of leaves, needles or bark and work as an electronic field guide too.

Another field where fruit recognition can be applied is the industrial sector. There were attempts to use it in point-of-sale (POS) systems at supermarkets to help the cashier identify the fruit, and in fruit sorting, quality check or on-tree localization of fruit, allowing automatic harvesting.

The goal of this thesis is to find a suitable method of semi-automatic plant recognition based on the images of fruit in natural environment, thus expanding the capabilities of computer-based plant identification and maybe also lay the groundwork for fruit recognition in the industrial sector. Expected input is a photo of a fruit, usually located in the center (see Figure 2). The proposed method consists of two main stages: semi-automatic segmentation, and recognition. The segmentation divides image into two segments: foreground (the fruit) and background. The foreground is processed by recognition algorithm and the result is a list of the candidates, from the most to the least probable. The flow of the process is in the Figure 1.

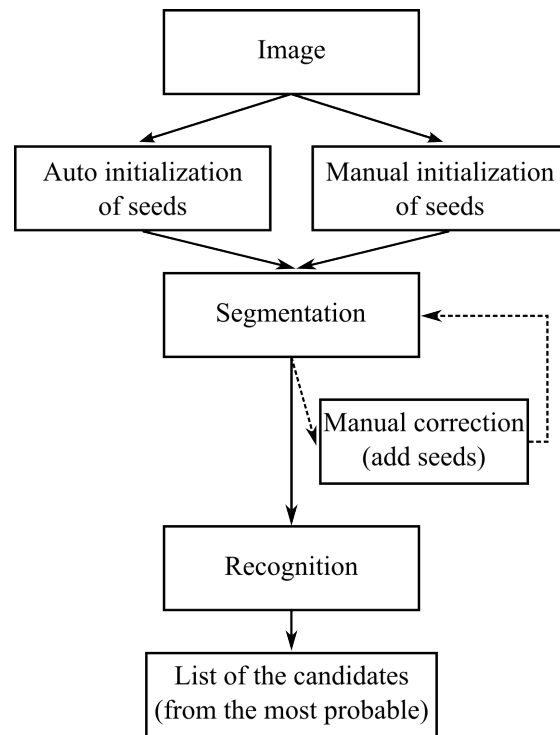


Figure 1. Flow of the recognition process

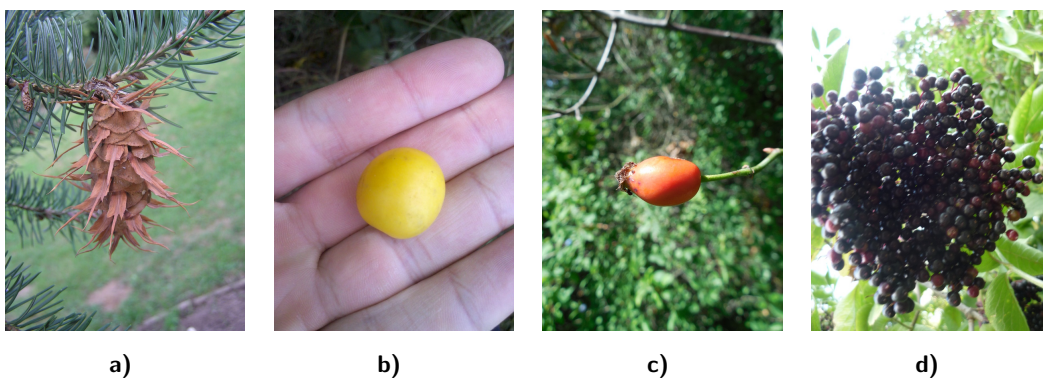


Figure 2. The examples of the input images

2. Segmentation

2.1. State of the art

The image segmentation is a division of the image into the several segments, in this case into the two segments – foreground and background. It is one of the most important steps towards an image analysis or recognition. It allows us to process only the part of image we are interested in (the foreground). There are many methods of image segmentation (e.g. the ones described in [4, page 175–320]). The following are implemented in the modern computer vision library, such as OpenCV¹.

2.1.1. Thresholding

Basic fixed-level thresholding is a simple segmentation process based on the a priori knowledge that the foreground pixels lie in a different range of values than the background pixels. We can therefore find threshold T which divides pixels into the two sets. This is typically used to get binary image g from a gray-level image f .

$$g(i, j) = \begin{cases} 1 & \text{if } f(i, j) > T \\ 0 & \text{if } f(i, j) \leq T \end{cases} \quad (1)$$

Threshold does not necessarily have to be predetermined. There are methods for threshold detection, usually based on histogram analysis.

If the scene in the image is inconsistently lit, the fixed-level thresholding may perform poorly. In these cases adaptive thresholding may work better because the threshold $T(i, j)$ is computed for each pixel separately, using $n \times n$ pixel neighborhood $N(i, j)$. As a threshold function,

$$T(i, j) = \left(\frac{1}{|N|} \sum_{p \in N(i, j)} p \right) - C \quad (2)$$

is usually used.

¹<http://opencv.org/>

2.1.2. Marker-controlled watershed segmentation

Watershed segmentation relies on watershed ridges and catchment basins. The input image is transformed to a gray-level gradient image. This image can be interpreted as a topographic surface where low-gradient regions (e.g. solid color regions) are catchment basins and high-gradient regions (e.g. edges) are watershed ridges. Set of initial markers helps prevent oversegmentation – not every basin is a separate region. We can then imagine filling the surface with water through the markers, same water level in all basins. The pixel where the water levels from different markers meet is a borderline.

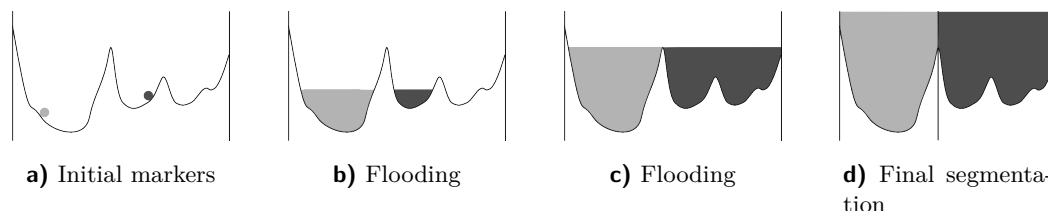


Figure 3. 1D example of a watershed segmentation

2.1.3. Graph cut

Graph-optimization algorithms as a powerful segmentation tool for N-dimensional images were presented for the first time by Boykov and Jolly in 2001 [5]. The graph cuts are used to find a globally optimal segmentation of the image.

Our goal is to divide an image into two segments: background and foreground. First the user is supposed to mark some pixels as hard constraints (seeds) that are definitely part of the foreground or background. For that reason, in our case, the user has to choose a rectangle – the region of interest (ROI) – inside which is the the whole foreground. Outside of the ROI, there is expected to be background, and it is marked as such. Then the user chooses few pixels of the foreground that are marked appropriately. After that, it is possible to start building the graph. The following graph cut algorithm is described according to the Boykov and Jolly [5].

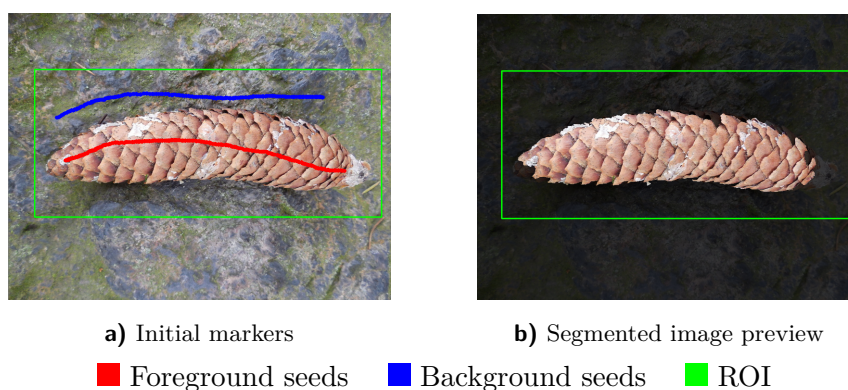


Figure 4. Example of a segmentation initialization and the result

Energy function

Let I denote the set of all image pixels \mathbf{p} and \mathbf{L} denote the labeling vector that assigns label to each pixel from I . Energy function $E(\mathbf{L})$ is a γ -weighted combination of a data term (a regional property term) $R(\mathbf{L})$ and a smoothness term (a boundary property term) $B(\mathbf{L})$. $R(\mathbf{L})$ is a cost of labeling pixels by \mathbf{L} , similarly $B(\mathbf{L})$ is a cost of discontinuity in labeling. Energy $E(\mathbf{L})$ is minimized to achieve optimal segmentation.

$$E(\mathbf{L}) = R(\mathbf{L}) + \gamma B(\mathbf{L}) \quad (3)$$

$$R(\mathbf{L}) = \sum_{\mathbf{p} \in I} R_p(L_p) \quad (4)$$

$$B(\mathbf{L}) = \sum_{\{\mathbf{p}_i, \mathbf{p}_j\} \in N} B_{i,j} \delta_{i,j} \quad (5)$$

$$\delta_{i,j} = \begin{cases} 1 & \text{if } L_i \neq L_j \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Neighborhood links (n-links)

The n-links connect pairs of neighboring pixels – each pixel has 6 neighbors. Let N denote the set of unordered neighboring pixels $\mathbf{p}_i, \mathbf{p}_j$, and \mathbf{w} denote the weight vector of individual color channels.

$$\Delta_{\mathbf{i},\mathbf{j}} = (\Delta_{i,j_k}); \quad k = 1, 2, 3 \quad (7)$$

$$\Delta_{i,j_k} = \sqrt{w_k} (p_{i_k} - p_{j_k}) \quad (8)$$

$$\sigma^2 = \frac{\sum_{\{\mathbf{p}_i, \mathbf{p}_j\} \in N} \Delta_{\mathbf{i},\mathbf{j}} \cdot \Delta_{\mathbf{i},\mathbf{j}}}{|N|} \quad (9)$$

$$\beta = \frac{1}{2\sigma^2} \quad (10)$$

$$B_{i,j} = \frac{1}{\sqrt{2}} e^{-\beta \Delta_{\mathbf{i},\mathbf{j}} \cdot \Delta_{\mathbf{i},\mathbf{j}}} \quad (11)$$

$B_{i,j}$ is a cost of n-link, which is high for small differences between pixels ($\Delta_{\mathbf{i},\mathbf{j}}$) and low for boundary regions.

Terminal links (t-links)

The t-links connect pixel nodes with terminal nodes (b - background node, f - foreground node) and its cost is determined by probability distribution or by a hard constraint. $P(p_i|fgd)$ and $P(p_i|bgd)$ denotes probability that a particular pixel p_i belongs either to the foreground (fgd) or the background (bgd).

$$R_i(b) = \begin{cases} K & \text{if } \mathbf{p}_i \text{ is marked as background} \\ 0 & \text{if } \mathbf{p}_i \text{ is marked as foreground} \\ -\ln [P(\mathbf{p}_i|bgd)] & \text{otherwise} \end{cases} \quad (12)$$

$$R_i(f) = \begin{cases} 0 & \text{if } \mathbf{p}_i \text{ is marked as background} \\ K & \text{if } \mathbf{p}_i \text{ is marked as foreground} \\ -\ln [P(\mathbf{p}_i|fgd)] & \text{otherwise} \end{cases} \quad (13)$$

$$K = 1 + \max_{\mathbf{p}_i \in I} \sum_{\mathbf{p}_j: \{\mathbf{p}_i, \mathbf{p}_j\} \in N} B_{i,j} \quad (14)$$

Finding a minimum cut

Minimizing energy function means finding a minimum cut (min-cut) in a graph. The cost of a cut is a sum of costs of all arcs in the cut and min-cut is the one with the lowest cost. There is a dual problem to the min-cut: maximum flow (max-flow) - which enables us to get solution more easily. Both min-cut and max-flow are well known combinatorial problems that are solved by many polynomial-time algorithms, e.g. Goldberg-style "push relabel" [6] or by augmenting path algorithms such as Ford-Fulkersons [7]. In this case, I used a modified OpenCV GrabCut algorithm based on Boykov and Kolmogorov augmenting path algorithm [8].

2.1.4. GrabCut

GrabCut is an algorithm that uses iterated graph cuts (see section 2.1.3). It is an efficient tool for interactive foreground segmentation, and, compared to watershed segmentation, gives more accurate results. In less difficult cases, it is sufficient to just mark the object with a rectangle, in the more difficult ones, it is necessary to add some seeds. GrabCut is computationally more demanding than watershed segmentation, and is therefore less suitable for real-time or near-real-time processing.

2.2. The Proposed method

For an application in fruit segmentation, I chose graph cut segmentation in combination with color histograms to avoid the time complexity of GrabCut at least partially. Graph cut is suitable for interactive segmentation because it can be efficiently recomputed when the user changes or adds seeds. The following settings and modifications of the standard graph cut (described in section 2.1.3) were applied:

2.2.1. Terminal links (t-links)

The cost of t-links is determined by color histograms or by a hard constraint. There are separate histograms for the foreground H_f and for the background H_b . The cost of pixels without hard constraints is calculated by using values from foreground histogram k_{fi} and background histogram k_{bi} for a given pixel \mathbf{p}_i . The expression in the logarithm can be interpreted as a relative frequency of the given color of \mathbf{p}_i .

Two modifications of the t-links were considered: R, R' . In R' including the "1" into the numerator results in preferring the label with smaller set of seeds.

$$R_i(b) = \begin{cases} \lambda & \text{if } \mathbf{p}_i \text{ is marked as background} \\ 0 & \text{if } \mathbf{p}_i \text{ is marked as foreground} \\ -\ln\left(\frac{k_{fi}}{\|H_f\|} + 1\right) & \text{otherwise} \end{cases} \quad (15)$$

$$R_i(f) = \begin{cases} 0 & \text{if } \mathbf{p}_i \text{ is marked as background} \\ \lambda & \text{if } \mathbf{p}_i \text{ is marked as foreground} \\ -\ln\left(\frac{k_{bi}}{\|H_b\|} + 1\right) & \text{otherwise} \end{cases} \quad (16)$$

$$\|H\| = \sqrt{\sum_{k_i \in H} k_i^2}. \quad (17)$$

$$R'_i(b) = \begin{cases} \lambda & \text{if } \mathbf{p}_i \text{ is marked as background} \\ 0 & \text{if } \mathbf{p}_i \text{ is marked as foreground} \\ -\ln\left(\frac{k_{fi} + 1}{\|H'_f\|}\right) & \text{otherwise} \end{cases} \quad (18)$$

$$R'_i(f) = \begin{cases} 0 & \text{if } \mathbf{p}_i \text{ is marked as background} \\ \lambda & \text{if } \mathbf{p}_i \text{ is marked as foreground} \\ -\ln\left(\frac{k_{bi} + 1}{\|H'_b\|}\right) & \text{otherwise} \end{cases} \quad (19)$$

$$\|H'\| = \sqrt{\sum_{k_i \in H} (k_i + 1)^2}. \quad (20)$$

$$\lambda = 9\gamma \tag{21}$$

2.2.2. Color space

I picked and evaluated two different color spaces: RGB and CIE L*a*b*.

RGB is a linear color space that uses three primary colors (red, green and blue). The value of a color is represented as a three-element vector that informs about intensities of the three primary colors. The disadvantage of RGB is that all channels include not only hue, but also brightness information. It means that all channels are affected by the shift in brightness often caused by illumination change or different camera setting.

L*a*b* is a uniform color space based on CIE XYZ color space. L dimension represents lightness and a, b dimensions represent color-opponent dimensions. Forsyth and Ponce stated that the distance in uniform color space is a fair indicator of the difference between two colors perceived by the human eye [9, page 112].

2.2.3. Histogram

I evaluated uniform histograms with different bin sizes and histograms with bin centers computed using k-means clustering algorithm. Clustering was applied to 90 randomly chosen images from the dataset and the histogram was then computed by assigning image pixels to the nearest center. To speed the assigning up, the $64 \times 64 \times 64$ size matrix, similar to uniform histogram, was created. Each element of it contains reference to the nearest center for this particular bin.

2.2.4. Constants

The γ constant determines the weight of boundary property term $B(\mathbf{L})$ in comparison to regional property term $R(\mathbf{L})$.

Vector \mathbf{w} determines the weight of the individual color space channels. In RGB, the channels are usually equally important. In L*a*b*, the L (lightness) channel represents different quality than the other two, therefore it could also be a different contribution to the segmentation.

2.3. The Experiments

The choice of constants, color space and the histogram type is essential to get an effective and well-functioning algorithm. The following settings were evaluated and tuned:

- Color space: RGB or CIE L*a*b*
- Histogram type: Uniform or k-means
- T-links: R or R'
- Constants: γ , \mathbf{w}

The algorithm with each setting was run on the dataset (see section 3) with automatic initialization – the 10 pixel border was marked as background and the circle in the middle with 10 pixel radius was marked as foreground. The result was compared with the correctly segmented images (the ground truth) giving the number of incorrectly labeled pixels. The time elapsed computing different parts of algorithm was also evaluated because the time complexity of the task is very important, especially for real-time and near-real-time applications.

The following graphs show the dependence of the segmentation error, displayed as an average percentage of mislabeled pixels, and the time elapsed when computing the segmentation estimation. The graph and histogram calculation times are in the Table 1.

| Histogram type | Average time of a histogram calculation | Average time of a graph construction |
|--|---|--------------------------------------|
| RGB histogram | 2 ms | 372 ms |
| LAB histogram | 16 ms | 434 ms |
| k-Means histogram, $k = 150$ | 58 ms | 867 ms |
| k-Means histogram, $k = 250$ | 77 ms | 1155 ms |
| k-Means histogram, $k = 500$ | 126 ms | 1910 ms |
| k-Means histogram, $k = 1000$ | 219 ms | 3384 ms |
| k-Means histogram with a reference matrix | 18 ms | 530 ms |

Table 1. Average elapsed times of the graph and histogram calculations.

- Average segmentation error
- Average elapsed time of segmentation estimation

Figure 5. Legend for the graphs in Figures 6–15

2.3.1. RGB histogram

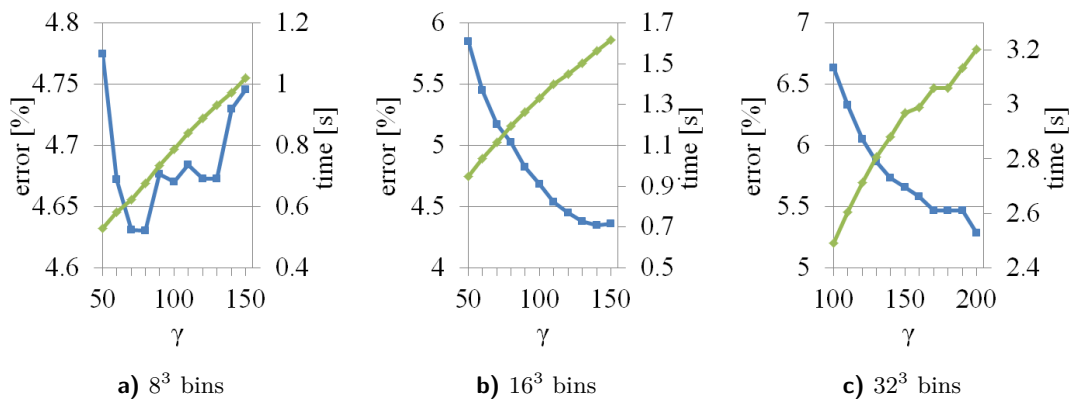


Figure 6. RGB histogram results, t-links: R' . For legend, see Figure 5.

From the Figure 6, we can see that the best performing histogram is the one with 16^3 bins, $\gamma = 140$ and t-links R' . However, with this setting, the algorithm is quite slow, therefore the best setting for near-real-time purposes would be $\gamma = 70$ 8^3 bins and t-links R' .

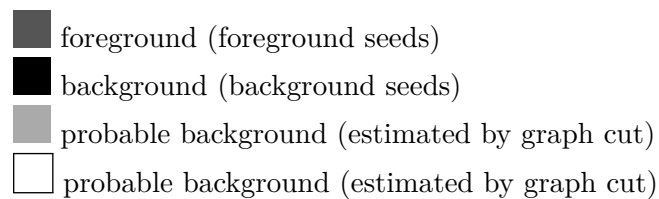


Figure 7. Labelling of the pixels in the mask image

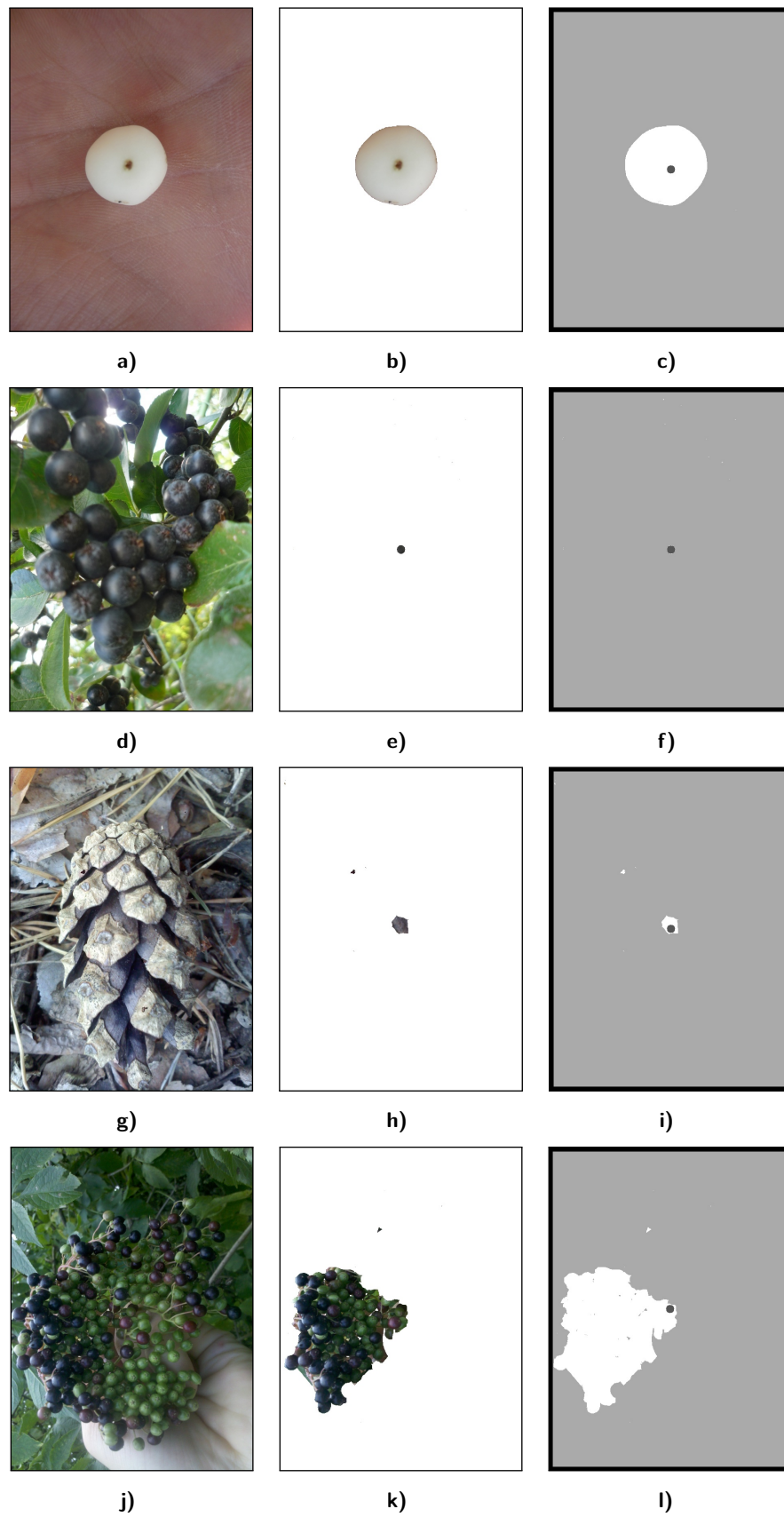


Figure 8. The original, foreground, and mask image of the best (a–c) and the worst (d–l) cases. 16^3 bins, $\gamma = 140$, t-links R' . For mask legend see Figure 7

2.3.2. L*a*b* histogram

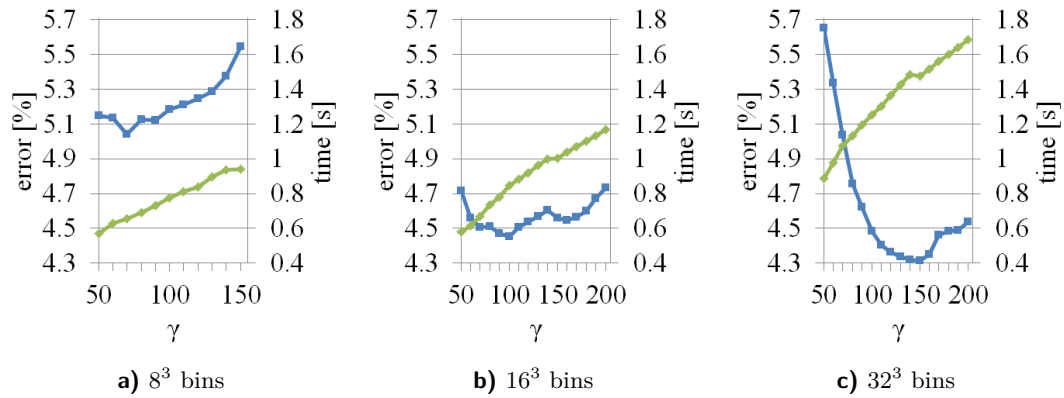


Figure 9. L*a*b* histogram results, $\mathbf{w} = (1, 1, 1)$, t-links R' . For legend, see Figure 5.

The Figure 9 shows that 16^3 bins are the best option, taking into account the time consumption. Although the histogram with 32^3 bins gives better results, the average elapsed time is too high for a near-real-time application. The Figure 10 specifies that the combination $\gamma = 140$, 16^3 bins and $\mathbf{w} = (1, 10, 10)$ would be the best setting. All that is with setting: t-links R' .

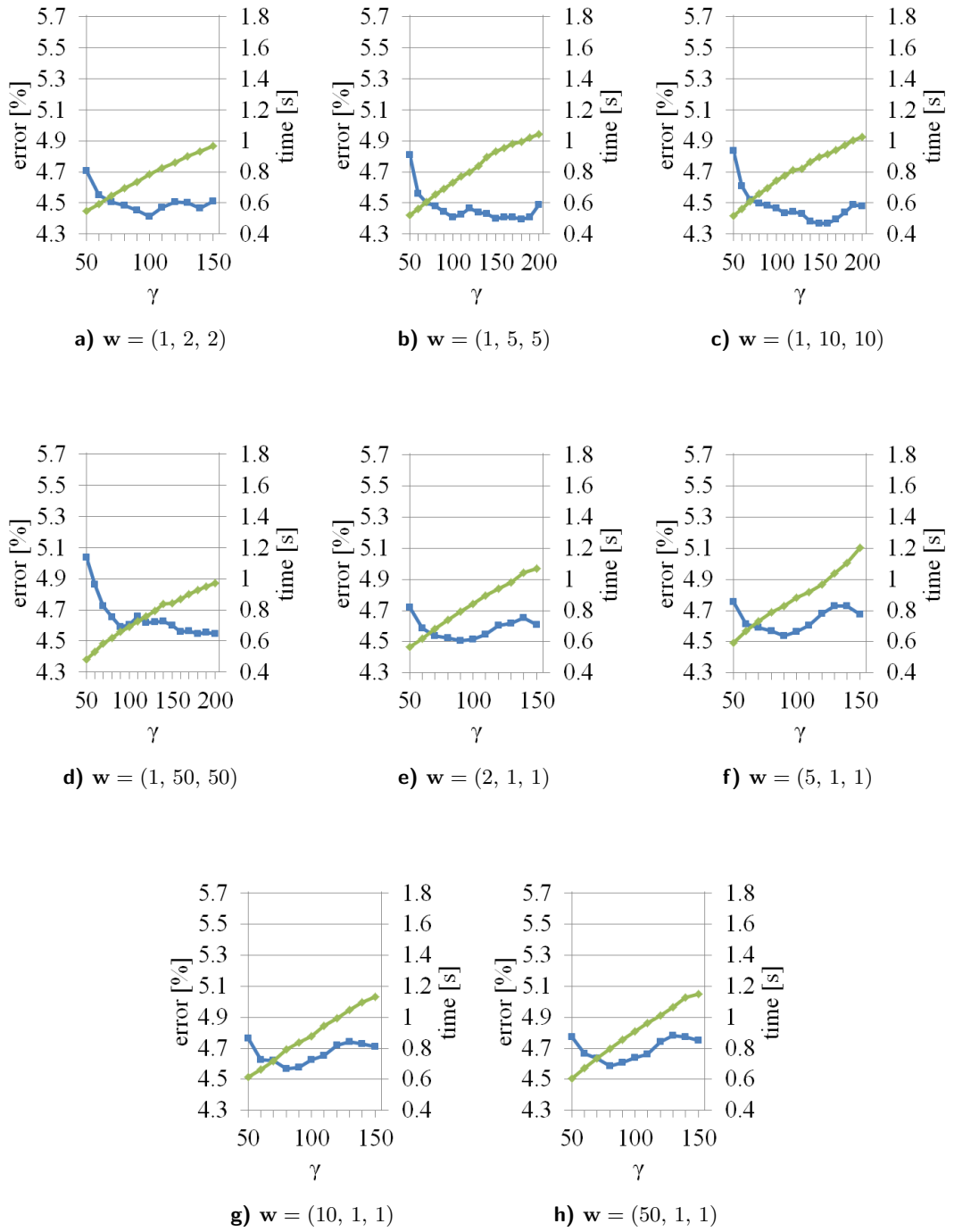


Figure 10. $L^*a^*b^*$ histogram results, 16^3 bins, t-links R' . For legend, see Figure 5.

2. Segmentation

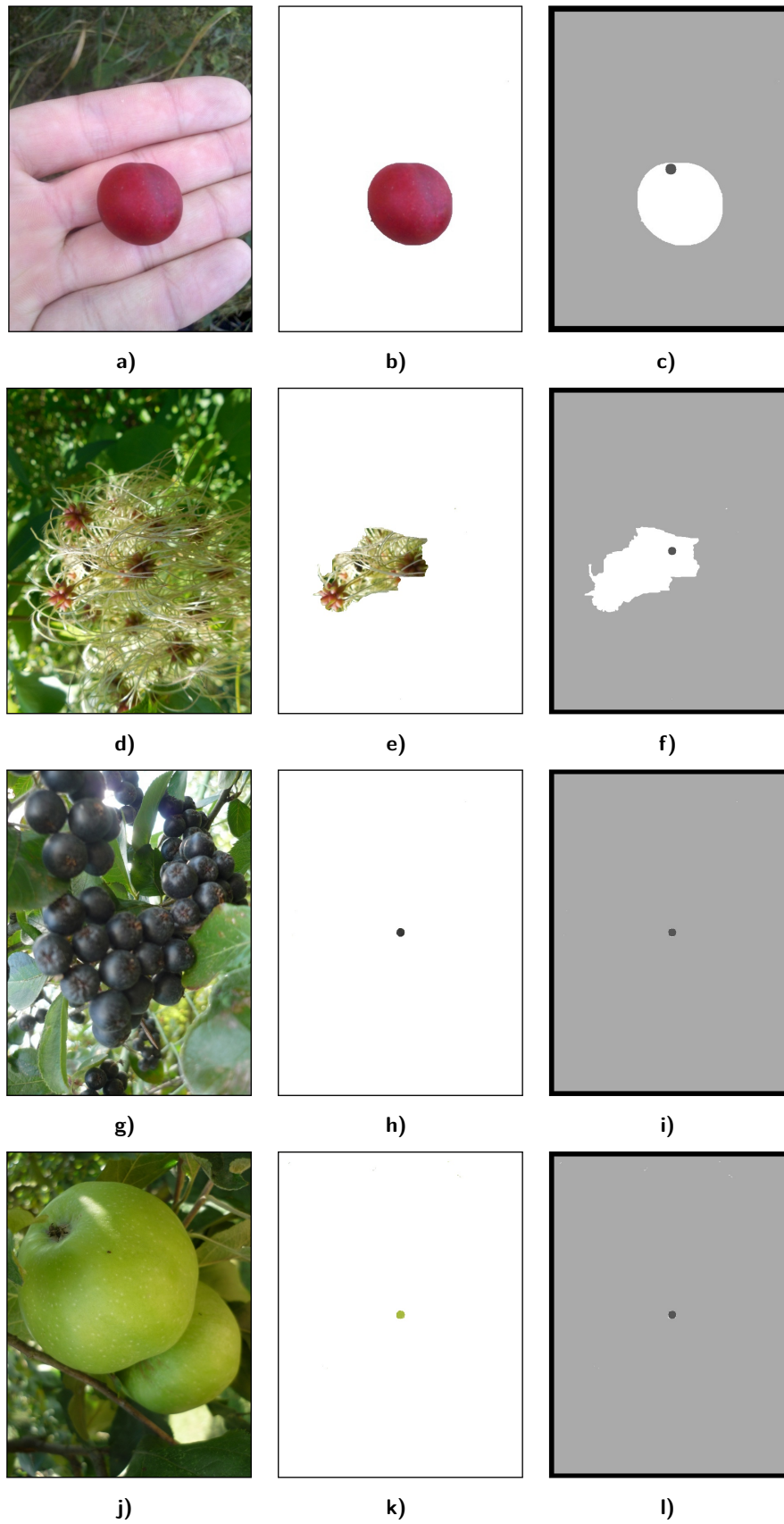


Figure 11. The original, foreground, and mask image of the best (a–c) and the worst (d–l) cases. 16^3 bins, $\gamma = 140$, $\mathbf{w} = (1, 10, 10)$, t-links R . For mask legend see Figure 7.

2.3.3. k-Means histogram

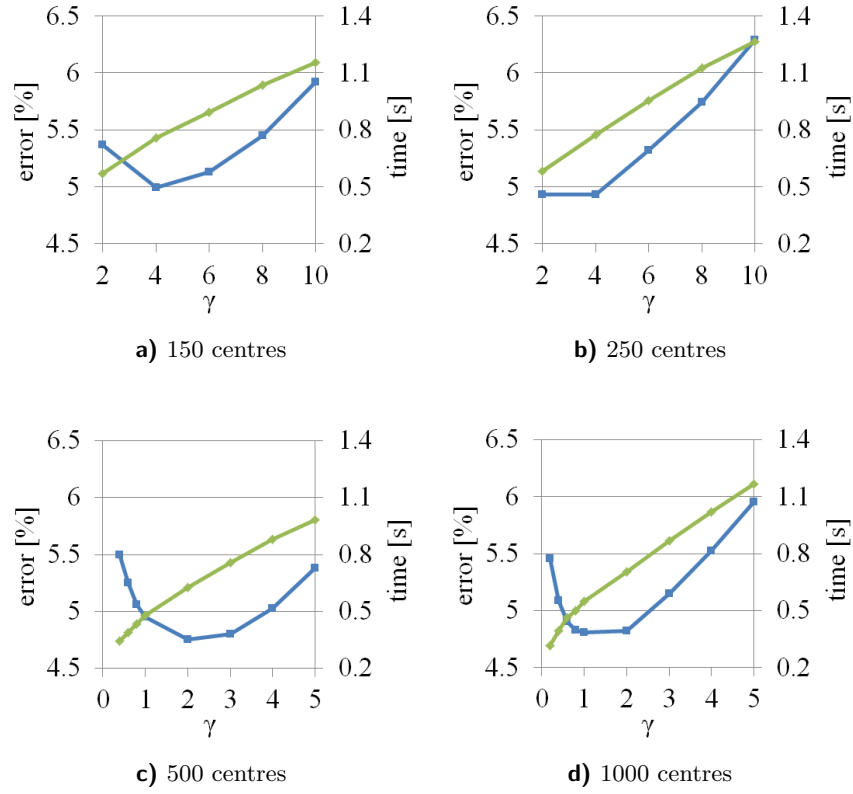


Figure 12. k-Means histogram results, $\mathbf{w} = (1, 1, 1)$, t-links R . For legend, see Figure 5.

The Figure 12 shows that $k = 500$, $\gamma = 2$ and t-links R is the best option with the $\mathbf{w} = (1, 1, 1)$. From the Figure 13, it is clear that changing the weights to $\mathbf{w} = (1, 10, 10)$ improves the segmentation.

2. Segmentation

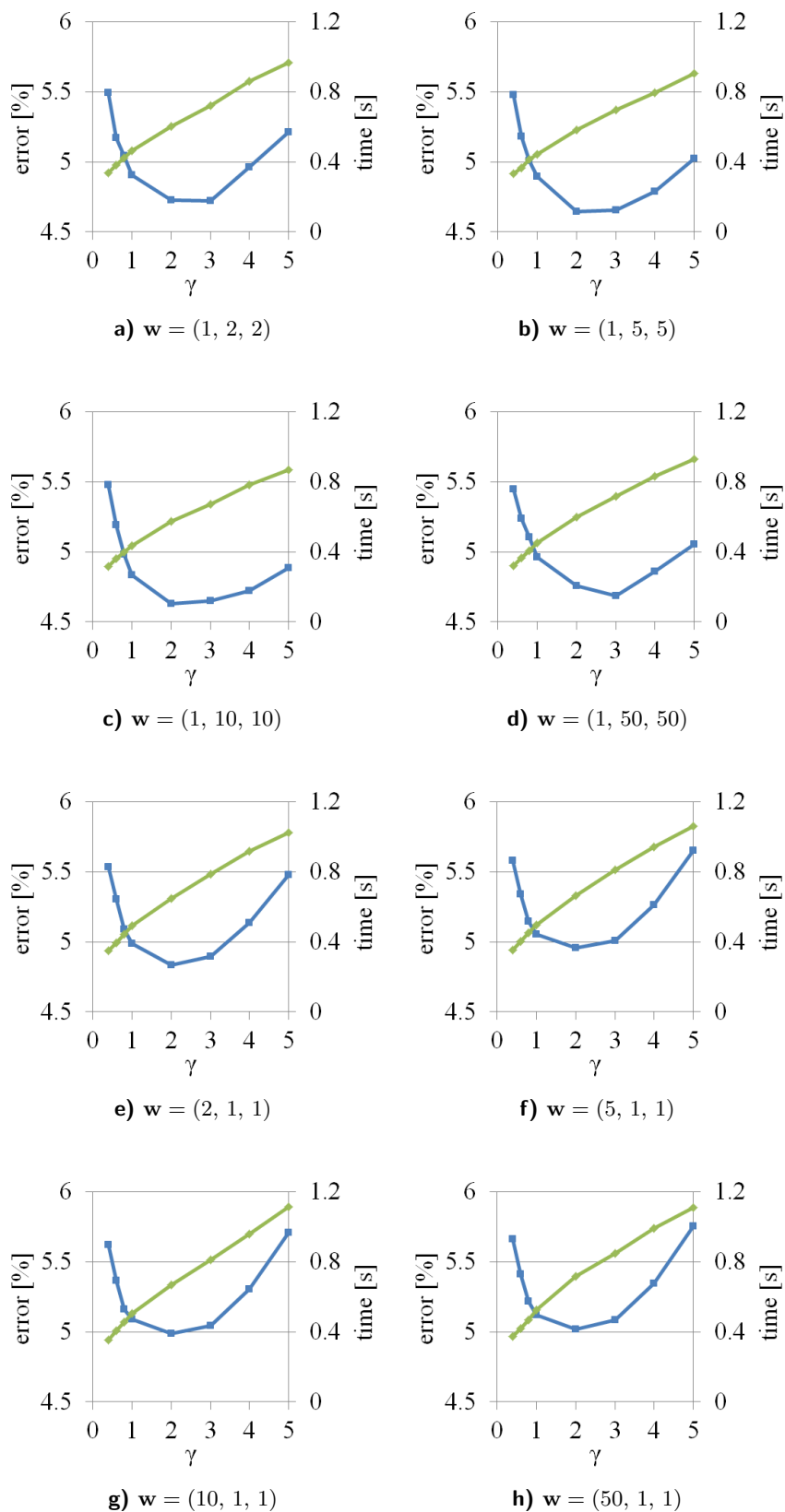


Figure 13. k-Means histogram results, $k = 500$, $t\text{-links } R$. For legend, see Figure 5.

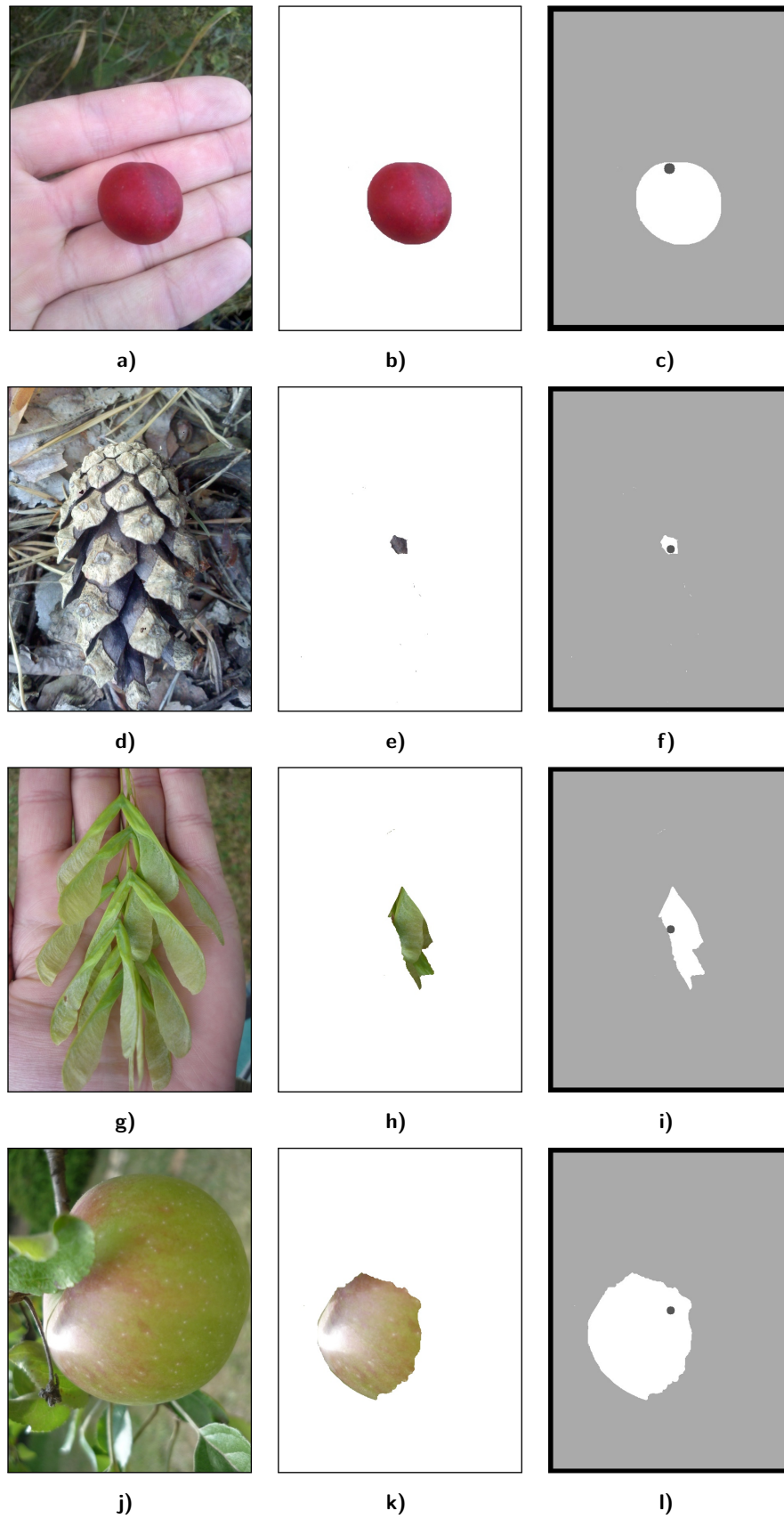


Figure 14. The original, foreground, and mask image of the best (a–c) and the worst (d–l) cases. $k = 500$, $\gamma = 2$, $\mathbf{w} = (1, 10, 10)$, t-links R . For mask legend see Figure 7

2.3.4. T-links

Two modifications of t-links were considered: R , R' (for the details, see section 2.2.1). Previous experiments were run with t-links R' for uniform histograms and R for k-means histograms. Figure 15 illustrates that other way around, the segmentation error is higher, for the k-means very significantly.

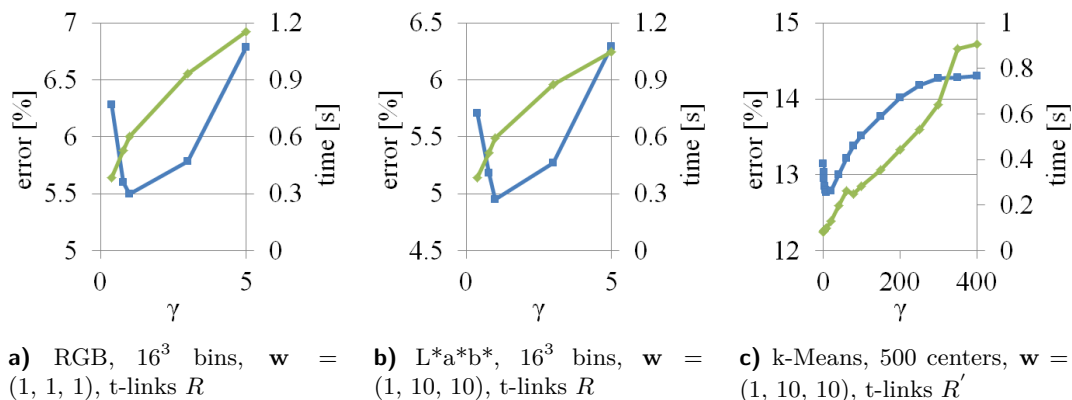


Figure 15. T-link modifications. For legend, see Figure 5

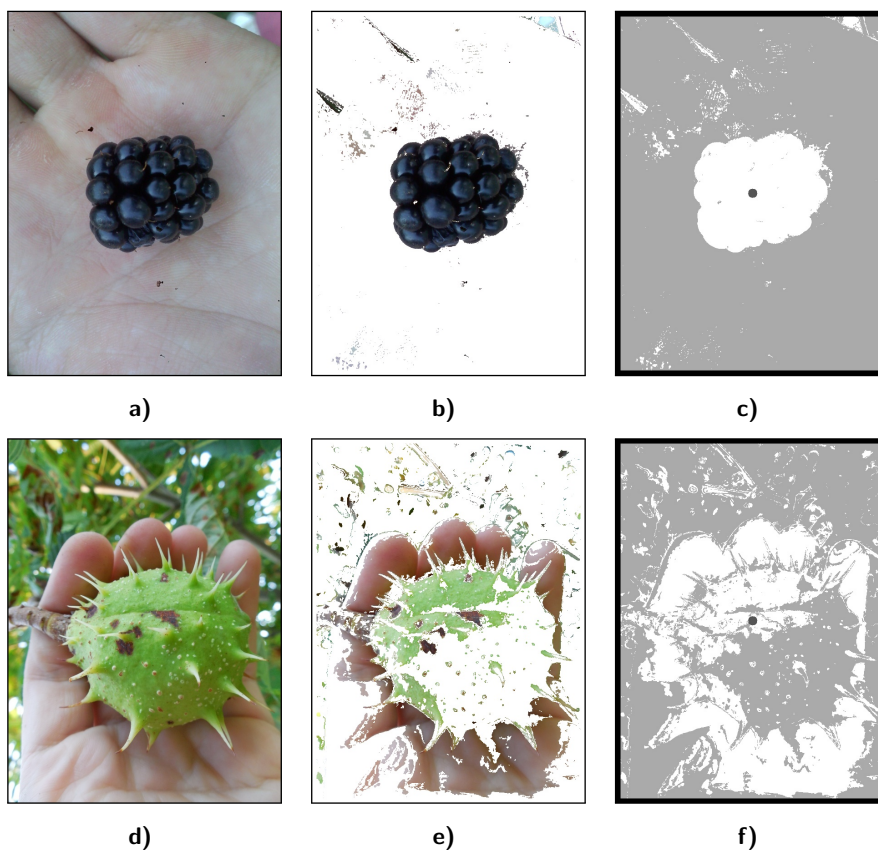


Figure 16. The original, foreground, and mask image of the best (a-c) and the worst (d-f) cases. k-Means $L^*a^*b^*$ histogram, t-links R' , $k = 500$, $\gamma = 2$, $\mathbf{w} = (1, 10, 10)$. For mask legend see Figure 7.

2.4. Results

The Table 2 presents the comparison of the best settings of histograms for the graph cut segmentation. The lowest error has the uniform $L^*a^*b^*$ histogram, however, this is redeemed by the biggest time-consumption. The RGB and k-means histogram error is on the same level, average total elapsed time is slightly better for the RGB setting. The main time difference, lies in the graph construction stage.

For my purposes, the optimal setting is the k-means histogram from the Table 2, because it is reasonably quick and the histogram computed in segmentation stage can be later used for recognition (see section 4.2). For a standalone near-real-time segmentation, the best setting is the RGB histogram from the Table 2, which is much faster than the best performing $L^*a^*b^*$.

| Histogram type | Average error | Average time | | | |
|---|---------------|-----------------------|--------------------|-------------------------|---------|
| | | Histogram calculation | Graph construction | Segmentation estimation | Total |
| RGB, 8^3 bins, $\gamma = 70$, t-links R' , $\mathbf{w} = (1, 1, 1)$ | 4.63% | 2 ms | 367 ms | 623 ms | 992 ms |
| $L^*a^*b^*$, 16^3 bins, $\gamma = 140$, t-links R' , $\mathbf{w} = (1, 10, 10)$ | 4.38% | 17 ms | 483 ms | 863 ms | 1363 ms |
| k-Means, $k = 500$, $\gamma = 2$, t-links R , $\mathbf{w} = (1, 10, 10)$ | 4.63% | 18 ms | 512 ms | 573 ms | 1103 ms |

Table 2. Comparison of the chosen histograms

3. Dataset

There is no publicly available dataset for fruit segmentation or recognition in natural environment. Therefore for development and testing purposes I photographed, gathered and labeled images of fruits. They were photographed by a compact camera or mobile phone. I took most of them, the rest was taken by Ing. Petra Stašáková and Václav Tyle. All images were taken in the Czech Republic and they capture the fruit that is common in this area.

Labeling the tree species is a complicated task, and since the author of this thesis is not a dendrologist, there is no guarantee that the labeling is entirely correct. However, it should be accurate enough for the purposes of this thesis.


The dataset contains 538 images of 54 species on natural background (mainly leaves or a hand). Most of the images are photographed in automatic mode, therefore the white-balance is also automatic. For the purposes of this thesis is an original resolution too high, therefore the images were resized to 800×600 px resolution.

This dataset is an attachment of this thesis, see Appendix A.

| | Name | Images | | Name | Images |
|---|--|--------|--|---|--------|
|  | Alnus (Alnus spp.) | 11 |  | Apple tree (Malus spp.) | 12 |
|  | Ash (Fraxinus excelsior) | 12 |  | Bean (Phaseolus spp.) | 8 |
|  | Birch (Betula spp.) | 15 |  | Black elder (Sambucus nigra) | 14 |
|  | Black locust (Robinia pseudoacacia) | 2 |  | Blackberry (Rubus spp.) | 24 |
|  | Blackthorn (Prunus spinosa) | 6 |  | Bladder campion (Silene vulgaris) | 6 |
|  | Common dogwood (Cornus sanguinea) | 6 |  | Common grape vine (Vitis vinifera) | 7 |
|  | Common plum (Prunus domestica spp.) | 2 |  | Common Whitebeam (Sorbus aria) | 5 |
|  | Cranberry (Vaccinium vitis-idaea) | 4 |  | Cucumber (Cucumis sativus) | 4 |
|  | Datura (Datura stramonium) | 4 |  | Dog rose (Rosa canina) | 18 |
|  | Dogwood (Cornus spp.) | 5 |  | English hawthorn (Crataegus laevigata) | 17 |

3. Dataset

| | Name | Images | | Name | Images |
|---|--|--------|---|---|--------|
|  | English walnut (<i>Juglans regia</i>) | 11 |  | European beech (<i>Fagus sylvatica</i>) | 3 |
|  | European blueberry (<i>Vaccinium myrtillus</i>) | 7 |  | European larch (<i>Larix decidua</i>) | 20 |
|  | European spindle (<i>Euonymus europaeus</i>) | 14 |  | European hornbeam (<i>Carpinus betulus</i>) | 13 |
|  | Fir (<i>Abies</i> spp.) | 10 |  | Firethorn (<i>Pyracantha</i> spp.) | 18 |
|  | Globe thistles (<i>Echinops</i> spp.) | 2 |  | Greater burdock (<i>Arctium lappa</i>) | 10 |
|  | Hazel (<i>Corylus</i> spp.) | 10 |  | Horse chestnut (<i>Aesculus hippocastanum</i>) | 7 |
|  | Cherry plum (<i>Prunus cerasifera</i>) | 7 |  | Chokeberries (<i>Aronia</i> spp.) | 6 |
|  | Lime tree (<i>Tilia</i> spp.) | 9 |  | Maize (<i>Zea mays</i> spp.) | 13 |
|  | Maple (<i>Acer</i> spp.) | 12 |  | Oak (<i>Quercus</i> spp.) | 18 |
|  | Old man's beard (<i>Clematis vitalba</i>) | 4 |  | Peach (<i>Prunus persica</i>) | 4 |

| | Name | Images | | Name | Images |
|---|--|--------|--|---|--------|
|  | Pine (<i>Pinus</i> spp.) | 26 |  | Plum (<i>Prunus domestica</i> spp.) | 17 |
|  | Raspberry (<i>Rubus idaeus</i>) | 14 |  | Snowberry (<i>Symphoricarpos</i> spp.) | 9 |
|  | Spruce (<i>Picea</i> spp.) | 19 |  | Sumac (<i>Rhus</i> spp.) | 5 |
|  | Thicket Creeper (<i>Parthenocissus vitacea</i>) | 2 |  | Wayfaring tree (<i>Viburnum lantana</i>) | 5 |
|  | White cedar (<i>Thuja occidentalis</i>) | 3 |  | White currant (<i>Ribes glandulosum</i>) | 3 |
|  | White pine (<i>Pinus strobus</i>) | 2 |  | Whitebeam (<i>Sorbus</i> spp.) | 14 |
|  | Wild privet (<i>Ligustrum vulgare</i>) | 7 |  | Zucchini (<i>Cucurbita pepo</i>) | 2 |

4. Recognition

4.1. State of the Art

Fruit recognition is mostly applied in automatic harvesting systems, usually limited to the one particular fruit species. That allows to determine the key features for identifying the fruit on a tree and distinguish ripe fruit from the unripe, if necessary. These systems (e.g. the one in Figure 17) are sometimes also equipped with sophisticated sensors e.g. stereo cameras or odor sensors.

The second commonly researched problem, quite similar to the one discussed in this thesis, is a fruit sorting and fruit identification in artificial conditions (e.g. automatic fruit identification in supermarkets).



Figure 17. Automatic fruit picking system proposed by Zhao et al. (Source [10])

Aibinu et al. [11] proposes a fruit identification algorithm for development of an automatic fruit identification and sorting system. Classification is based on the minimum Euclidean distance of the feature vectors. The features are derived using an artificial neural network, Fourier descriptors and spatial domain analysis. The mean values of RGB components in the combination with neural network are used to accurately detect the color of a fruit. The discrete Fourier transform is applied on a boundary signature, giving the Fourier descriptor. The spatial domain analysis of a shape contains:

- Compactness test $C = \frac{4\pi A}{P^2}$, where A is the area and P is the perimeter of the shape.
- Ratio test - the ratio of minor axis to major axis of the shape.
- Eccentricity of an ellipse.

Lei et al. [12] suggests Fuzzy recognition method based on matching degree of multi-characteristics. Five shape and color characteristics are considered:

- Size parameter – average length of sides of minimum bounding rectangle
- Shape parameter – aspect ratio of the same rectangle
- 3 color characteristics – red, green and blue intensity

Mustafa et al. [13] develops a fruit sorting system based on four-layer probabilistic neural network. Seventeen features are used:

- Morphological features
 - Area A
 - Major axis length
 - Minor axis length
 - $\frac{P^2}{A}$, where P is the perimeter
 - Major axis X and minor axis Y ratio $\frac{X}{Y}$
- Color features – mean and standard deviation of all RGB and HSI components

Rocha et al. [14] presents automatic multi-class produce classification system demonstrated on the images of fruit and vegetable, a system suitable for integration in the supermarket point-of-sale systems, similar to the outdated VeggieVision [15]. For the classification, it uses the Bagging Ensemble of Linear Discriminant Analysis (BLDA) with 17 iterations. Rocha et al. also considers using the Support Vector Machines (SVM) for classification, but comes to the conclusion that the SVM is more computationally demanding than LDA and in this specific case are the two approaches comparable in effectiveness, therefore the BLDA is the better option. The following image descriptors are used:

- Unser's descriptors [16] for the texture feature
- Color Coherence Vectors (CCVs) [17]
- Border/Interior pixel classification (BIC) [18]
- Appearance descriptors – vocabulary of parts related to [19] and [20]

Seng and Mirisae [21] proposes a fruit recognition based on k-nearest neighbours (k-nn) classification. The following features were used:

- Mean RGB color values of the fruit
- Perimeter P
- Area A
- Roundness $\frac{4\pi A}{P^2}$

Arivazhagan et al. [22] publishes a fruit recognition system built on a minimum distance classifier that relies on a combination of color and texture features extracted from HSV image. The statistical color features are derived from the both H and S chrominance channels:

- Mean
- Standard deviation
- Skewness
- Kurtosis

4. Recognition

Luminance V is decomposed using Discrete Wavelet Transform, after that the co-occurrence matrix is constructed and the following texture features are extracted from it:

- Contrast
- Energy
- Local homogeneity
- Cluster shade
- Cluster prominence

Wan-gan et al.[23] presents a method based on a Scale Invariant Feature Transform (SIFT) for feature extraction. It transforms an image to a set of feature vectors, each of which is invariant to scaling, translation and rotation and robust to noise and illumination changes. The Euclidean distance as a similarity metrics is implemented, allowing to set a threshold to detect the false matching points.

Kyaw et al. [24] suggests an automatic shape-based sorting of agricultural produce using SVM for the classification. The following shape features are extracted via edge detection method:

- Area A
- Major axis length X
- Minor axis length Y
- $\frac{P^2}{A}$, where P is the perimeter
- Major axis X and minor axis Y ratio $\frac{X}{Y}$

4.2. Recognition based on the color and shape features

Color is one of the most important characteristics of a fruit. I decided to represent the color with a histogram because it carries more information than the often-used mean and standard deviation of the particular color channels. I evaluated the same type of histograms as described in section 2.2.3: RGB/L*a*b* uniform histogram and L*a*b* k-means histogram.

Most of the fruit has an oval shape, therefore I represented shape with an ellipse. For ellipse fitting I used Direct Least Square Fitting [1]. Ellipse is a conic section which can be represented as $F(\mathbf{a}, \mathbf{x})$:

$$F(\mathbf{a}, \mathbf{x}) = \mathbf{a} \cdot \mathbf{x} = Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0, \quad (22)$$

where $\mathbf{a} = (A, B, C, D, E, F)^T$ and $\mathbf{x} = (x^2, xy, y^2, x, y, 1)^T$. The algebraic distance of a point (x_i, y_i) to the conic $F(\mathbf{a}, \mathbf{x}) = 0$ is $F(\mathbf{a}, \mathbf{x}_i)$. Minimizing the sum in (23) subject to the constrain (24) gets us the best-fit ellipse (in our terms).

$$\operatorname{argmin}_{\mathbf{a}} \left[\sum_{i=1}^N F(\mathbf{a}, \mathbf{x}_i)^2 \right] \quad (23)$$

$$4AC - B^2 = 1 \quad (24)$$

For classification, I used nearest neighbor classifier with the metric:

$$d_{ij} = d_h^2(H_i, H_j) + \zeta \left| \frac{b_i}{a_i} - \frac{b_j}{a_j} \right| \quad (25)$$

where $d_h(H_i, H_j)$ is one of the histogram metrics described in the section 4.2.1, ζ is a weight of shape term, a is a major axis and b is a minor axis of an ellipse. Using the axes ratio ensures the invariance to rotation and scaling.

4.2.1. Metrics for histogram comparison

Let H_i, H_j denote two histograms with the same number of bins M , and $d_h(H_i, H_j)$ denote a distance between two histograms. The following metrics could be used for histogram comparison [25]

- Bhattacharyya distance

$$d_h(H_i, H_j) = \sqrt{1 - \frac{1}{\sqrt{\bar{H}_i \bar{H}_j M^2} \sum_{n=1}^M \sqrt{H_i(n) \cdot H_j(n)}}, \quad \bar{H} = \frac{1}{M} \sum_{n=1}^M H(n) \quad (26)$$

- Correlation

$$d_h(H_i, H_j) = \frac{\sum_{n=1}^M [(H_i(n) - \bar{H}_i) (H_j(n) - \bar{H}_j)]}{\sqrt{\sum_{n=1}^M [(H_i(n) - \bar{H}_i)^2 (H_j(n) - \bar{H}_j)^2]}} \quad (27)$$

- Chi-Square

$$d_h(H_i, H_j) = \sum_{n=1}^M \frac{(H_i(n) - H_j(n))^2}{H_i(n)} \quad (28)$$

- Histogram Intersection

$$d_h(H_i, H_j) = \sum_{n=1}^M \min(H_i(n), H_j(n)) \quad (29)$$

4.3. Results

The output of the segmentation stage is an image divided into the two segments: a foreground and a background. In the recognition stage, we are only interested in the foreground. Since the segmentation method is semi-automatic, the user can fix the segmentation output in cases when the result of automatic initialization is incorrect. We can therefore assume that the user-approved segmentation is close to the ground truth. For that reason, I use the ground truth belonging to the dataset presented in section 3 for segmentation evaluation.

4.3.1. Recognition rates for color feature

The Figures 19–21 show that Bhattacharyya distance is in this case the best metric for histogram comparison. The increase of the histogram centers k has only minor influence.

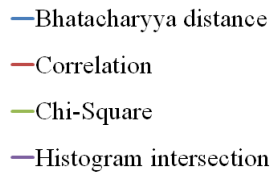


Figure 18. Legend for the Figures 19–21

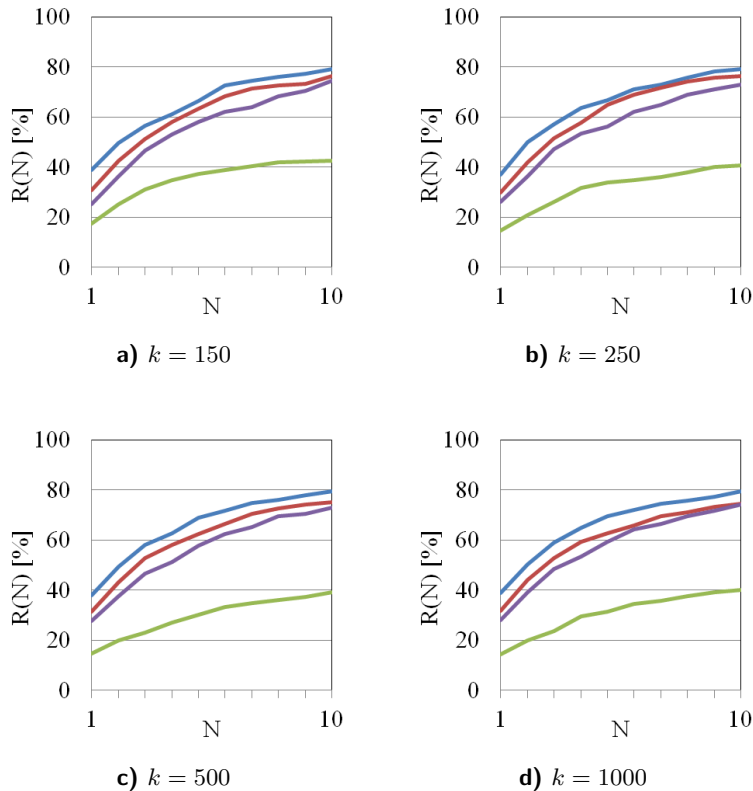


Figure 19. Recognition rates $R(N)$ of the top N results for k-means L*a*b* histogram

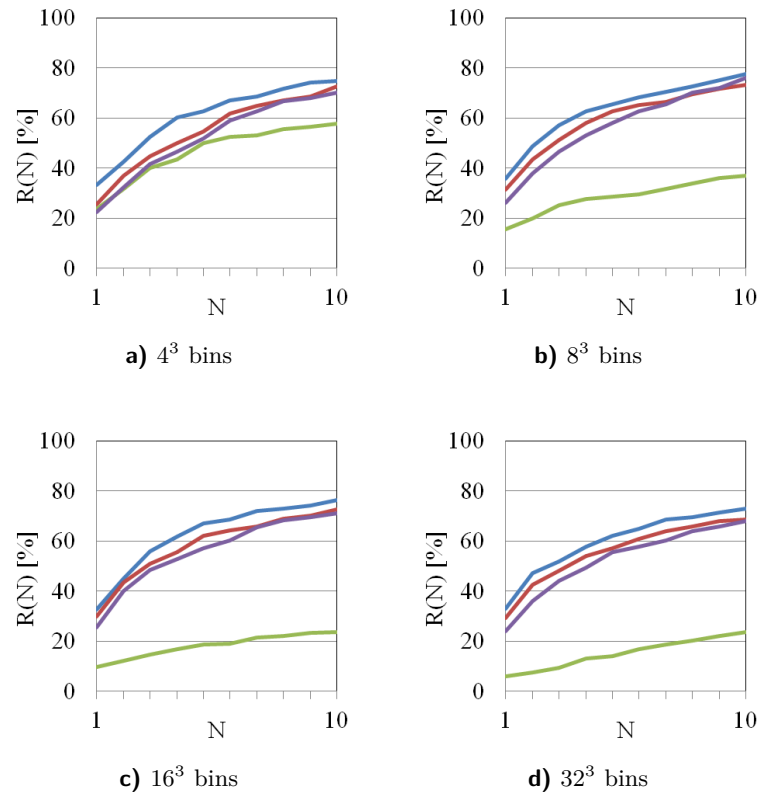


Figure 20. Recognition rates $R(N)$ of the top N results for RGB histogram

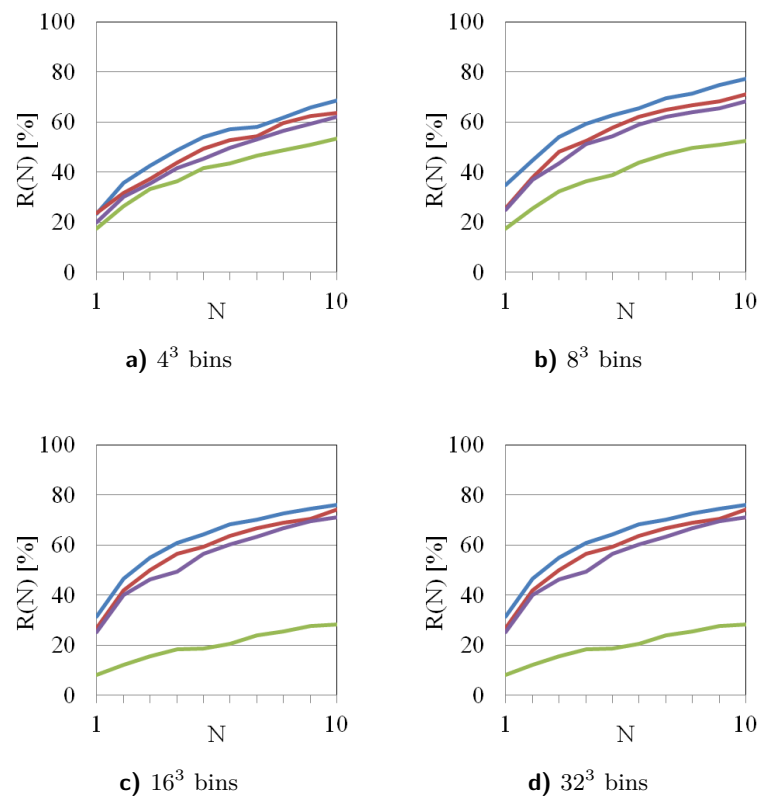


Figure 21. Recognition rates $R(N)$ of the top N results for $L^*a^*b^*$ histogram

4.3.2. Recognition rates for color and shape features

Adding a shape feature lead to about 5 percentage point increase of a recognition rate. The comparison is in the Figure 23.

The best setting of a weight is, according to Figure 22, $\zeta = 4.8$.

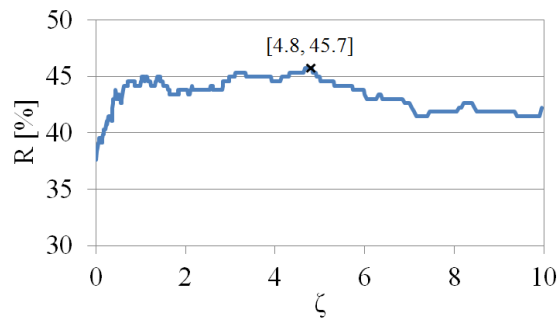


Figure 22. Recognition rates R with different ζ setting

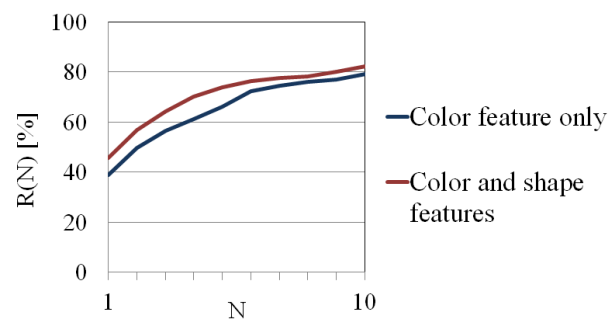


Figure 23. Recognition rates $R(N)$ of the top N results for a k-Means L^*a^*b histogram ($k = 250$) without and with shape feature ($\zeta = 4.8$)

5. Conclusion

In this thesis, I dealt with semi-automatic plant identification based on images of fruit in the natural environment. This task has two stages: segmentation and recognition.

Graph cut algorithm in a combination with color histogram is proposed as a segmentation method. The experiments show that the choice of constants and the histogram type has major influence on the segmentation. Nevertheless, with the optimal setting, all general histogram types (RGB uniform, L*a*b* uniform and k-means L*a*b* histogram) have comparable results. The slowest, but with the lowest average error 4.38%, is L*a*b* histogram. The quickest RGB and medium-quick k-means L*a*b* have the same average error of 4.63%. In this case, the best choice is a k-means L*a*b* histogram, because it can be used in recognition stage and therefore save some time.

Recognition is based on a nearest neighbor classifier using color and shape features. The color of the fruit is represented by color histogram, from which the best results is achieved by k-means L*a*b* histogram, giving the recognition rate of 39% for the top result and 69% for the top five results. The shape feature is represented by ratio of minor and major axis of an ellipse fitted by Direct Least Square Fitting method [1]. Color and shape feature combined give the recognition rate of 46% for the top result and 74% for the top five results.

For the development and testing purposes, I photographed, gathered and labeled dataset containing 538 images of 54 fruit species, which is, together with a demo application and ground truth, an attachment of this thesis.

5.1. Future work

Segmentation time-consumption could be improved by implementing a more efficient algorithm than OpenCV GrabCut, the promising candidate is the GridCut [26]. Modifications of t-links evaluated in this thesis shows, that using the a priori probability of the foreground and background could have a positive effect on the segmentation, that deserves a deeper study.

The basic recognition proposed in this thesis could be improved by using more features, especially shape and texture ones. The testing of more sophisticated classifiers also offers a wide range of improvements to this problem.

Appendix A.

Contents of the attached DVD

| Directory | Content description |
|------------------------|---|
| Fruit Dataset original | Images of fruit (original size) |
| Fruit Dataset 800x600 | Images of fruit (size 800 × 600 px) |
| gcFruit | Demo application for segmentation and recognition of fruit (NetBeans project) |
| Ground Truth | Segmented images with additional files (mask, original image, contour image) |
| Thesis | This thesis in PDF format |

Bibliography

- [1] Andrew Fitzgibbon, Maurizio Pilu, and Robert B Fisher. “Direct least square fitting of ellipses”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 21.5 (1999), pp. 476–480.
- [2] Tomáš Sixta. “Image and Video-based Recognition of Natural Objects”. MA thesis. Czech Technical University in Prague, 2011.
- [3] Milan Šulc. *Image-based Recognition of Plants*. 2012.
- [4] Milan Šonka, Václav Hlaváč, and Roger Boyle. *Image processing, analysis, and machine vision*. 3rd ed. Thomson, 2008.
- [5] Yuri Y Boykov and M-P Jolly. “Interactive graph cuts for optimal boundary & region segmentation of objects in ND images”. In: *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*. Vol. 1. IEEE. 2001, pp. 105–112.
- [6] B. V. Cherkassky and A. V. Goldberg. “On Implementing the Push—Relabel Method for the Maximum Flow Problem”. English. In: *Algorithmica* 19.4 (1997), pp. 390–410. ISSN: 0178-4617. DOI: 10.1007/PL00009180. URL: <http://dx.doi.org/10.1007/PL00009180>.
- [7] D. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton, NJ, USA: Princeton University Press, 2010. ISBN: 0691146675, 9780691146676.
- [8] Yuri Boykov and Vladimir Kolmogorov. “An Experimental Comparison of Min-cut/Max-flow Algorithms for Energy Minimization in Vision”. English. In: *Energy Minimization Methods in Computer Vision and Pattern Recognition*. Ed. by Mário Figueiredo, Josiane Zerubia, and AnilK. Jain. Vol. 2134. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2001, pp. 359–374. ISBN: 978-3-540-42523-6. DOI: 10.1007/3-540-44745-8_24. URL: http://dx.doi.org/10.1007/3-540-44745-8_24.
- [9] David Forsyth and Jean Ponce. *Computer vision*. Prentice Hall, c2003.
- [10] Jun Zhao, J. Tow, and J. Katupitiya. “On-tree fruit recognition using texture properties and color data”. In: *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*. Aug. Pp. 263–268. DOI: 10.1109/IROS.2005.1545592.
- [11] A.M. Aibinu et al. “Automatic Fruits Identification System Using Hybrid Technique”. In: *Electronic Design, Test and Application (DELTA), 2011 Sixth IEEE International Symposium on*. 2011, pp. 217–221. DOI: 10.1109/DELTA.2011.47.
- [12] Jingtao Lei, Tianmiao Wang, and Zhenbang Gong. “Study on machine vision fuzzy recognition based on matching degree of multi-characteristics”. In: *Life System Modeling and Intelligent Computing* (2010), pp. 459–468.

- [13] N.B.A. Mustafa et al. “Classification of fruits using Probabilistic Neural Networks - Improvement using color features”. In: *TENCON 2011 - 2011 IEEE Region 10 Conference*. Nov. Pp. 264–269. DOI: 10.1109/TENCON.2011.6129105.
- [14] Anderson Rocha et al. “Automatic produce classification from images using color, texture and appearance cues”. In: *Computer Graphics and Image Processing, 2008. SIBGRAPI'08. XXI Brazilian Symposium on*. IEEE. 2008, pp. 3–10.
- [15] Ruud M Bolle et al. “Veggievision: A produce recognition system”. In: *Applications of Computer Vision, 1996. WACV'96., Proceedings 3rd IEEE Workshop on*. IEEE. 1996, pp. 244–251.
- [16] Michael Unser. “Sum and difference histograms for texture classification”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 1 (1986), pp. 118–125.
- [17] Greg Pass, Ramin Zabih, and Justin Miller. “Comparing images using color coherence vectors”. In: *Proceedings of the fourth ACM international conference on Multimedia*. ACM. 1997, pp. 65–73.
- [18] Renato O Stehling, Mario A Nascimento, and Alexandre X Falcão. “A compact and efficient image retrieval approach based on border/interior pixel classification”. In: *Proceedings of the eleventh international conference on Information and knowledge management*. ACM. 2002, pp. 102–109.
- [19] Shivani Agarwal, Aatif Awan, and Dan Roth. “Learning to detect objects in images via a sparse, part-based representation”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 26.11 (2004), pp. 1475–1490.
- [20] Frederic Jurie and Bill Triggs. “Creating efficient codebooks for visual recognition”. In: *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*. Vol. 1. IEEE. 2005, pp. 604–610.
- [21] Woo Chaw Seng and S.H. Mirisae. “A new method for fruits recognition system”. In: *Electrical Engineering and Informatics, 2009. ICEEI '09. International Conference on*. Vol. 01. Aug. Pp. 130–134. DOI: 10.1109/ICEEI.2009.5254804.
- [22] S Arivazhagan et al. “Fruit recognition using color and texture features”. In: *Journal of Emerging Trends in Computing and Information Sciences* 1.2 (2010), pp. 90–94.
- [23] Song Wan-gan, Guo Hong-xia, and Wang Yan. “A Method of Fruits Recognition Based on SIFT Characteristics Matching”. In: *Artificial Intelligence and Computational Intelligence, 2009. AICI'09. International Conference on*. Vol. 3. IEEE. 2009, pp. 119–122.
- [24] M.M. Kyaw, S.K. Ahmed, and Z.A.M. Sharrif. “Shape-based sorting of agricultural produce using support vector machines in a MATLAB / SIMULINK environment”. In: *Signal Processing Its Applications, 2009. CSPA 2009. 5th International Colloquium on*. March, pp. 135–139. DOI: 10.1109/CSPA.2009.5069203.
- [25] *OpenCV documentation*. [Online] [Accessed December 27, 2013]. URL: <http://docs.opencv.org/modules/imgproc/doc/histograms.html?highlight=comparehist#cv2.compareHist>.
- [26] Ondrej Jamriska, Daniel Sykora, and Alexander Hornung. “Cache-efficient graph cuts on structured grids”. In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE. 2012, pp. 3673–3680.