A combined structural/statistical texture analysis of monolayer ovarian cancer cell nuclei

by

Pål Nordby

THESIS

for the degree of

MASTER OF SCIENCE

Modelling and Data Analysis

Faculty of Mathematics and Natural Sciences
University of Oslo

December 2010

Acknowledgements

This thesis has been submitted to the Faculty of Mathematics and Natural Sciences at the University of Oslo in partial fulfillment of the requirements for the degree Master of Science in Modeling and Data Analysis. The work started in September 2009 and was finished in December 2010.

My supervisors through this thesis work has been Professor Fritz Albregtsen at the Digital Signal Processing and Image Analysis Group at the Department of Informatics and Professor Dr. Håvard E. Danielsen at the Department of Medical Informatics at The Norwegian Radium Hospital.

I would like to thank my supervisors, especially Fritz who introduced me to this field, and for the patience and sincere interest he has shown in this project. I also would like to thank my family, where my three nephews, Marcus, Nicolai and Oliver, have been a huge inspiration.

Oslo, December 2010
Pål Nordby

Contents

Abstract

Determining the prognosis in an early stage of human cancer can be essential for the choice of optimal therapy. Digital image analysis of cell nuclei is a very useful tool to obtain quantitative information for robust and reliable prognosis. A substantial number of papers have been published on the use of various texture analysis methods for diagnostic and prognostic work on human cancer, and most of the studies are based on texture analysis of the gray levels in the images.

We will take another approach, and use a refined adaptive segmentation method developed in this thesis to describe the structures inside the cell nuclei images. The refined thresholding method is spatially adaptive within each image, while its parameters are adapted to the histogram of each image. In a novel approach, we evaluate the characteristics of the segmented structures statistically to decide the prognosis per image, and finally a rule is formed to classify each patient.

The data set analyzed consists of 134 patients with early ovarian cancer. The problems with such small data sets is addressed, and a solution based on statistical bootstrapping is proposed. This gives a more robust estimate of the correct classification rate (CCR) than the traditional single CCR estimate would, and in addition gives a CCR uncertainty estimate.

Dividing the data set into two groups based on DNA-ploidy, effectively introducing a simple two-step classification scheme, substantially improved the performance of the classification. Combining the structural features extracted from the objects inside each cell nucleus with the best statistical gray level feature - an adaptive entropy matrix feature from a previous study on the same material - further improved the correct classification rate, leading to a CCR close to 90%.

In conclusion, the significant improvement in correct classification rate obtained by combining the best statistical and structural texture features seems to hold a promise of very high CCRs, which would be immensely valuable in prognostic work on human cancers. This may be true beyond the present data set, and possibly quite generally. But obviously some caution is called for, and more tests on different and larger data sets should be performed.

Chapter 1

Introduction

The main aim of this thesis has been to segment and separate structures inside cell nuclei images and develop and evaluate structural features with a potential prognostic value for early ovarian cancer. Determining the prognosis in an early stage of cancer, can be essential for selection of therapy.

In this project we will focus on images from a study on ovarian cancer, but studies in image texture-based methods have been used in a wide range of human cancers [14].

While most of the published works in this field are based on statistical analysis of the gray level of the images, and the use of e.g. gray level co-occurrence matrix [7] (GLCM) to obtain classification features, we will take another approach and use segmentation to find bright and dark objects within the cell nuclei. Then extract features from these object and try to correctly classify each of the images, and in the end correctly classify the patients

The final result is dependent on every step of the automated algorithm that is developed, but the crucial step is the segmentation. A modification of Niblack's adaptive thresholding method is proposed, introducing two thresholds and estimating the parameters involved from analytical expressions based on the histogram of each image, instead of using pure guesswork or finding a set of parameters suitable for most of the images - as is the usual practice.

Given only a small data set, 134 patients, we must be careful with the experiment design and choose few, but informative features. There are also other issues with small data sets that will be discussed, when it comes to training and evaluating the features.

As a result of this thesis work, it seems possible to combine the best of the gray level texture features with a few structural features obtained by our improved adaptive thresholding within the cell nuclei, to obtain a significant improvement in the correct prognostic classification rate for patients with early ovarian cancer.

Organization of the thesis

The thesis is organized in the same way as the automated algorithm is developed. All chapters start with some of basic theory needed, and then a discussion of the specific choices made for our needs.

The next chapter gives a description of the material used in this thesis, i.e., the cell nuclei images and how they are processed. Chapter 3 presents a detailed introduction to segmentation and a closer look at the most popular algorithms for both global and adaptive segmentation. Here, the detail of our new variation of Niblack's adaptive thresholding method are given. In chapter 4 morphological theory and algorithms is briefly described and an algorithm for separating the segmented objects is described. Chapter 5 discusses the extraction of features and the complexity of feature selection. Another subject which is presented in this chapter is classification, and different classification methods are described and discussed. In chapter 6 the results and discussion of different experiments are shown. Even though we only have one data set we will use some variations of this set, implying that we will use different approaches for dividing the data set, and for each partitioning there will be several analyses. The last chapter sums up our findings and proposes some improvements and suggestions to further research.

Chapter 2

The Images

The images used in this project are from a study (done in 1982-1989) on ovarian cancer. The patients were treated at The Norwegian Radium Hospital and then followed up for at least 10 years. The patient who survived with no relapse were categorized as good prognosis and patients that died in the follow up or relapsed as bad prognosis. The data set consists of 40 patients with bad prognosis and 94 patients with good prognosis.

2.1 Monolayers

This section is a description of how the cell nuclei are imaged, and because of detailed and complicated terms, the rest of the section is a citation from *B.Nielsen et al.'s* [12] article.

"Parafin-embedded tissue samples fixed in 4% buffered formalin were sectioned ($2x50\mu m$) and enzymatically digested (Sigma protease, type XXIV, Sigma Chemical C., St. Louise, Missouri, USA) for the preparation of isolated nuclei (monolayers). The nuclei were Feulgen-Schiff stained according to an established protocol. Blocks were selected by the pathologist, who selected the tumour tissue to be prepared.

The Fairfield DNA Ploidy System (Farfield Imaging LTD, Kent, England), which consisted of a Zeiss Axionplan microscope equipped with a 40/0.75 objective lens (Zeiss), a 546 nm green filter and a black and white high resolution digital camera (C4742-95, Hamamatsu Photonics K.K., Hamamatsu, Japan) was used. A shade correction was performed for each image field and the image was stored with 10 bits per pixel. The pixel resolution was 166 nm per pixel on the cell specimen.

Trained personnel performed a screening of the cells in the microscope and selected tumour cells for the analysis. Stromal cells, necrotic cells, doublest or cut cells were disregarded. The nuclei were segmented from the background by using a global threshold. The histograms of all nuclei images were normalized to the same mean values (650.0) and standard deviation(120.0). This was done to normalize the first order statistics while utilizing the whole range of gray levels, and avoid clipping of the histograms."

Figure 2.1 shows some examples of the monolayer cell nuclei images. The four images in the upper row are from a patient with good prognosis, while the four cell nuclei images in the bottom row are from a patient with bad prognosis.

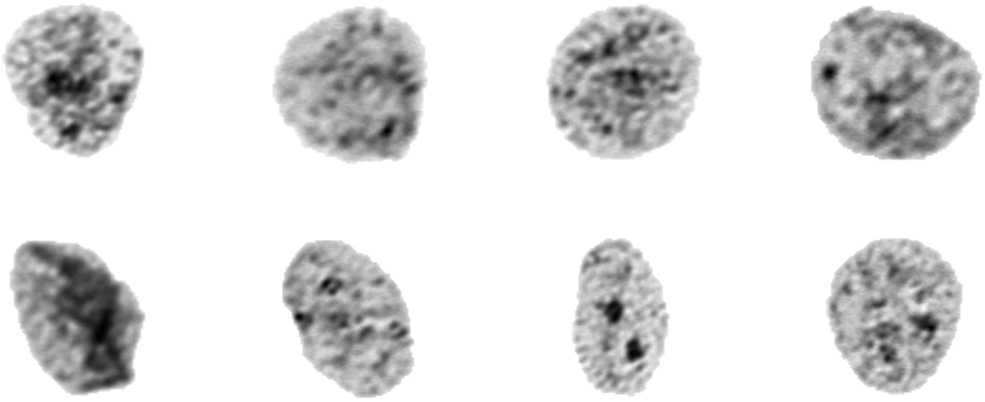


Figure 2.1: *Four cell nuclei from a patient with good prognosis (upper) and four cell nuclei from a patient with bad prognosis (lower).*

2.1.1 Earlier research

As stated in the introduction, a lot of has been done using digital image analysis studying human cancer. However, most of the published work where image texture-based methods have been used for digital microscopy screening, diagnosis, and prognosis of human cancer [14] are based on feature extraction from the gray level run length matrix (GLRLM) [4] and GLCM [7], as well as fractal texture methods [13]. The material of early ovarian cancer, used in this thesis, has also been used in several other studies [12], [11], [15], [13].

2.1.2 Our aim/goal

Our goal with this thesis, is certainly to classify all of the patients into the correct prognostic class, but also to take another approach on the analysis the data. Our contributions will hopefully give a better CCR than in previous studies, and when the different approaches are combined it will result in a robust classification scheme. Which in the end will help patients getting the best possible therapy in their struggle against ovarian cancer.

Chapter 3

Segmentation

Segmentation is one of the most important steps in an image analysis task. The purpose of segmentation is to divide the image into meaningful regions or objects. How much detail that is needed depends on the problem that is being solved.

Segmentation algorithms are based on two basic categories dealing with properties of intensity values; discontinuity and similarity [5]. Edge-based segmentation is an example of discontinuity, where we use the fact that the discontinuity boundaries of regions are separating them from each other. Region-based segmentation, where the image is divided into regions that are similar according to a set of predefined criteria, is an example of the other.

3.1 Edge detection

3.1.1 Derivatives of digital functions

To get the edge information in an image we will need to find the derivative for the pixels in it. Derivatives of digital functions are defined in terms of differences, and there are several ways to approximate these differences, as long as one follows some restrictions [5].
For the first order derivative we will require that

1. the derivatives must be zero in areas of constant intensity

2. the derivatives must be nonzero at the onset of an intensity step or ramp

3. the derivatives must be nonzero at points along an intensity ramp.

We have almost the same rules for the second derivative:

1. the derivatives must be zero in areas of constant intensity

2. the derivatives must be nonzero at the onset and end of an intensity step or ramp

3. the derivatives must be zero along intensity ramps.

The first order derivative of a function $f(x)$ at a point x is approximated by expanding the function $f(x+\Delta x)$ into Taylor series about x, letting $\Delta x = 1$, and only care about the linear terms we get

$$\frac{df}{dx} = f'(x) = f(x+1) - f(x) \tag{3.1}$$

For the second order derivative we have

$$\frac{d^2f}{dx^2} = f''(x) = f'(x+1) - f'(x) = f(x+2) - 2f(x+1) + f(x) \tag{3.2}$$

Since we are interested in the second derivative about a point x, we shift by one in the formula above and get

$$\frac{d^2f}{dx^2} = f(x+1) - 2f(x) + f(x-1) \tag{3.3}$$

As we shall see later, by the use of spatial filters we can compute derivatives at every pixel location in an image.

3.1.2 The image gradient

Image gradient is used to find the edge strength and direction at pixel (x,y) in an image f. The gradient of an image is denoted by ∇f, and is defined as

$$\nabla f = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \tag{3.4}$$

The magnitude of ∇f, is given by

$$M(x,y) = \sqrt{g_x^2 + g_y^2}. \tag{3.5}$$

The direction of the gradient vector is given by an angle

$$\alpha(x,y) = \tan^{-1}\left(\frac{g_y}{g_x}\right). \tag{3.6}$$

3.1.3 Gradient operators

To find the gradient values one must compute the partial derivatives of the image, and since we are dealing with digital quantities, we must approximate these and we get

$$g_x = \frac{\partial f(x,y)}{\partial x} = f(x+1,y) - f(x,y) \tag{3.7}$$

and

$$g_y = \frac{\partial f(x,y)}{\partial y} = f(x,y+1) - f(x,y) \tag{3.8}$$

We could approximate these partial derivatives by convolving the image with the asymmetric 1-D filters, for the x and y direction, given by

$$H_x = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & -1 \\ \hline 0 & 0 & 0 \\ \hline \end{array}\ ,\ H_y = \begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \tag{3.9}$$

or the symmetric 1-D filters

$$H_x = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 1 & 0 & -1 \\ \hline 0 & 0 & 0 \\ \hline \end{array}\ ,\ H_y = \begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline \end{array} \tag{3.10}$$

But the problem with these approximations is, for the asymmetric case, that the gradient estimate refer to a point between two pixels, and the x- and y-estimate does not refer to the same place in the image and the symmetric operator is also very sensitive for noise. The solution to these problems is to calculate the partial derivatives for three symmetric pairs. Thus we get an operator that is more robust to noise.

There are many types of such symmetric filters, and the simplest one is the Prewitt-operator which is given by

$$H_x(i,j) = \begin{array}{|c|c|c|}\hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline \end{array}\ ,\ H_y(i,j) = \begin{array}{|c|c|c|}\hline -1 & -1 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \tag{3.11}$$

And we see that the partial derivatives with this operator is then

$$g_x = \frac{\partial f}{\partial x} = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3) \tag{3.12}$$

$$g_y = \frac{\partial f}{\partial y} = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7) \tag{3.13}$$

where z_i is the intensity value in the image that the filter mask is covering.

A slight variation of the Prewitt operator is the Sobel operator which is given by

$$H_x(i,j) = \begin{array}{|c|c|c|}\hline 1 & 0 & -1 \\ \hline 2 & 0 & -2 \\ \hline 1 & 0 & -1 \\ \hline \end{array}, \; H_y(i,j) = \begin{array}{|c|c|c|}\hline -1 & -2 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 2 & 1 \\ \hline \end{array} \tag{3.14}$$

The Sobel operator emphasize more on the closest neighbors of the center pixel, i.e., the 4-connected neighbors, and give these positions a weight of 2, while the 8-connected neighbors have weight 1. And we get the following partial derivatives

$$g_x = \frac{\partial f}{\partial x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \tag{3.15}$$

$$g_y = \frac{\partial f}{\partial y} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \tag{3.16}$$

Note that all the operators sum to zero, which implies zero response in an area of constant intensity values, which was required in rule one for derivatives of digital functions.

From these operators we get an estimate of the partial derivatives for x- and y-directions, by convolution, and we can estimate the edge strength and direction for every pixel point. Computing the magnitude can be performed by the formula given above, but because of the computational burden of having to square the partial derivatives and take the square root of the sum, another method can be used to approximate the magnitude. This method uses absolute values to find the magnitude of the gradient:

$$M(x,y) \approx |g_x| + |g_y| \tag{3.17}$$

There are also different ways to modify the different gradient operators. If the interest is of edges in diagonal directions a modified Sobel operator would look like

$$H_x(i,j) = \begin{array}{|c|c|c|}\hline 2 & 1 & 0 \\ \hline 1 & 0 & -1 \\ \hline 0 & -1 & -2 \\ \hline \end{array}, \; H_y(i,j) = \begin{array}{|c|c|c|}\hline 0 & -1 & -2 \\ \hline 1 & 0 & -1 \\ \hline 2 & 1 & 0 \\ \hline \end{array} \tag{3.18}$$

This is an example of a more general "compass operator". We can also expand gradient operators so that they are larger than 3x3 filters, an example of 5x5 operator of a modified Sobel operator is

$$H_x(i,j) = \begin{array}{|c|c|c|c|c|}\hline 1 & 2 & 0 & -2 & -1 \\ \hline 4 & 8 & 0 & -8 & -4 \\ \hline 6 & 12 & 0 & -12 & -6 \\ \hline 4 & 8 & 0 & -8 & -4 \\ \hline 1 & 2 & 0 & -2 & -1 \\ \hline \end{array}, \; H_y(i,j) = \begin{array}{|c|c|c|c|c|}\hline -1 & -4 & -6 & -4 & -1 \\ \hline -2 & -8 & -12 & -8 & -2 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 2 & 8 & 12 & 8 & 2 \\ \hline 1 & 4 & 6 & 4 & 1 \\ \hline \end{array} \tag{3.19}$$

which is formed by convolving the 3x3 Sobel operator by a 3x3 Gaussian filter. The advantage of larger filters is that they are less sensitive to noise. The disadvantage, on the other hand, is that larger filters will smooth out details. Given the small size of our objects, this rules out the use of larger filters. It also rules out the use of more sophisticated edge operators like the Canny operator [1], which may include substantial smoothing.

3.2 Thresholding

Thresholding is divided into two main areas, namely global thresholding and local adaptive thresholding. For global segmentation there is one single threshold for the entire image, while for local adaptive thresholding every pixel will have it's own threshold.

3.2.1 Global thresholding

Global thresholding is the simplest way to divide an image into different regions. Typically the image is divided into foreground and background, resulting in a binary image with pixel values zero and one. Given an input image $f(x, y)$ the output image, after the thresholding, will be given as

$$g(x,y) = \begin{cases} 0, \text{ if } & 0 \leq f(x,y,j) \leq T \\ 1, \text{ if } & T < f(x,y) \leq G-1 \end{cases} \tag{3.20}$$

where G is the number of gray levels in the image and T is the threshold.

But if there are several classes of objects there is possible to divide the image into several regions by multilevel thresholding. For M gray level intervals the segmented image is given by

$$g(x,y) = \begin{cases} 0, \text{ if } & 0 \leq f(x,y,j) \leq t_1 \\ 1, \text{ if } & t_1 < f(x,y) \leq t_2 \\ . & \\ . & \\ . & \\ M-1, \text{ if } & t_m - 1 < f(x,y) \leq G-1 \end{cases} \tag{3.21}$$

3.2.2 Thresholding by minimizing the classification error

It would be tempting to threshold the image where the image-histogram has a dip, as seen in figure 3.1, but this will not always be the best threshold, and it could also be that the histogram doesn't have a dip at all. So we need something

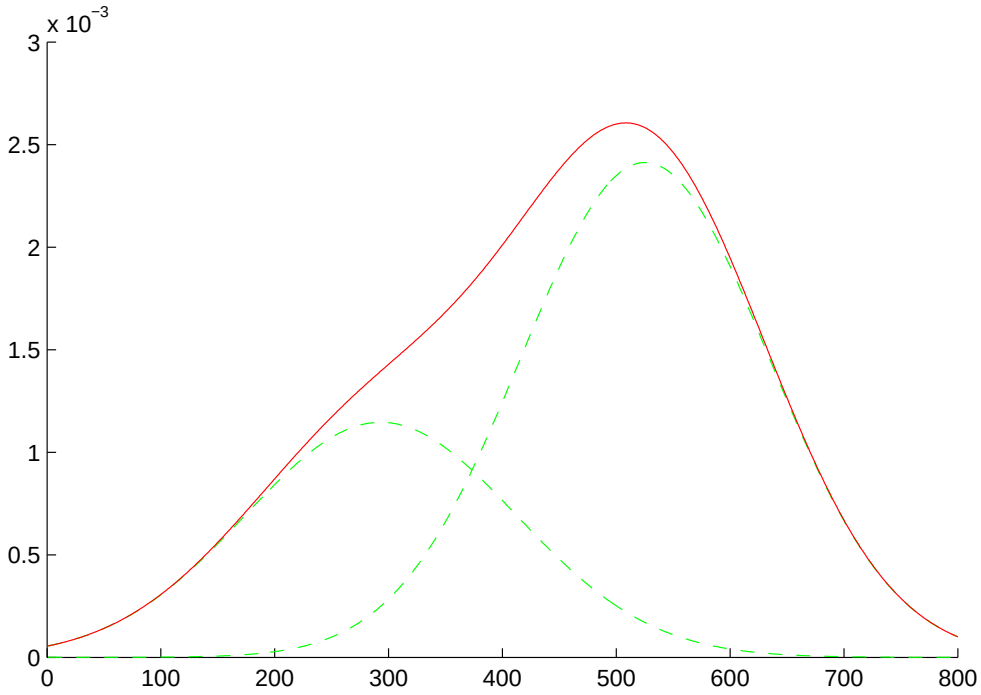


Figure 3.1: *Histograms of image, background and foreground.*

to tell us how good the threshold is, and what the thresholding error is. Because when we threshold the image some of the background pixels will be classified as foreground pixels and vice versa.

Assume that the histogram for the image is a sum of two distributions $b(z)$ and $f(z)$, where b and f are the normalized background- and foreground histograms, respectively. Then we could write the normalized histogram as

$$p(z) = B \cdot b(z) + F \cdot f(z) \tag{3.22}$$

where B and F are the apriori foreground and background probabilities, which sum to one. Then the probability of mis-classifying a pixel, given a threshold t, is given by

$$E_B(t) = \int_{-\infty}^{t} f(z)dz \tag{3.23}$$

$$E_F(t) = \int_{t}^{\infty} b(z)dz \tag{3.24}$$

So the total error we make with a given threshold is

$$E(t) = F \cdot E_B(t) + B \cdot E_F(t) = F \int_{-\infty}^{t} f(z)dz + B \int_{t}^{\infty} b(z)dz \tag{3.25}$$

If we find the derivative of this expression and put the derivative equal to zero we get the threshold which minimizes the error

$$\frac{dE(t)}{dt} = 0 \Rightarrow F \cdot f(T) = B \cdot b(T) \tag{3.26}$$

If we assume that the background and foreground is normally distributed then we can write the normalized histogram as

$$p(z) = Bb(z) + F(f(z) = \frac{B}{\sqrt{2\pi}\sigma_B} e^{-\frac{(z-\mu_B)^2}{2\sigma_B^2}} + \frac{F}{\sqrt{2\pi}\sigma_F} e^{-\frac{(z-\mu_F)^2}{2\sigma_F^2}} \tag{3.27}$$

and we get the threshold that minimizes the error as

$$\frac{B}{\sqrt{2\pi}\sigma_B} e^{-\frac{(T-\mu_B)^2}{2\sigma_B^2}} = \frac{F}{\sqrt{2\pi}\sigma_F} e^{-\frac{(T-\mu_F)^2}{2\sigma_F^2}} \tag{3.28}$$

If we take the logarithm on each side of this equation we get

$$\frac{(T-\mu_F)^2}{2\sigma_F^2} - \ln\left(\frac{F}{\sigma_F}\right) = \frac{(T-\mu_B)^2}{2\sigma_B^2} - \ln\left(\frac{B}{\sigma_B}\right) \tag{3.29}$$

which gives the following second order equation:

$$(\sigma_B^2 - \sigma_F^2)T^2 + 2(\mu_B\sigma_F^2 - \mu_F\sigma_B^2)T + \sigma_B^2\mu_F^2 - \sigma_F^2\mu_B^2 + 2\sigma_B^2\sigma_F^2 \ln\left(\frac{B\sigma_F}{F\sigma_B}\right) = 0 \tag{3.30}$$

If the standard deviations of the two distributions are equal, the equation above will simplify to

$$2(\mu_B - \mu_F)T - (\mu_B + \mu_F)(\mu_B - \mu_F) + 2\sigma^2 \ln\left(\frac{B}{F}\right) = 0 \tag{3.31}$$

$$\Longleftrightarrow T = \frac{\mu_B + \mu_F}{2} + \frac{\sigma^2}{\mu_B - \mu_F} \ln\left(\frac{F}{B}\right) \tag{3.32}$$

And if the apriori probabilities F and B are equal, we get a solution given by

$$T = \frac{\mu_B + \mu_F}{2} \tag{3.33}$$

Ridler and Calvard

From equation (3.33) it follows a simple iterative thresholding algorithm known as Ridler and Caldvard's [19] method. The iterative procedure is done in the following way:

1. Start with threshold t = the global mean value.
2. Then:
   - Calculate the mean value of the pixels with gray level lower than t, written as $\mu_1(t)$
   - Calculate the mean value of the pixels with gray levels higher or equal to the threshold t, written as $\mu_2(t)$
3. The new threshold is given as

$$t = \frac{\mu_1(t) + \mu_2(t)}{2} \tag{3.34}$$

4. Repeat 2 and 3 until the difference between the old and new threshold is less than a small value ϵ.

We have done some assumptions earlier and when those assumptions don't apply the algorithm will break down. This is certainly the case if the apriori class probabilities are very different. This is also the case for the next algorithm we are going to look at, namely Otsu's segmentation.

Otsu's segmentation algorithm

A popular, and often used global segmentation algorithm, is the one proposed by Ostu [16]. This method seeks a threshold by minimizing the within class variances and maximizing the difference between the class means. We shall see later that this is essentially a classification method with similarities to Fisher's LDF.

If we assume normalized histograms defined as before, we then have defined the apriori class probabilities, the mean and class variance in equations (3.43)-(3.45). In order to evaluate how good a certain threshold is, we'll need to define some criterion which measures the class separability:

$$\lambda = \frac{{\sigma^2}_{BE}}{{\sigma^2}_W} \ , \ \kappa = \frac{{\sigma^2}_T}{{\sigma^2}_W} \ , \ \eta = \frac{{\sigma^2}_{BE}}{{\sigma^2}_T} \tag{3.35}$$

where

$${\sigma^2}_W = B{\sigma^2}_B + F{\sigma^2}_F \tag{3.36}$$

$${\sigma^2}_{BE} = B(\mu_B - \mu_T)^2 + F(\mu_F - \mu_T)^2 = BF(\mu_F - \mu_B)^2 \tag{3.37}$$

$${\sigma^2}_T = \sum_{i=0}^{G-1}(i - \mu_T)^2 p(i) \tag{3.38}$$

and

$$\mu_T = \sum_{i=0}^{G-1} ip(i) \tag{3.39}$$

are the within-class variance, between-class variance, total variance and the total mean.

We now have to optimize one of the discriminant criterion functions to find the threshold which separates the classes best. Since there are equivalence between the discriminant functions it is enough to maximize one on them to obtain the threshold, and because of simplicity η is chosen.

The optimal threshold t is the one that maximizes η, which is the same as maximizing σ^2_{BE} since ${\sigma^2}_T$ is constant. By some manipulation of the expressions we can write ${\sigma^2}_{BE}$ as

$${\sigma^2}_{BE}(t) = \frac{[\mu_T B(t) - \mu(t)]^2}{B(t)[1 - B(t)]} \tag{3.40}$$

Then the optimal threshold t^* is the value of t which maximizes ${\sigma^2}_{BE}(t)$, i.e.,

$${\sigma^2}_{BE}(t*) = \max_{0 \le t < G-1} {\sigma^2}_{BE}(t) \tag{3.41}$$

Kittler and Illingswoth's minimum error thresholding

Kittler and Illingworth [10] use another criterion than Otsu's algorithm, and is based on minimizing the Kullback information distance [8]. The optimal threshold for gray level, t, is the one minimizing this criterion function $J(t)$, which could be written as

$$J(t) = 1 + 2[B(t)\ln\sigma_B(t) + F(t)\ln\sigma_F(t)] - 2[B(t)\ln B(t) + F(t)\ln F(t)] \tag{3.42}$$

where

$$B = \sum_{i=0}^{t} p(i) \ , \ F = 1 - B \tag{3.43}$$

$$\mu_B = \frac{1}{B}\sum_{i=0}^{t} ip(i) \ , \ \mu_F = \frac{1}{F}\sum_{i=t+1}^{G-1} ip(i) \tag{3.44}$$

$$\sigma_B^2 = \frac{1}{B}\sum_{i=0}^{t}(i-\mu_B)^2 p(i) \ , \ \sigma_F^2 = \frac{1}{F}\sum_{i=t+1}^{G-1}(i-\mu_F)^2 p(i) \tag{3.45}$$

Thus, Kittler and Illingworth's method does not assume $\sigma_B = \sigma_F$ or $B = F$, but makes truncated estimates of these parameters from the image histogram.

The Expectation-Maximization(EM) Algorithm

The expectation-maximization algorithm is an iterative method to compute the maximum likelihood (ML) estimate for missing or hidden data. In maximum likelihood estimation we wish to estimate the model parameters for the underlying distributions, for which there are no observed data. In our case we want the ML estimates for the parameters of the Gaussian distribution, namely the mean, μ, and standard deviation σ for both foreground and background.

Each iteration consists of two steps

1. The E-step:
   The hidden/missing data are estimated given the observed data and current estimate of the model parameters. This is done by using conditional expectation.

2. The M-step:
   The likelihood function is maximized under the assumption that the hidden/missing data are known.

Convergence is guaranteed since the algorithm increases the likelihood at each iteration.

3.2.3 Adaptive thresholding

In adaptive thresholding one can better handle local variations and contrast in the background. There are several ways of obtaining these thresholds, but the idea is to calculate a threshold in a moving window such that every pixel has a threshold, which is dependent on the neighboring pixels. The windows may be overlapping or non-overlapping, and the size of the window will be of importance. Many methods are developed to find local thresholds [22], some quite easy to implement and other more complex.

Niblack's method

Niblack's method is an example of local adaptive threshold algorithm. This algorithm uses the mean and the standard deviation, computed within a moving (wxw) window. The local threshold value, t, is given by

$$t(i,j) = \mu_W(i,j) + k\sigma_W(i,j) \tag{3.46}$$

The output image is given by

$$g(i,j) = \begin{cases} 0, \text{ if } & f(i,j) \leq t(i,j) \\ 1, \text{ if } & f(i,j) > t(i,j) \end{cases} \tag{3.47}$$

3.3 Selecting a segmentation algorithm

Segmentation is an important part for the overall result of an image analysis experiment, maybe the most important of them all. So choosing an algorithm is a crucial step. As described earlier in this chapter, there are some different alternatives when it comes to segmentation. The first choice is if we are going to use a global or local adaptive segmentation. Just by looking at the images it is somehow obvious that a global segmentation algorithm is going fail, because of the complex structures and local variations within the cell nuclei. Figure 3.2 shows an example of five cell nuclei, the original images in the first row and the same image segmented with Otsu's segmentation algorithm in the second row. And as we can see Otsu's algorithm doesn't capture the fine details inside the cell nucleus, at least not all details. So a local adaptive segmentation routine is certainly needed.

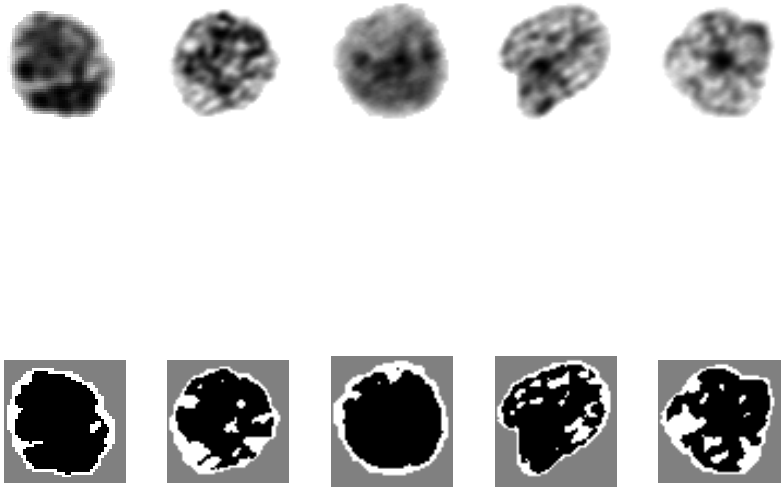


Figure 3.2: *The result of segmentation of five cell nuclei.*

Now, even though we are one step further to a segmentation algorithm, we still have some choices to make, because there are several algorithms to choose from. In the survey of *Trier et al.* [22], where they compared different local adaptive segmentation algorithms (and some global), they found that Niblack's method is

the one that does best in their OCR setting. Another interesting thing they conclude with, is that if the validation step proposed by Yanowitz and Bruckstein [25] is included, it improves all of the algorithms and the difference in performance between the algorithms becomes less apparent. Even though the images used in the *Trier et al* [22] paper are not of the some sort that we are analyzing, there are still some similarities in the need for capturing details and handling of local variations. Settling on Niblack's method, defined in (3.46), gives rise to some other questions;

- What is the optimal value of k?
- How large should the moving (wxw) window be?
- What about two thresholds?

3.3.1 A criterion based on the validation step of Yanowitz and Bruckstein

As stated earlier Yanowitz and Bruckstein [25] proposed a post processing step which removes "ghost" objects. This post processing step removes objects who's average gradient value at the edge of the object is smaller that a chosen threshold T.

To decide the window size, w, and/or k-value that we are going to use, we need a criterion that will separate the different values of w and k and choose the ones that are doing "best". The criterion used is based on the validation step of Yanowitz and Bruckstein [25], with some modification to our purpose. This is done in the following way:

For all w and k

1. Segment the original image with Niblack's method for given window size and k-value
2. For all objects in the segmented image, find the gradient magnitude value averaged over the edge pixels that are 4-connected to the background.
3. Then calculate the average edge gradient magnitude of all objects.

The combination of w and k that gives the highest average object gradient magnitude is chosen. In this way we hope to find the optimal size of the moving window, w, and the optimal constant k.

3.3.2 One threshold

Finding the constant, k, and the window size by the gradient image

The first approach is to do an iterative search in the k-w space to find the value that maximizes the criterion given above. We do this iteratively for w=(5,7,9) and k$\in (0.5, 2)$ for each gray level image.

Finding k from minimizing the classification error

The second approach is to compute the constant, k, from the foreground and background histograms[1], then using this k value and do an iterative search, like the one above, but now only for the window size.

If we assume that the background and foreground are normally distributed, from equation (3.32) we have seen that this threshold is given by

$$T = \frac{\mu_B + \mu_F}{2} + \frac{\sigma^2}{\mu_B - \mu_F} \ln\left(\frac{F}{B}\right) \tag{3.48}$$

We now assume that $\sigma_B = \sigma_F = \sigma > 0$, and that we can write

$$\mu_F = \mu_B + d\sigma \tag{3.49}$$

We then get the threshold, which minimizes the error, given as

$$T = \frac{\mu_B + \mu_B + d\sigma}{2} + \frac{\sigma^2}{\mu_B - \mu_B - d\sigma} \ln\left(\frac{F}{B}\right) = \mu_B + \left[\frac{d}{2} - \frac{1}{d} \ln\left(\frac{F}{B}\right)\right] \sigma \tag{3.50}$$

where d is the difference between μ_F and μ_B, given in terms of σ.

The expression for μ_w and σ_w, given the distribution parameters F, B, μ_F, μ_B and σ will on the average be

$$\mu_w = F\mu_F + B\mu_B \tag{3.51}$$

and

$$\sigma_w^2 = B\sigma_F^2 + F\sigma_F^2 + (\mu_B - \mu_w)^2 + (\mu_F - \mu_w)^2 F = B\mu_B^2 + F\mu_F^2 + \sigma^2 \tag{3.52}$$

Substituting $\mu_F = \mu_B + d\sigma$ and solving for μ_B and σ we get

$$\mu_B = \mu_w - Fd\sigma \tag{3.53}$$

[1]This is based on an idea and some simple simulations by F.Albregtsen (private communication, 2009) showing that we can compute the optimal k-values, provided that we can estimate the histogram distribution parameters properly.

$$\sigma^2 = \frac{\sigma_w^2}{1 - d^2(F^2 - F)} \tag{3.54}$$

Putting this into the expression for the adaptive threshold(3.78), we get

$$T = \mu_B + \left[\frac{d}{2} - \frac{1}{d}\ln\left(\frac{F}{B}\right)\right]\sigma = \mu_w + \left[\frac{d\left(\frac{1}{2} - F\right) - \frac{1}{d}\ln\left(\frac{F}{1-F}\right)}{\sqrt{1 - d^2(F^2 - F)}}\right]\sigma_w \tag{3.55}$$

If we estimate F and d from the images, under the assumption of equal standard deviations, the k-value for Niblack's method gives the threshold $T = \mu_W + k\sigma_W$ that will minimize the error we make in the segmentation, and as we can see from table 3.1, with simulated values, the optimal k will vary a lot for different values of d and F.

d/F	0.5	0.25	0.125	0.0625	0.03125	0.015625
0.5	0.00	2.27	4.02	5.59	7.08	8.51
1	0.00	1.24	2.20	3.06	3.84	4.59
2	0.00	0.79	1.44	2.01	2.51	2.95
4	0.00	0.64	1.20	1.74	2.24	2.66

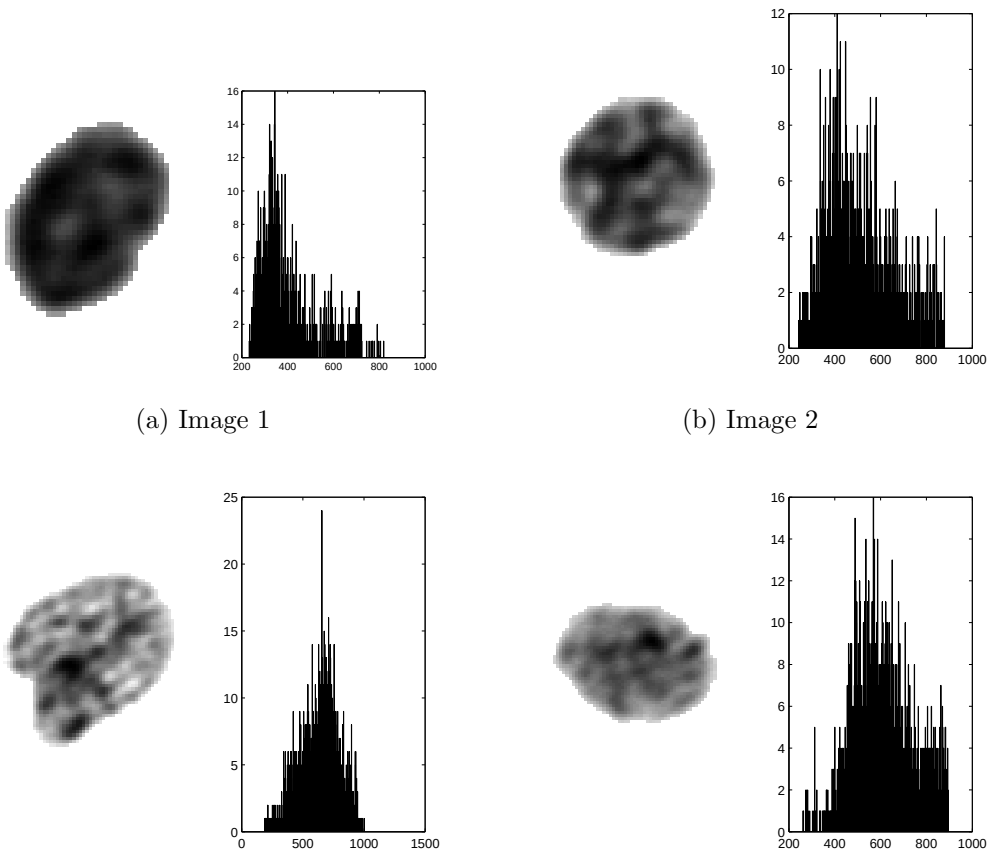
Table 3.1: *Table with optimal k-values for simulated values of d and F*

So to calculate k we need to find the apriori probability for the foreground and the mean values, μ_F and μ_B, for the foreground and background. This has been done in two different ways. First we tried Kittler and Illingworth's method [10] which will give an approximation to the parameters needed, based on truncated foreground and background distributions. In the second approach we have used the EM-algorithm to find the parameters.

Summing it up we now got three different methods to find k and w, i.e., an iterative search in the k-w space, finding k from a normal approximation of the histograms using Kittler and Illingworth's method [10] or the EM-algorithm to get the parameters needed to calculate k.

Some test images

In the beginning of this project I have worked with only four different cell nuclei images, in order to have a good overview over the methods as they develop and to see how the choices made are influencing the result in each step. As methods and algorithms converge, tests with larger sets of images are done. Before I decided which images that I would use in the pilot tests, I looked through a lot of them and tried to find four images that were different in structure and size, such that the methods could handle variations. In figure (3.3) the four images used in the pilot experiments are shown along with the graylevel histograms.



(a) Image 1

(b) Image 2

(c) Image 3

(d) Image 4

Figure 3.3: *The four test images with histograms of the gray levels*

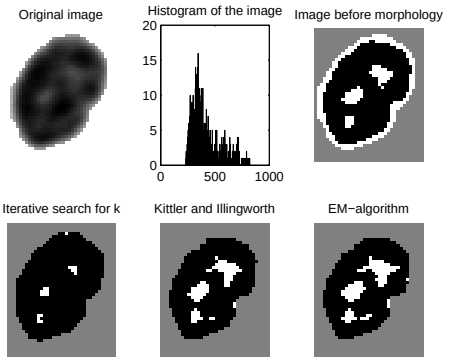
Some results with one threshold

In figure 3.4 these test images are segmented with one threshold and organized in the following way:

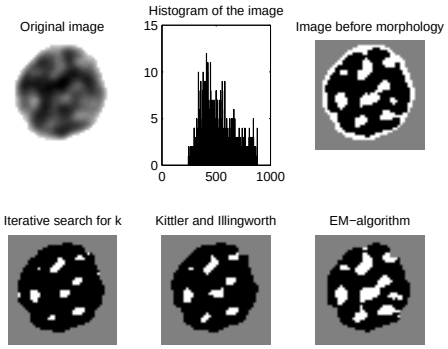
- In the top left corner of the four figures is the original gray level image.
- Then the histogram of the image.
- The upper right image is the segmented image with fixed window size and k-value (w=9 and k=1), but without any morphology
- The image to the left in the bottom row is the result of an iterative search for w and k.
- In the middle of the bottom row is the segmented image with k found by min error and the parameters determined by Kittler and Illingworth's method.

The window size is found by using our criterion with an iterative search for different w-values.

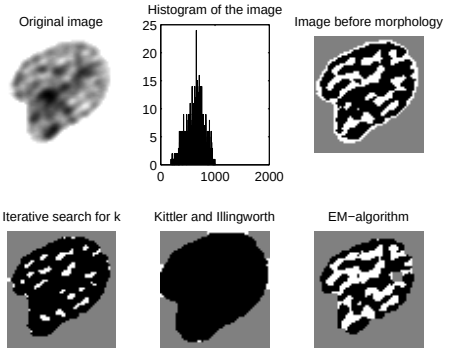
- The bottom right image is the segmented image with k found by min error, and the parameters determined with the EM-algorithm. The window size is found by using our criterion with an iterative search for different w-values.



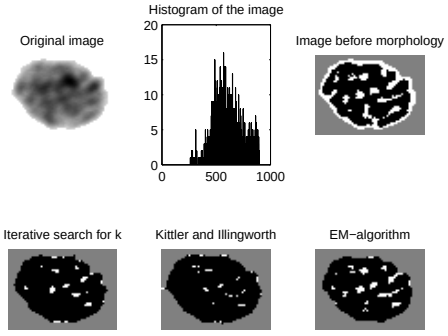
(a) Image1



(b) Image 2



(c) Image3



(d) Image 4

Figure 3.4: *The four test images segmented with one threshold.*

3.3.3 Two thresholds

What about two thresholds?

We would be interested in both bright and dark objects, and the question then is: Is one threshold sufficient? Will one threshold capture and represent the structures inside the cell nuclei in a satisfying way?

From the results in the last section the need for two thresholds can indeed be justified. We then have to extend Niblack's method to two thresholds. But again some questions need answers. Should the k values for the two thresholds be equal such that the threshold for dark objects is given as

$$t_d(x,y) = \mu_w(x,y) - k\sigma_w(x,y) \tag{3.56}$$

and for bright objects

$$t_b(x,y) = \mu_w(x,y) + k\sigma_w(x,y) \tag{3.57}$$

Or should there be different k values for the two thresholds? Thus for different k's the threshold for dark objects is given as

$$t_d(x,y) = \mu_w(x,y) - k_d\sigma_w(x,y) \tag{3.58}$$

and for bright objects as

$$t_b(x,y) = \mu_w(x,y) + k_b\sigma_w(x,y) \tag{3.59}$$

where w is the size of the moving window for which the mean, μ_W, and standard deviation, σ_W, are calculated and k_i is a constant. The output image for both methods is then given as

$$g(x,y) = \begin{cases} 0, \text{ if} & f(x,y) \leq t_d(x,y) \\ 128, \text{ if} & t_d(x,y) < f(x,y) \leq t_b(x,y) \\ 255, \text{ if} & f(x,y) > t_b(x,y) \end{cases} \tag{3.60}$$

Finding the constant, k, and the window size

To find the k-values and window size we use the same methods as descried in sections 3.3.2, and again we have three different algorithms to choose from.

Iterative search

As for the case with one threshold we can do an iterative search in the k-w space to find the value that maximizes the criterion given in section 3.3.1. We do this iteratively for w=(5,7,9) and k$\in$ $(0.5, 2)$. This is also done in two different ways, for k equal for both thresholds, and for different k's for the two thresholds.

Finding k from minimizing the classification error with normal approximation

The second approach is to find the constant, k, based on image histograms, then using this k-value and do an iterative search for the window size, in the same manner as in section 3.3.2, but for different k-values for bright and dark objects.

From equation(3.55) we have that

$$T = \mu_B + \left[\frac{d}{2} - \frac{1}{d}\ln\left(\frac{F}{B}\right)\right]\sigma = \mu_w + \left[\frac{d\left(\frac{1}{2} - F\right) - \frac{1}{d}\ln\left(\frac{F}{1-F}\right)}{\sqrt{1 - d^2(F^2 - F)}}\right]\sigma_w \tag{3.61}$$

This will only give us one k-value, and not one for bright and one for the dark objects. But we could do the same calculations as for one threshold, thus we have to expand our normalized histogram to contain three instead of two distributions, and we get

$$p(z) = D \cdot d(z) + G \cdot g(z) + B \cdot b(z) \tag{3.62}$$

where D and G and B are the apriori dark, gray and bright probabilities, which sum to one. Then the probability to mis-classify a pixel, given the thresholds $t = t_1, t_2$, is given by

$$E_b(t) = \int_{-\infty}^{t_2} b(z)dz \tag{3.63}$$

$$E_g(t) = \int_{-\infty}^{t_1} g(z)dz + \int_{t2}^{\infty} g(x)dz \tag{3.64}$$

$$E_d(t) = \int_{t_1}^{\infty} d(z)dz \tag{3.65}$$

So the total error we make with a given threshold is then

$$E(t) = B \cdot E_b(t) + G \cdot E_g(t) + D \cdot E_d(t) \tag{3.66}$$

If we find the partial derivatives of this expression and put the derivatives equal to zero we get the thresholds which minimize the error

$$\frac{\partial E(t)}{dt_1} = 0 \Rightarrow D \cdot d(T_1) = G \cdot g(T_1) \tag{3.67}$$

and

$$\frac{\partial E(t)}{dt_2} = 0 \Rightarrow B \cdot b(T_2) = G \cdot g(T_2) \tag{3.68}$$

If we assume that the three distributions are normally distributed then we can write the normalized histogram as

$$p(z) = \frac{D}{\sqrt{2\pi}\sigma_d}e^{-\frac{(z-\mu_d)^2}{2\sigma_d^2}} + \frac{G}{\sqrt{2\pi}\sigma_g}e^{-\frac{(z-\mu_g)^2}{2\sigma_g^2}} + \frac{B}{\sqrt{2\pi}\sigma_b}e^{-\frac{(z-\mu_b)^2}{2\sigma_b^2}} \tag{3.69}$$

and we get the thresholds that minimizes the errors as

- $\mathbf{T_1}$:

$$\frac{D}{\sqrt{2\pi}\sigma_d}e^{-\frac{(T_1-\mu_d)^2}{2\sigma_d^2}} = \frac{G}{\sqrt{2\pi}\sigma_g}e^{-\frac{(T_1-\mu_g)^2}{2\sigma_g^2}} \tag{3.70}$$

- $\mathbf{T_2}$:

$$\frac{G}{\sqrt{2\pi}\sigma_g}e^{-\frac{(T_2-\mu_g)^2}{2\sigma_g^2}} = \frac{B}{\sqrt{2\pi}\sigma_b}e^{-\frac{(T_2-\mu_b)^2}{2\sigma_b^2}} \tag{3.71}$$

Using the same type of arguments as for one threshold, we then get thresholds as

$$T_1 = \frac{\mu_g + \mu_d}{2} + \frac{\sigma^2}{\mu_g - \mu_d}\ln\left(\frac{D}{G}\right) \tag{3.72}$$

$$T_2 = \frac{\mu_g + \mu_b}{2} + \frac{\sigma^2}{\mu_b - \mu_g}\ln\left(\frac{B}{G}\right) \tag{3.73}$$

Under the assumption $\sigma_d = \sigma_g = \sigma_b = \sigma > 0$, we can write

$$\mu_d = \mu_g - d_1\sigma \tag{3.74}$$

and

$$\mu_b = \mu_g + d_2\sigma \tag{3.75}$$

where d_i are the Mahalanobis distances between the distributions. We then get the thresholds, which minimize the error, given as

$$T_1 = \frac{\mu_g + \mu_g - d_1\sigma}{2} + \frac{\sigma^2}{\mu_g - \mu_g + d_1\sigma}\ln\left(\frac{D}{G}\right) = \mu_g + \left[\frac{1}{d_1}\ln\left(\frac{D}{G}\right) - \frac{d_1}{2}\right]\sigma \tag{3.76}$$

$$T_2 = \frac{\mu_g + \mu_g + d_2\sigma}{2} + \frac{\sigma^2}{\mu_g + d_2\sigma - \mu_g}\ln\left(\frac{B}{G}\right) = \mu_g + \left[\frac{1}{d_2}\ln\left(\frac{B}{G}\right) + \frac{d_2}{2}\right]\sigma \tag{3.77}$$

which both resemble the Niblack equation

$$T(x,y) = \mu_w(x,y) + k\sigma_w(x,y) \tag{3.78}$$

The expression for μ_w and σ_w, given the distribution parameters D, G, B, μ_d, μ_g, μ_b and σ, will on the average be

$$\mu_w = D\mu_d + G\mu_g + B\mu_b \tag{3.79}$$

and

$$\sigma_w^2 = D\sigma_d^2 + G\sigma_g^2 + B\sigma_b^2 + D(\mu_d - \mu_w)^2 + G(\mu_g - \mu_w)^2 + B(\mu_b - \mu_w)^2 \tag{3.80}$$

Substituting $\mu_d = \mu_g - d_1\sigma$ and $\mu_b = \mu_g + d_2\sigma$ and solving, we get

$$\mu_w = D\mu_d + G\mu_g + B\mu_b = D(\mu_g - d_1\sigma) + G\mu_g + B(\mu_g + d_2\sigma) \tag{3.81}$$

$$= (D + G + B)\mu_g + (Bd_2 - Dd_1)\sigma = \mu_g + (Bd_2 - Dd_1)\sigma \tag{3.82}$$

$$\Rightarrow \mu_g = \mu_w + (Dd_1 - Bd_2)\sigma \tag{3.83}$$

From equation (3.80)

$$\sigma_w^2 = \sigma^2 + D(\mu_d - \mu_w)^2 + G(\mu_g - \mu_w)^2 + B(\mu_b - \mu_w)^2 \tag{3.84}$$

If we calculate the squared expressions separately and using the relations in equations (3.74), (3.75) and (3.82) and substitute we get

$$\mu_d - \mu_w = \mu_g - d_1\sigma - [\mu_g + (Bd_2 - Dd_1)\sigma] = [d_1(D-1) - Bd_2]\sigma \tag{3.85}$$

$$\mu_g - \mu_w = \mu_g - \mu_g + (Bd_2 - Dd_1)\sigma = [Bd_2 - Dd_1] \tag{3.86}$$

$$\mu_b - \mu_w = \mu_g + d_2\sigma - [\mu_g + (Bd_2 - Dd_1)\sigma] = [d_2(1-B) + Dd_1]\sigma \tag{3.87}$$

This gives

$${\sigma_w}^2 = \sigma^2\{1 + D[d_1(D-1) - Bd_2]^2 + G[Bd_2 - Dd_1]^2 + B[d_2(1-B) + Dd_1]^2\} \tag{3.88}$$

$$= \sigma^2\{1 + D[(D-1)^2{d_1}^2 - 2B(D-1)d_1d_2 + B^2{d_2}^2] + \tag{3.89}$$

$$G[B^2{d_2}^2 - 2BDd_1d_2 + D^2{d_1}^2] + \tag{3.90}$$

$$B[(1-B){d_2}^2 + 2(1-B)Dd_1d_2 + D^2{d_1}^2\} \tag{3.91}$$

$$= \sigma^2\{1 + {d_1}^2[D(D-1)^2 + GD^2 + BD^2] + \tag{3.92}$$

$$d_1 d_2[2B(1-B)D - 2GBD - 2BD(D-1)] + \tag{3.93}$$

$$d_2{}^2[DB^2 + GB^2 + B(1-B)^2]\} \tag{3.94}$$

$$= \sigma^2 C \Rightarrow \sigma = \frac{\sigma_w}{\sqrt{C}} \tag{3.95}$$

Putting this into the expression for the adaptive thresholds (3.72) and (3.73), we get the thresholds

$$T_1 = \mu_w + \left[\frac{Dd_1 - Bd_2}{\sqrt{C}}\right]\sigma_w + \left[\frac{(\frac{1}{d_1}\ln\left(\frac{D}{G}\right) - \frac{d_1}{2}}{\sqrt{C}}\right]\sigma_w \tag{3.96}$$

$$= \mu_w + \left[\frac{Dd_1 - Bd_2 + \frac{1}{d_1}\ln\left(\frac{D}{G}\right) - \frac{d_1}{2}}{\sqrt{C}}\right]\sigma_w \tag{3.97}$$

$$T_2 = \mu_w + \left[\frac{Dd_1 - Bd_2}{\sqrt{C}}\right]\sigma_w + \left[\frac{\frac{1}{d_2}\ln\left(\frac{B}{G}\right) + \frac{d_2}{2}}{\sqrt{C}}\right]\sigma_w \tag{3.98}$$

$$= \mu_w + \left[\frac{Dd_1 - Bd_2 + \frac{1}{d_2}\ln\left(\frac{B}{G}\right) + \frac{d_2}{2}}{\sqrt{C}}\right]\sigma_w \tag{3.99}$$

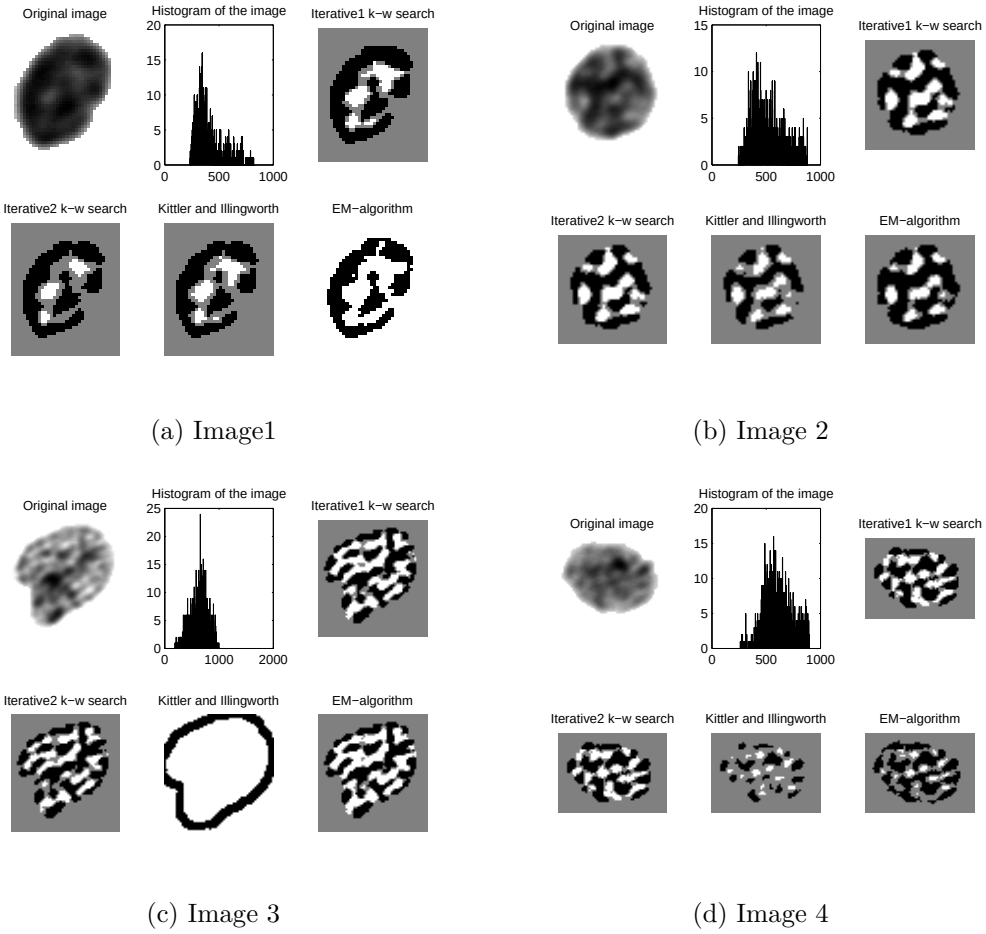
So to find the k_i's we need to find the apriori probabilities and the mean for the three truncated distributions, which can be done with Kittler and Illingworth's method [10] or the EM-algorithm.

Some test images

Below is the result from our different methods on 4 images, each series of images is organized in the following way:

- In the top left corner of the four figures is the original gray level image.
- Then the histogram of the image.
- The upper right image is the segmented image by using the iterative search for k and w, with one k-value.
- The image to the left in the bottom row is also done by an iterative search for w and k, but with different k-values for bright and dark objects.
- In the middle of the bottom row is the segmented image with k found by min error and the parameters determined by Kittler and Illingworth's method. The window size is found by using our criterion with an iterative search for different w-values. Note that this method gives only one value for k.

- The bottom right is the segmented image with k-values found by min error, and the parameters determined with the EM-algorithm. The window size is found by using our criterion with an iterative search for different w-values.



(a) Image1

(b) Image 2

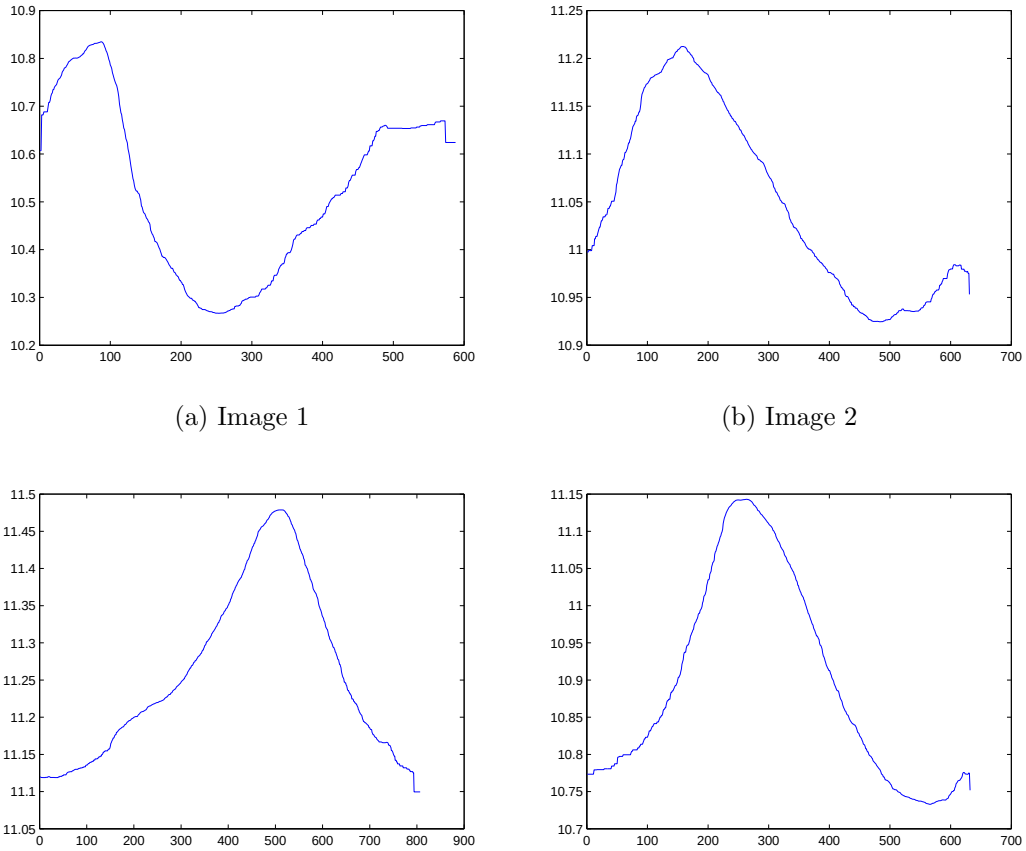
(c) Image 3

(d) Image 4

Figure 3.5: *The four test images segmented with two threshold.*

3.3.4 Some comments on Kittler and Illingworth's method

As seen in figure 3.4(c) and 3.5(c), the Kittler and Illingworth's method collapses and the segmentation becomes meaningless. And from experimenting with other images this happens now and then. A closer look at the histograms of these images shows that when the images have unimodal histograms, the function J(t) in Kittler and Illingworth's algorithm [10] does not have a minimum, or a global minimum that is. This is illustrated in figure 3.6 where the J(t) function in Kittler and Illingworth's method [10] is plotted for the four test images. For Image 3 this means that J(t) will have a minimum in the beginning or at the end



(a) Image 1

(b) Image 2

(c) Image 3

(d) Image 4

Figure 3.6: J(t) function for the four test images.

of the gray-level distribution, which leads to huge differences in the parameters in Kittler and Illingworth's method [10] and then at last a meaningless k-value. But there are ways to deal with this problem.

Determination of bimodality

To determine if the Kittler and Illingworth's method [10] is going to give meaningful results, we will have to use a bimodality criterion. Demirkaya and Asyiali [3] proposes the maximum of the between class variance, $B_{max}(t)$, of Otsu's method [16] as such a criterion, with

$$B(t) = \frac{\sigma_B^2(t)}{\sigma_{TOT}^2(t)} \tag{3.100}$$

where σ_B^2 is the between class variance and σ_{TOT}^2 is the total variance. Demirkaya and Asyali [3] then claim that if the bimodality criterion $B_{max}(t) > 0.65$ the Kittler and Illingworth method will show good results. But in our experience with

different apriori class probabilities this doesn't seem to hold. But if we use the Mahalanobis distance as a bimodality criterion we get meaningful minima of J(t) when this distance is bigger than 2. Note that we shouldn't use the aposteriori class parameters as given in the function J(t) since they are truncated and overlapping. But we could use the parameters from the EM-algorithm.

3.3.5 Choosing a segmentation algorithm

We have to make a choice on which segmentation method we are going to use. From the last sections there are many reasons not to use Kittler and Illingworth's method [10].

- We have to test for bimodality ⇒ we have to use the EM-algorithm, which gives the same parameters we are looking for.
- If Mahalanobis distance larger than 2 ⇒ Otsu's method [16]

From this Kittler and Illingworth's method [10] is ruled out.

The difference in segmentation results between the iterative method and the method which uses the EM-algorithm is not that clear. Another issue is time, with over hundred patients and about 300 images per patient, time will be of importance. And in the temporal perspective the EM-algorithm is superior. While the iterative method uses about four minutes per image, the method that uses the EM-algorithm takes under 30 seconds. So the obvious choice will be the normal approximation with the parameters from the EM-algorithm, when time and segmentation results is considered.

Summing up we want a segmentation algorithm that separates well between dark and bright object, which means that we will need an algorithm with 2 thresholds. Since there is significant difference between the bright and dark objects in size and area, it will be wise to use two different k-values in Niblack's method. Using the EM-algorithm to get the parameters needed for calculating the k_i's, and use an iterative search for optimal window size, we know have everything we need to segment the images.

The next natural step is morphology, we have already used some morphology in the segmentation algorithm to remove some unwanted artefacts on the border of the cell nuclei, but now we will use it to separate the segmented objects inside the cell nuclei.

Chapter 4

Morphology

Morphology, or mathematical morphology, is based on mathematical operators to manipulate the shape or understand the structure of connected pixels in digital images. Morphological methods are based on set-theory and play an important part in many digital image processing applications, such as object recognition and computer vision.

The methods were at first intended for use on binary images, but the theory has been developed for use in grayscale images and even color images. But in this thesis we will only focus on binary morphology and the simplest of the operators. For a fuller description and alternative formulations with proofs see the article of *Haralick et al.*[6] or the textbook of Gonzalez and Woods[5].

4.1 Some set-theory

Let A and B be sets in $\mathbb{Z}^2$, then

- If a point $a = (a_1, a_2)$ is a member of A, then this is written as: $a \in A$
- If a is not an element in A, this is denoted as $a \notin A$
- The empty set is denoted as $\emptyset$
- If a set B is a part of A, then B is called a subset of A and denoted as: $A \subseteq B$
- **Union** of the sets A and B, denoted as $A \cup B$, is the set of all points which are elements of both A and B.
- **Intersection** of the sets A and B, denoted $A \cap B$, is the set of all points which are in both A and B
- **Complement** of the set A, denoted A^c, is defined as all the points that are not elements of A.

4.2 Structure elements

All morphological operators are based on evaluating subsets of connected pixels in an image. This subset is determined by a structure element, which is a small matrix usually having binary elements, but not always.

Structure elements can have different sizes and shapes and always have an origo which determines the position of the output to the resulting image. Origo can be anywhere in the structure element, even outside, but usually is in the center position of the structure element. Even though a structure element usually contains only binary values, which is called a flat structure element, the origo can have other values and is then a non-flat element. In figure 1.1 there are some examples of different structure elements, some flat and some non-flat.

In a binary image with connected regions of pixels there will be three different ways a structure element can overlap the regions.

- There will be areas in the image where there is no overlap of the connected pixels and the structure element.
- There will be areas where the structure element partly overlaps a connected region, i.e., it hits the object.
- Three will be areas where the whole structure element is inside an object, i.e., it fits the object.

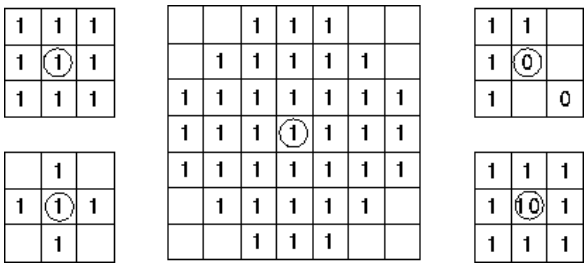


Figure 4.1: *Some different structure elements.*

4.3 The basic operators

Erosion and dilation are the two basic operations of mathematical morphology, and many of the other morphological operations can be broken down to these two operators.

4.3.1 Erosion

Given a set f and a structure element S, an erosion of f with S is defined as the position of all pixels x such that S is included in f when origo of S is at x. This is denoted as

$$f \ominus S = \{x | S_x \subseteq f\}$$

In other words this means that when S is placed in f, such that origo is in pixel (x, y), the output image, g, is then

$$g(x, y) = \begin{cases} 1, \text{ if } S \text{ fits f} \\ 0, \text{ otherwise} \end{cases} \tag{4.1}$$

Figure 4.2 shows an example of erosion of a binary image with a flat 3x3 square structure element. The result of an erosion will shrink the object.

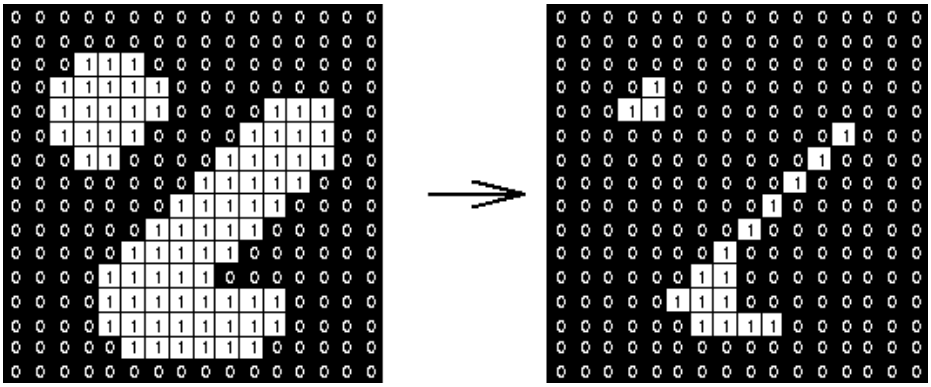


Figure 4.2: *The result of erosion with a 3x3 flat square structure element*

4.3.2 Dilation

Dilation of a set f with a structure element S is defined as the position of all pixels x such that S overlaps with at least one pixel in f when origo is placed at x

$$f \oplus S = \{x | S_x \cap f \neq \emptyset\}$$

This operator gives an output image, g given as

$$g(x,y) = \begin{cases} 1, \text{ if } S \text{ hits f} \\ 0, \text{ otherwise} \end{cases} \tag{4.2}$$

In figure 4.3 there is an example of dilation with the same image and structure element as in figure 4.2. As we can see, this has the opposite effect of erosion and will expand the object.

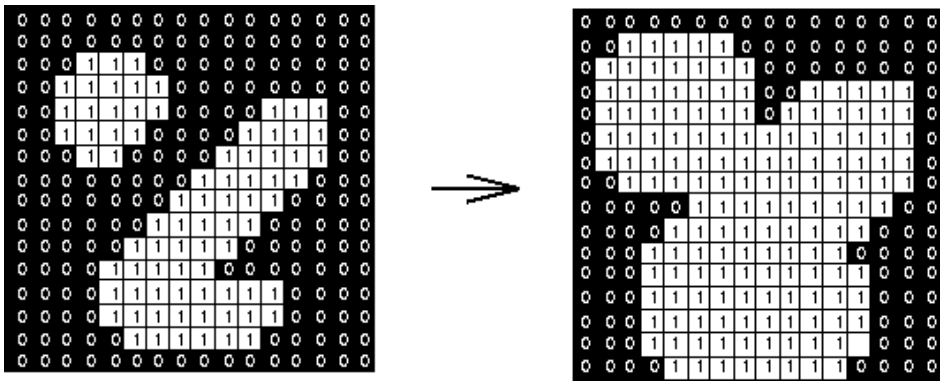


Figure 4.3: *The result of dilation with a 3x3 square structure element.*

4.4 Other operators

There exist a lot of different morphological operators and we are going to look at two of these, and as we will see we can break them down to the fundamental operators.

4.4.1 Opening

Morphological opening is an operator which will remove small bridges between structures in the image and will smooth the contour of the objects. A morphological opening, denoted as $f \circ S$, is just an erosion followed by a dilation

$$f \circ S = (f \ominus S) \oplus S$$

Figure 4.4 shows an example of opening with a 3x3 structure element, notice how this operator smooths out and changes the contour of the smallest object.

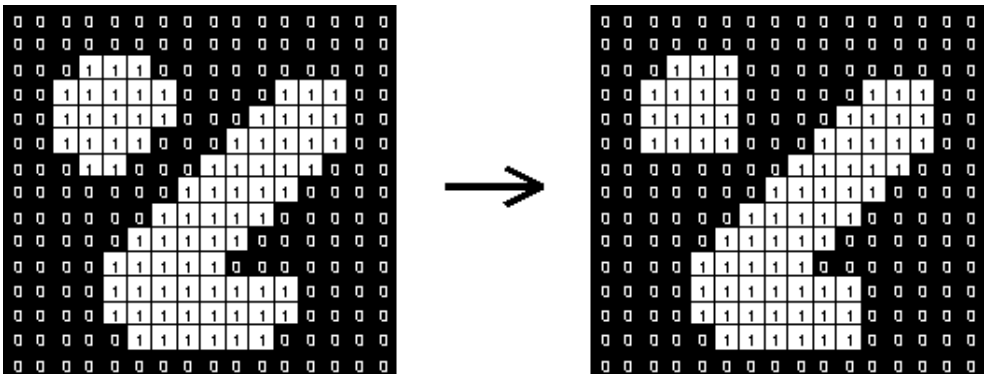


Figure 4.4: *The result of opening with a 3x3 square structure element.*

4.4.2 Closing

Morphological closing will have the opposite effect of opening, i.e., this operator will connect object with small gaps. A morphological closing, denoted as $f \bullet S$, is just a dilation followed by a erosion.

$$f \bullet S = (f \oplus S) \ominus S$$

Figure 4.5 shows how morphological closing will change the structures inside an image.

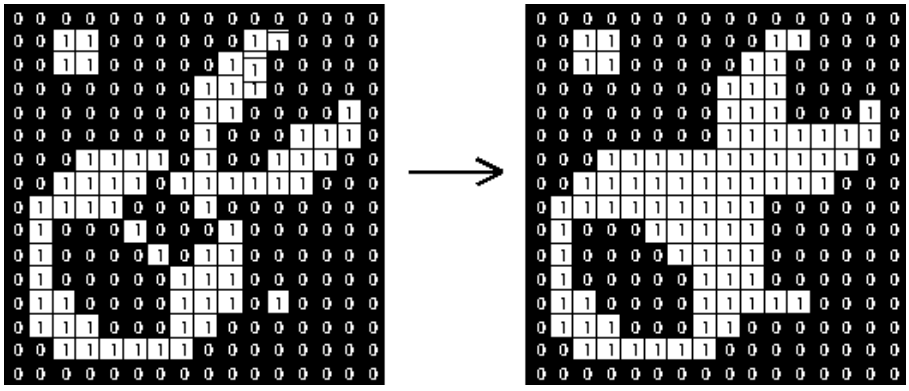


Figure 4.5: *The result of closing with a 3x3 square structure element.*

4.5 Separating the objects

In a project like this we would like to have well defined objects, and if a region of pixels contains more than one object we want to separate them and later find some characteristics describing each of them. Our eyes are excellent to distinguish between such objects and it is easy for us to say where the objects should be separated. But in an automated algorithm that we are trying to make this is not that simple, and we will have to use morphology to do it.

Another problem that occurs, is how the computer should understand that a region of pixels should be divided into two or more objects. A way to solve this is to calculate the solidity of the objects, and use this information to decide if a region consist of more than one object. Solidity is defined as the proportion of pixels in the convex hull of a region that are also in the region, and is defined from zero to one. Of course it would be meaningless to have an algorithm which would split every region into multiple objects with the criterion that the solidity is less than one, so a threshold must be found. After some testing, a solidity less than 0.80 seems like a good criterion to split a region.

And finally we have to choose a structure element. We would like to separate the object, but we also want to keep the object as "real" as possible. The problem with morphology is that it often removes parts of the objects in the separating process. A too large structure element could easily remove big parts of objects or even the entire object. A problem with the images in this project is that they have very low resolution, which means that objects don't contain that many pixels. A 3x3 structure element, which is often used, could "eat" up much of the

information about the objects, so we have chosen to use linear structure element.

4.5.1 Linear structure elements

In this project we will not use square structure elements, because these elements most likely will be to rough for the low resolution cell nuclei images. We will instead use linear structure elements and rotate them to separate the objects, while still not removing to much of them. Later we will give a full description on how the splitting of objects is done, but as an example a linear structure element of size 3, and how it rotates, is given below.

$$\begin{array}{|c|c|c|}\hline 0&0&0\\\hline 1&1&1\\\hline 0&0&0\\\hline\end{array}\ ,\ \begin{array}{|c|c|c|}\hline 0&0&1\\\hline 0&1&0\\\hline 1&0&0\\\hline\end{array}\ ,\ \begin{array}{|c|c|c|}\hline 0&1&0\\\hline 0&1&0\\\hline 0&1&0\\\hline\end{array}\ ,\ \begin{array}{|c|c|c|}\hline 1&0&0\\\hline 0&1&0\\\hline 0&0&1\\\hline\end{array} \tag{4.3}$$

The morphological algorithm is iterative and includes the following steps

1. Divide the segmented image into two subimages, one with bright objects and one with dark objects as shown in figure 4.6.

2. Label the objects and find the solidity of each object.

3. If an objects solidity is below some constant, c, and the object is larger than a certain size: Split the object into two or more objects.

4. Go back to step 2 until all objects have solidity greater than c or are smaller than the chosen size.

(a) Original image (b) Dark objects (c) Bright objects

Figure 4.6: *Image 3 divided into dark and bright labeled objects*

A detailed description of the morphological operations

The previous step-wise description of the iterative morphological algorithm didn't give any details on how the objects where separated. Here we will give a thorough description on how this algorithm actually works, illustrated with images. Assuming we now have a segmented image with only bright or dark objects, which are labeled, the splitting process is as follows:

1. For each object check if the solidity is higher than 0.80 and contains more than a certain number of pixels, if not split the object into two or more. Figure 4.7(a) shows the first object that is checked from figure 4.6(b), but solidity is 0.8519 and nothing is done. The object in (b) in the same figure has solidity less than 0.80 and has to split.

2. The object is eroded with a linear structure element of size two. If the objects in the region are not separated, the linear element is rotated, as shown in (4.3) above. The result after erosion is illustrated in figure 4.7(c).

3. If the structure element is too small to divide the region into several objects, then the length of the element is increased by one and we are back at step 2.

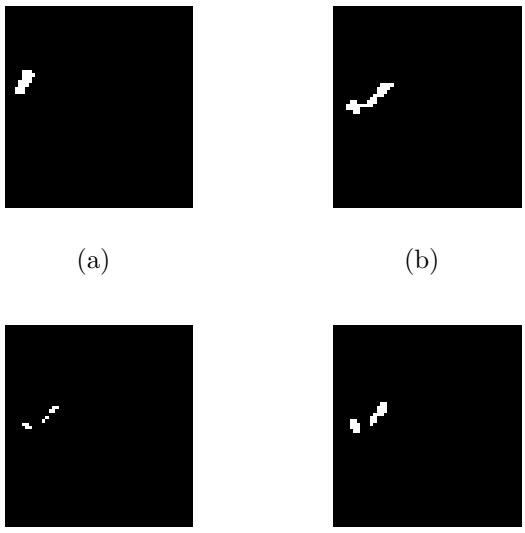
4. After the object is split then the image is dilated with the same rotated structure element the same number of times as it was eroded, and we now have two or more objects. Figure 4.7(d) show the two objects after dilation.

5. Some objects are more problematic than others, and if and object is eroded enough, it will disappear. If that is the case the original object is saved and later put back into the image.

6. When an object is split, the process starts from step one again and is completed when all objects fulfill the solidity and size criterion.

7. Finally we use morphological opening to remove small object and gaps which the iterative algorithm couldn't handle.

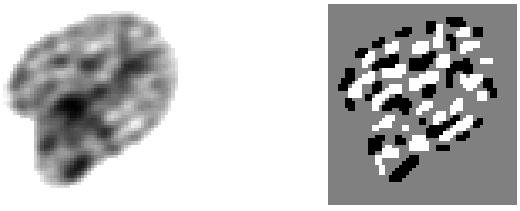
Figure 4.8 shows the original Image 3 in (a) and the image after segmentation and morphology in (b). The next step will be to extract some features which describe the objects and then classify the images based on the information extracted from the objects.



(a) (b)

(c) (d)

Figure 4.7: *Morphology on Image 3.*



(a) (b)

Figure 4.8: *Morphology on Image 3.*

Chapter 5

Features and Classification

In this chapter we will describe the last steps in our algorithm, namely features, or object descriptors, and classification. There is a lot of theory about these subjects and we will go briefly through some of it, and the choices made in our algorithm.

Features

The goal of an image analysis task is in the end to classify an image, or the objects within it, into one of several classes. We have a number of patients belonging to two classes: good and bad prognosis. For each of the patients we have about 300 cell images. Each of the images contains a single cell nucleus.

After the segmentation and morphology are done, we have two labeled bitmaps per image, giving the pixels belonging to dark and bright objects, which are the basic texture structures of the cell nuclei. As with the morphology we will handle the dark and bright objects separately.

5.1 Object descriptors

In the literature there exist descriptions of a lot of object descriptors that could be extracted, see [5], [8], [23]. We could certainly generate a long list of features describing the structures, and then perform some type of feature dimensionality reduction in order to end up with a low dimensional set of features. However, we do not want to have too many feature candidates to choose from, as this would only increase the risk of selecting seemingly useful but actually useless features, given the limited number of samples available [20]. We therefor limit ourselves to intuitively useful features that may contribute to separation of the two classes.

Object descriptors can be very intuitive and simple, e.g., area and perimeter of an object, but a feature could also be very complex and not that intuitive, e.g., fourier descriptors [5].

But because of the low resolution images, the segmented structures inside the cell nuclei will be small and of a limited number of pixels. It therefor seems reasonable to use simple features to describe the objects. From each of the bright and dark objects we have chosen to extract the following area-related and shape-related features:

1. The area of the object
2. The relative area of the object (relative to the area of the nucleus)
3. Compactness
4. Eccentricity
5. Orientation relative to radial direction

Our main attention so far has been on the object structures inside the cell nuclei. But our aim is to correctly classify each cell nucleus image, and in the end classify the patients. So it might also be useful to extract information about each of the cell nuclei as well, such as:

1. Area
2. Compactness
3. Eccentricity
4. Mean gray level
5. Variance of gray level
6. The number of dark and bright objects

We note that the coordinates of the center of mass and perimeter length for both the cell and the objects will be stored. These will not be used directly as features, but will be useful when computing the object features.

5.1.1 Moments

Most of the features used to describe the objects and the cell nuclei can be derived from moments. Regular moments of order (p+q) is defined as [5]:

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y) \tag{5.1}$$

If we move the origin to the center of gravity we get the central moments [5], given as

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y) \tag{5.2}$$

where

$$\bar{x} = \frac{m_{10}}{m_{00}} \ , \ \bar{y} = \frac{m_{01}}{m_{00}} \tag{5.3}$$

and the first order regular moments give the center of mass. With these moments we now have all we need to defined most of the features that we are going to use.

- Area:

  For binary images the area of an object in a sub-image is defined as the zero order moment, i.e.,

  $$A = m_{00} = \sum_x \sum_y f(x, y) \tag{5.4}$$

  which is just the sum of all object pixels. This will give us the area of the objects, A_o, and the area of each cell nucleus, A_c, in pixels. Then the relative area of the objects are given as

  $$A_r = \frac{A_o}{A_c} \tag{5.5}$$

- Compactness:

  Compactness, γ, is defined as

  $$\gamma = \frac{P^2}{4\pi A} \tag{5.6}$$

  where P is the perimeter and A is the area. The perimeter is measured as the sum of distances between boundary pixels, where 1 is the vertical or horizontal distance between adjacent pixels and $\sqrt{2}$ is the distance between pixels on diagonals. With this formula the most compact shape is the circle, with $\gamma = 1$. Higher values of γ could indicate both very elongated simple shapes and a complex shaped object.

  There exist some alternative formulations around the same definition. Gonzalez and Woods [5, p.222] uses P^2/A as the measure of compactness and defines a circularity ratio given as $4\pi A/P^2$, which is just the inverse of γ.

- Major and minor axis:

  Is defined as the length of the axes of the ellipse that has the same second order central moments as the object. This is the ellipse that fits best to the object-region, see figure 5.1. The semi-major and semi-minor axis are given as

$$(\hat{r}, \hat{q}) = \sqrt{\frac{2[\mu_{20} + \mu_{02} \pm \sqrt{(\mu_{20} + \mu_{02})^2 + 4\mu_{11}^2}]}{\mu_{00}}} \tag{5.7}$$

  and of course the major and minor axis are given as twice the semi-major and semi-minor length.

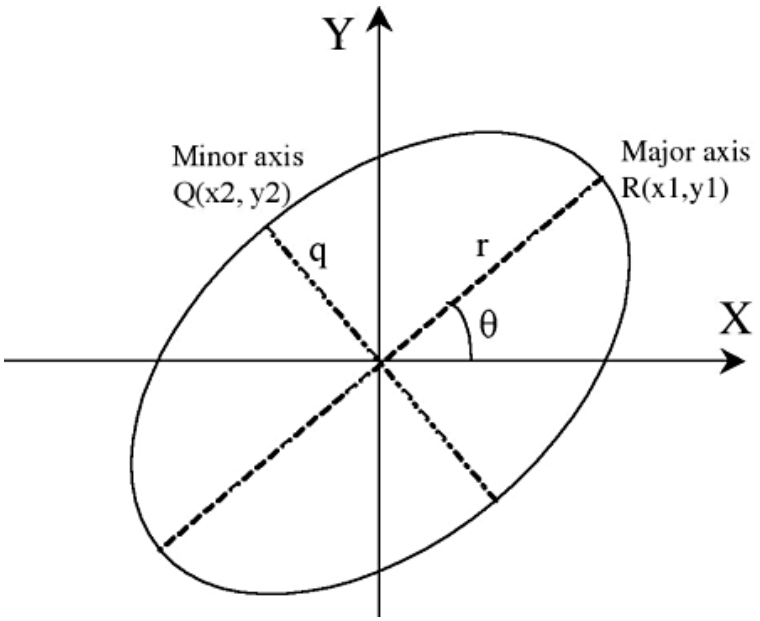


Figure 5.1: *Ellipse*

- Eccentricity:

  Eccentricity is a measure that tells something about how circular the shape of an object is. The eccentricity value is between 0 and 1, where an ellipse

whose eccentricity is 0 is a circle, while an ellipse with eccentricity equal to 1 is a line. From the last result, we also get the numerical eccentricity, given by

$$\hat{\epsilon} = \sqrt{\frac{\hat{r}^2 - \hat{q}^2}{\hat{r}^2}} \tag{5.8}$$

- Orientation:

  As with the last features, orientation is also derived by fitting an ellipse to the object region, the orientation is given as the angle between the major axis of the ellipse and the X-axis. Using figure 5.1 as reference, we assume as before that we have a 2-dimensional object $f(x, y)$. We also assume that the orientation of the object is unique, which means that there exist a rotated coordinate system (r,q), such that if we compute the second order central moment of the object around the r-axis, this will be the smallest possible second order central moment for this object. To find the orientation, θ, of this r-axis relative to the X-axis, we have to minimize the second order central moment of the object around the r-axis:

$$I(\theta) = \sum_r \sum_q q^2 f(r, q) \tag{5.9}$$

  where the rotated coordinates are given as

$$r = x\cos\theta + y\sin\theta \ , \ q = -x\sin\theta + y\cos\theta \tag{5.10}$$

  The second order central moment of the object around the r-axis, expressed in terms of x, y and the orientation angle θ of the object :

$$I(\theta) = \sum_x \sum_y [y\cos\theta - x\sin\theta]^2 f(x, y) \tag{5.11}$$

  We want to find the minimum of this moment, and therefor we will find the derivative of this function, putting this equal to zero and then solve the equation. We then have

$$\frac{\partial}{\partial\theta} I(\theta) = \sum_x \sum_y 2[y\cos\theta - x\sin\theta][-y\sin\theta - x\cos\theta] f(x, y) = 0 \tag{5.12}$$

$$\Rightarrow \sum_x \sum_y 2[xy(\cos^2\theta - sin^2\theta)] f(x, y) = \sum_x \sum_y 2[x^2 - y^2]\sin\theta\cos\theta f(x, y) \tag{5.13}$$

$$\Rightarrow 2\mu_{11}(\cos^2\theta - \sin^2\theta) = 2(\mu_{20} - \mu_{02})\sin\theta\cos\theta \tag{5.14}$$

$$\Rightarrow \frac{2\mu_{11}}{(\mu_{20} - \mu 02)} = \frac{2\sin\theta\cos\theta}{(\cos^2\theta - sin^2\theta)} = \frac{2\tan\theta}{1 - \tan^2\theta} = \tan(2\theta) \tag{5.15}$$

Which leads to the orientation angle given by the second order central moments as:

$$\theta = \frac{1}{2}\tan^{-1}\left(\frac{2\mu_{11}}{(\mu_{20} - \mu_{02})}\right) \tag{5.16}$$

Given the position of the object relative to the center of the nucleus, we can find the orientation relative to the radial direction. In figure 5.2 an example of a circular cell nucleus with one object inside is shown. With the center of mass for both the nucleus and object given we can easily calculate the angle ϕ by Pythagoras. Then with the angle θ already known from above, the angle, α, of the object relative to the radius through the center of mass of the object is given as

$$\alpha = \phi - \theta \tag{5.17}$$

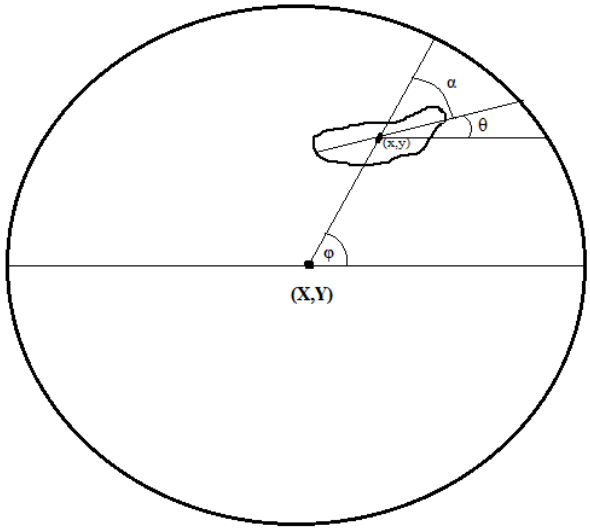


Figure 5.2: *Shows how the orientation relative the radial direction are given.*

5.1.2 Representation of the features

Thus far we have patients, which has a number of images, which has a number of objects, which has a number of descriptors. A possibility could be to classify each of the objects inside each cell nuclei, then by a majority vote scheme, classify each of the images. But, as mentioned before, the object structures are small with few pixels, so it is not very likely that this would work very well. We rather choose to represent the features of the objects inside the cell nucleus as distributions and then use a characteristic about these distributions as the feature for each cell nuclei image, and it is natural to use the mean as a starting point and maybe look at other characteristics, such as the variance and quantiles afterwards. With the features extracted from the objects it's time to move on to the classification part.

Classification

The purpose of classification is to assign an object or a pattern into a class. This is the same principle as segmentation, where a pixel is classified to one of the classes zero or one in binary segmentation. And as in segmentation there is almost never any perfect classifier which will classify every object correctly. So again our goal is to minimize the error we make.

Classification can be divided into two main areas, namely supervised and unsupervised classification [9]. In supervised classification, which we will focus on here, each object is assigned to one of a set of known classes, while in unsupervised classification the number of classes is unknown and the feature space is divided into a set of m clusters, where m must be estimated, and a pattern is classified to one of the classes based on similarity.

Jain et al. [9] list the four main approaches to classification:
1) template matching, 2) statistical classification, 3) syntactic or structural matching and 4) neural networks. In this thesis we will only go through the statistical approach.

In statistical classification a given pattern is assigned to one of n known classes $\omega_1, ..., \omega_n$. The decision made is based on the information of the $d-$dimensional feature vector $\mathbf{x} = (x_1, ..., x_d)$. Statistical classification can be divided in two; parametric classification and non-parametric classification.

5.2 Parametric classification

In parametric classification the observed data is assumed to be similar to some known distribution, e.g. the Gaussian distribution, and the parameters needed to specify this distribution are estimated from the feature data, e.g. the mean μ_i and Σ_i in the normal distribution.

5.2.1 Classification based on Bayesian theory

Bayes classification rule classifies an object to the class with highest posteriori probability $P(\omega_i|\mathbf{x})$. From statistical theory this well known probability is given as

$$P(\omega_i|\mathbf{x}) = \frac{P(\mathbf{x}|\omega_i)P(\omega_i)}{p(\mathbf{x})} \tag{5.18}$$

where $p(\mathbf{x})$ is the pdf of $\mathbf{x}$, which can be written as

$$p(\mathbf{x}) = \sum_{i=1}^{n} p(\mathbf{x}|\omega_i)P(\omega_i) \tag{5.19}$$

In other words, we classify the pattern $\mathbf{x}$ to the class which satisfies the rule

$$P(\mathbf{x}|\omega_i) > P(\mathbf{x}|\omega_j), \forall j \neq i \tag{5.20}$$

Given the n classes, we can assume that the apriori class probabilities $P(\omega_i)$ are known, which usually is estimated as the class frequencies. We then need to find the class-conditional probability functions $p(x|\omega_i)$, which describe the distribution of the feature vectors in each of the classes, also known as the likelihood function.

5.2.2 Discriminant functions

Instead of working directly with probabilities it is normal, and often useful, to represent a classifiers as discriminant functions $g_i(\mathbf{x}), i = 1, ...n$. The definition of this function is given as $g_i(\mathbf{x}) \equiv f(P(\omega_i|\mathbf{x}))$, where $f(\cdot)$ is a monotonic function [21]. We then classify $\mathbf{x}$ to class ω_i if

$$g_i(\mathbf{x}) > g_j(\mathbf{x}) \forall j \neq i \tag{5.21}$$

The decision surfaces, which separate the regions, are defined as [21]

$$g(\mathbf{x}) \equiv g_i(\mathbf{x}) - g_j(\mathbf{x}) = 0 \ , \ i, j = 1, ..., n \ , \ i \neq j \tag{5.22}$$

This means, for the Bayesian case, that the discriminant function is just the posteriori probability, and we get

$$g_i(\mathbf{x}) = P(\omega_i|\mathbf{x}) = \frac{p(\omega_i|\mathbf{x})P(\omega_i)}{p(\mathbf{x})} \tag{5.23}$$

This could be simplified, because (5.23) is proportional to

$$g_i(\mathbf{x}) = p(\omega_i|\mathbf{x})P(\omega_i) \tag{5.24}$$

which again is proportional to

$$g_i(\mathbf{x}) = \ln p(\omega_i|\mathbf{x}) + \ln P(\omega_i) \tag{5.25}$$

5.2.3 Gaussian distribution

The Gaussian distribution is one of the, or probably the most commonly used pdf in practice. Mainly because of it's well known properties and it's assymptotical nature, which means that if we have a large number of data, the Gaussian distribution will fit the data well. Given the d-dimensional feature vector $\mathbf{x} = x_1, ..., x_d$, the multivariate Gaussian density is defined as

$$p(\mathbf{x}|\omega_i) = \frac{1}{(2\pi)^{1/2}|\Sigma_i|^{1/2}} \exp\{-\frac{1}{2}(\mathbf{x} - \mu_i)^T\Sigma_i^{-1}(\mathbf{x} - \mu_i)\} \tag{5.26}$$

where μ_i and Σ_i is the maximum likelihood estimate. We now have all the ingredients to compute the aposteriori probability and i.e., the discriminant functions. The discriminant function is then given as

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \mu_i)^T\Sigma_i^{-1}(\mathbf{x} - \mu_i) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma_i| + \ln P(\omega_i) \tag{5.27}$$

Three cases of covariance structure

The covariance structure in the Gaussian density can be specified in one of three ways, this will change the complexity of the model, the decision surface and for sure the computation time. The three cases are:

1. $\mathbf{\Sigma_i = \sigma^2 I}$:
   In the first case we have the same variance for all classes and in fact the features are also independent, i.e., the correlation between the features is zero and the covariance matrix is just a matrix with a diagonal with equal elements, σ^2, and with the rest of the elements zero. We then get the discriminant functions as

$$g_i(x) = -\frac{1}{2\sigma^2}(\mathbf{x} - \mu_i)^T(\mathbf{x} - \mu_i) - \frac{1}{2}\ln|\sigma^2 I| + \ln P(\omega_i) \tag{5.28}$$

   The samples fall in equal-size hyperspherical cluster, centered about the mean. In figure 5.3 the prior probabilities are equal for the two classes, and the decision boundary is a linear function which is orthogonal to the line between the means and crosses this line exactly at the midpoint of the means. If the priors hadn't been the same the decision boundary would have been closer to the mean with lowest probability.

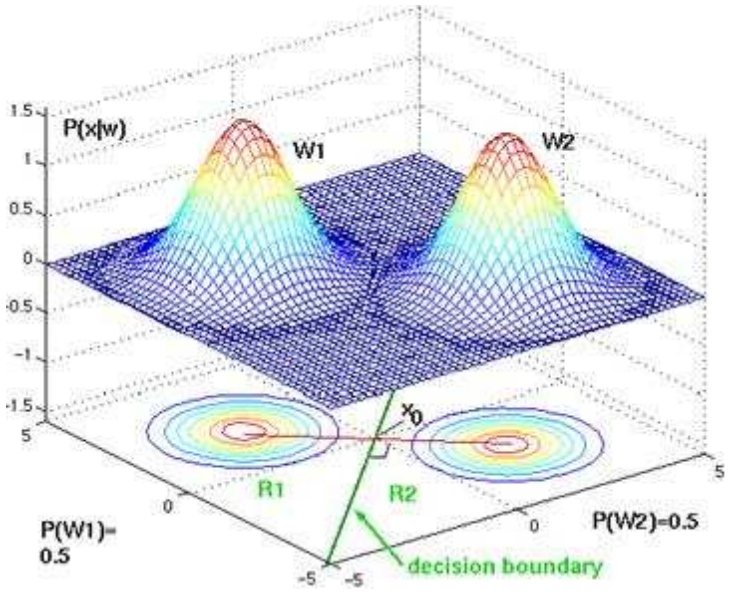


Figure 5.3: *Example with* $\Sigma_i = \sigma^2\mathbf{I}$

2. $\mathbf{\Sigma_i = \Sigma}$:
   In this case we will have a common covariance structure for the different classes, which will result in hyperellipsoidal clusters with the same shape and size. Now we will assume that there is correlation between the features and they are no longer independent. The discriminant function will be

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \mu_i)^T \Sigma^{-1}(\mathbf{x} - \mu_i) - \frac{1}{2}\ln|\Sigma| + \ln P(\omega_i) \tag{5.29}$$

   Again we will have a linear classifier, which is shown in figure 5.4. The two classes have the same prior probabilities and the decision rule is in principle the same as for the case above.

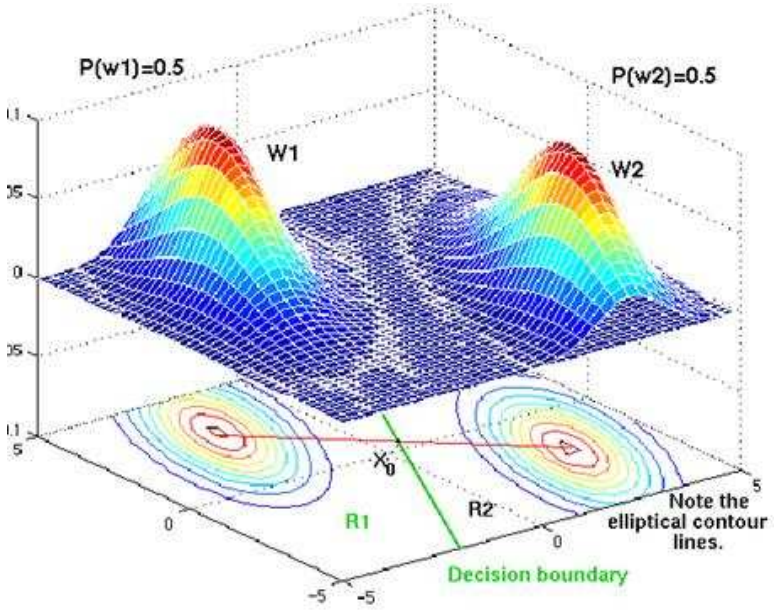


Figure 5.4: *Example with* $\Sigma_i = \Sigma$.

3. $\mathbf{\Sigma_i}$:
   In the general multivariate Gaussian model each class has it's own covariance matrix, and the discriminant functions will be quadratic. This means that we now have a complex decision boundary, but even if such a function could do a better separation of the objects, there still are some issues to consider. How many coefficients should be used to estimate the decision boundary? Do we want a classifier that complex? The discriminant function in this case, will be

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \mu_i)^T \Sigma_i^{-1}(\mathbf{x} - \mu_i) - \frac{1}{2}\ln|\Sigma_i| + \ln P(\omega_i) \tag{5.30}$$

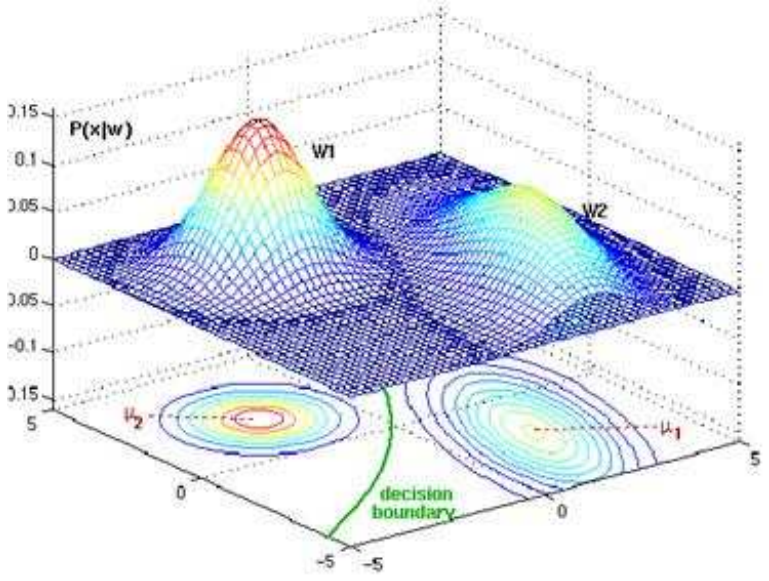


Figure 5.5: *The case with arbitrary covariance structure.*

It might be that the features computed from the different prognosis of cancer in our data have different variance, we will still assume independent features with the same variance.

5.3 Non-parametric classification

In the last section parameters from a known function where estimated and the same classes where assumed to have the shape and properties of the chosen function. But this might not be the best way to describe the classes. Why not let the data itself describe the model? Such models are non-parametric models, and we will go through one of these here.

5.3.1 K-nearest neighbor rule

The k-nearest neighbor rule is a very simple rule to understand, even though it can be trained to complex decision boundaries. As expected this rule assigns $\mathbf{x}$ to the class which the majority of the k-nearest neighbors belongs to [21]. This means that the k neighbors with the smallest distance to the $\mathbf{x}$ is examined, and the class that is most frequently represented amongst the k, will determine the class label of $\mathbf{x}$. There are several ways to measure the distance to neighbors, but usually the Euclidian distance is used.

Of course the question is: How many neighbors should be included in the vote? As before the usual way to decide this is to minimize the error in what we are doing, i.e., k is chosen such that the classification error is as small as possible. But as the number of neighbors included increases, the more probable it is to classify the pattern to the class with highest posteriori probability, and if all neighbors are considered this is obviously the case. For $k = 1$, this method is known as the nearest neighbor rule, and k-nearest neighbor rule is just a generalization of this rule. Crossvalidation is often used to estimate the optimal k, which means this method will be time consuming if there are a lot of features and samples.

5.4 Feature selection and dimension reduction

When features are extracted one often wants to reduce the dimensionality of the feature vector. There are two reasons to keep the dimension as small as possible [9]: measurement cost and classification accuracy. But there is a trade off, a too large feature vector could cause the curse of dimensionality [9] and a too small feature set could lead to loss in discrimination. There are mainly two ways to reduce the dimensionality of the feature space, namely feature selection and dimension reduction.

5.4.1 Feature dimension reduction

Feature dimension reduction reduces the d-dimensional feature space to a lower dimensional subspace in a linear or non-linear fashion. The popular choices of

linear transforms are Fisher's linear discriminant analysis (FLDA) and principle component analysis (PCA).

Principle component analysis

In PCA the $m < d$ largest eigenvectors of the dxd covariance matrix are computed and a linear transform is given as [9]

$$\mathbf{Y} = \mathbf{XH}, \tag{5.31}$$

where $\mathbf{Y}$ is the nxm matrix if the linear transformed data, $\mathbf{X}$ is the original nxd matrix of observed data and $\mathbf{H}$ is the dxm matrix of the m largest eigenvectors of the covariance matrix. The goal of such a transformation is to find an orientation where the projected data are well separated [14].

Fisher's linear discriminant analysis

As mentioned in the segmentation chapter, the similarities of FDLA and Otsu's segmentation algorithm [16] is that both methods maximizes the between class variance, while minimizing the within class variance. We may write the features as a linear combination

$$\mathbf{Y} = \mathbf{w}^T\mathbf{X} = w_1x_1 + \cdots + w_dx_d \tag{5.32}$$

where the weights is found by maximizing the function

$$J(\mathbf{w}) = \frac{|\tilde{m}_1 - \tilde{m}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2} \tag{5.33}$$

where $\tilde{m}_i$ and $\tilde{s}_i^2$ are the mean and variance for the projected points, in the two classes case.

5.4.2 Feature selection

Feature selection is a totally different approach to the dimensionality reduction process. Most of the methods used are basically testing out different combinations of the features and choosing the combination that perform best. The problem about this approach is that the number of different ways to select a subset out of the original d-dimensional vector, becomes enormous even with a moderate set of features. Actually an exhaustive search through the N features that is the optimal subset of d original feature set, we would have to test $\sum_{i=1}^{N} d!/[i!(d-i)!]$ combinations.

A suboptimal scheme is to select the best single features based on some quality criteria, such as estimated correct classification rate. However, a combination

of the best single features will often imply correlated features and will therefor be suboptimal [2], [9]. Sequential forward and backward selection [9], [24],implies that when a feature is selected or removed, this decision is final, resulting in a nesting problem [9]. Stepwise forward and backward selection [9] overcomes the nesting problem. This is a generalization of the "plus-l take away-r" method [9], which in turn has been improved into floating search [17] by making a number of forward and backward search steps data dependent. In almost any feature selection problem, these methods perform better than straight sequential search [9]. However, the number of feature set evaluations will certainly increase.

5.5 Test and training sets

Before choosing a classifier, the data has to be divided into a training set and test set. We want to train our classifier such that it could handle the classification of future patterns. Therefore we want a classifier that is not a perfect fit to the training data, because we want a classifier that can handle variations from the training data.

There are four main approaches to test the performance of a classifier [18]

1. Resubstitution:
   All data is used to train the classifier and then the same data set is used to evaluate the performance of the classifier.

2. Holdout method:
   This method partitions the data into two. Then one set is used for training and the other for testing and error estimating.

3. Cross-validation:
   The data set is divided into K equal subsets. Then the classifier is trained on the (K-1) subsets, and tested on the last subset. This routine is done K times, such that every subset is used for validation, and the average classification error is reported. A special case of this method is to partition the data into the total number of observations, which means that every observation except one is used for training and then tested on the single observation left out. This type of cross-validation is referred to as leave-one-out cross-validation.

4. Bootstrapping:
   There exist several bootstrap procedures that could be used. The essence of bootstrapping is to sample, with or without replacement, a new set of data equal to the size of the original data. The sampled set and the original set can then be used in several ways for training and testing. The procedure is

repeated r times, where r typically is in the range from 200-1000, and the mean error is reported.

Obviously there are several ways to divide the data into a training and test set, and this is certainly not trivial. The problem is that if the training set is too small the classification error will be large, while if the test set is too small the variance will be large. The optimal way to partitioning the data will depend on several factors, such as classifier used and dimensionality among others [18]. The ovarian cancer data consist of a relatively small number of patients(134), and is biased in the way that only 40 of the cases are with bad prognosis. We will try out different ways of partitioning the data. The obvious way to split the data is by dividing it 50/50, i.e., the data is divided into two, where the balance between good and bad prognosis is kept equal in both training and test set. Raudys and Jain [18] proposes some formulas for such a partitioning, where it turns out that 50/50 splitting is optimal if the dimensionality of the feature vector times the Mahalanobis distance between the classes is approximately equal to 30. But the distance between the classes is unknown and has to be estimated. We would also like to try out a balanced training set, which will result in fewer observations for training, and more data for validation.

5.6 Classifying the patients

The main goal of our experiment is of course to classify the patients correctly, but in the analyses of the images, each image is classified into one of two prognostic classes. This will in almost every case result in patients that will have images classified into both classes. The easiest and certainly the most intuitive way to solve this is just to take a vote and classify the patient to the class which has the majority of the votes.

Chapter 6

Results and discussion

This chapter presents the results and discussion of the experiments we have done. The results and discussion will be well separated, but is in the same chapter to ease the reading.

We only have one data set in this project, but we will use a carefully selected subset in the beginning to show the correlation between earlier work done and the methods we have used. Then the entire data set and a subset based on DNA-ploidy are examined for balanced and unbalanced training sets. Finally an experiment where the mean of all feature values for each patients are calculated and the patients are directly classified.

Based on the features that we have extracted it seems natural that we divide each of the experiments into three, i.e., one experiment with just the object features, one with the cell nuclei features and one which combine all features.

6.1 A balanced subset of 20 patients

From earlier work which have used the same data material used in this project, we have certain texture measurements that we can use to select a subset of the patients that we know is easier to separate than other patients. With this in mind we have selected a subset of 20 patients with texture measurements that are well separated for the two classes, i.e., the patients where carefully selected w.r.t. the average texture value of the cell nuclei images for each patient. Table 6.2 lists the patients selected, with prognosis, and correct classification rate (CCR) for each of the three classification schemes and the texture measurement. And as we can see, the patients with low texture values are patients with good prognosis and vice versa. Note that this is a balanced subset!

6.1.1 Classification using the object features

In this first experiment we will only use the object features to classify the images. To choose a classifier 5-folds cross-validation were used with all patients and the k-nearest neighbor, the Bayesian classification rule with linear and quadratic decision boundary were tested. In table 6.1 the result of the cross validation is shown, with the average classification error for each classifier in the second row. As we can see the linear classifier is the one doing best and this classifier is therefor chosen.

The images were then classified by a bayes normal classifier with leave-one-out cross-validation, i.e., the classifier was trained with 19 patients and then the one patient not used in training was classified. This is rotated such that all patients are used as a test set. All object features were use in the process, such that the feature vector is of dimension 10, 5 for the bright objects and 5 for the dark objects.

knnc	ldc	qdc
38.6	38.4	45.7

Table 6.1: *Classification error for 3 classifiers.*

Results

The results of classifying each image are shown in table 6.2, column 3, where the CCR for each of the patients is given. Using the classification rule established in the last chapter, i.e., classify the patients to the class which is most probable, will result in 6 patients being wrongly classified, leading to a correct classification rate for the 20 patients of 70%. This is visualized in figure 6.1, where a scatterplot of the classification for the 20 patients is shown. The circles indicate a patient with good prognosis, while the pluses are patients with bad prognosis. The line marks the 50% error /correct classification rate. The axis in the figure are the number of images, for each patient, that are classified into the two classes, with bad prognosis on the y-axis and good prognosis on the x-axis.

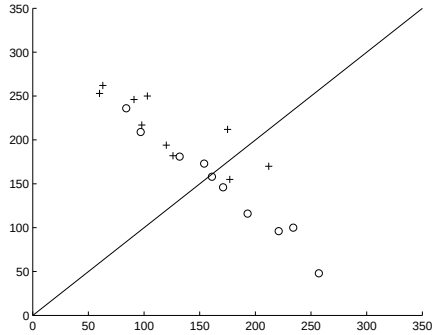


Figure 6.1: *Scatterplot for the classification of 20 patients, using only the object features (circle=good prognosis, +=bad prognosis).*

Patient id	Prognosis	CCR(object)	CCR(cell)	CCR(all)	Texture
L23-003	G	50.5%	79.9%	80.1%	0.8327
L23-006	G	26.3%	68.1%	67.2%	1.1756
L23-012	G	62.5%	90.6%	90.3%	0.5898
L23-040	G	84.3%	84.6%	83.6%	1.1685
L23-050	G	69.7%	78.9%	79.8%	0.9824
L23-074	G	47.1%	87.8%	87.2%	0.2102
L23-088	G	53.9%	89.3%	89.3%	0.5201
L23-099	G	42.2%	89.8%	90.1%	0.2878
L23-109	G	70.1%	78.1%	75.2%	0.6069
L23-115	G	31.7&	89.5%	89.5%	0.2092
L23-011	B	68.9%	55.2%	57.1%	5.3849
L23-020	B	46.7%	37.7%	37.1%	3.5704
L23-027	B	59.1%	94.5%	93.5%	6.9114
L23-051	B	54.8%	70.5%	71.6%	6.3881
L23-101	B	73.0%	82.8%	82.2%	7.9478
L23-110	B	70.8%	88.7%	87.2%	5.9707
L23-140	B	44.5%	59.7%	59.4%	4.7985
L23-169	B	80.6%	93.2%	93.9%	8.0907
L23-213	B	80.8%	90.7%	89.8%	5.5114
L23-370	B	61.8%	86.0%	86.3%	8.2697

Table 6.2: *Table of correct classification rate for the subset of 20 patients.*

6.1.2 Classification using the cell nuclei features

In the second approach we will only use the cell nuclei features to classify the images. The same procedure as in the last section, for both evaluation of classifiers and classification of the images is used. In this experiment the k-nearest neighbor classifier is chosen, as shown in table 6.3.

knnc	ldc	qdc
17.6	21.1	23.9

Table 6.3: *Classification error for 3 classifiers.*

Results

The results of classifying the images, based on the cell nuclei features, are shown in column 4 of table 6.2. Using the same patient classification scheme as earlier, we would only mis-classify one the patients, namely patient number 20. This will give a CCR equal to 95%. A scatterplot of the result is given in figure 6.2, which is the same type of figure as in the last section.

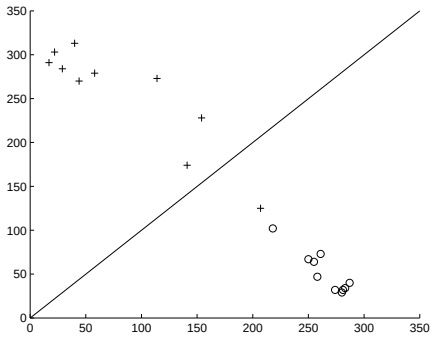


Figure 6.2: *Scatterplot of the classification of 20 patients, circle = good prognosis and += bad prognosis, based on cell nuclei features.*

6.1.3 Classification using all the features

In the last experiment with this subset, we will use all of the features and use the same procedures as before. Again the k-nearest neighbor classifier is the one that does best in the 5 fold-cross-validation, and the mean classification error with this classifier is 17.6%.

Results

The classification results are shown in column 5 of table 6.2. As with the cell feature experiment, the only patient which will be wrongly classified is patient number 20, such that the correct classification rate will be 95%. Figure 6.3 shows a scatterplot of this trial.

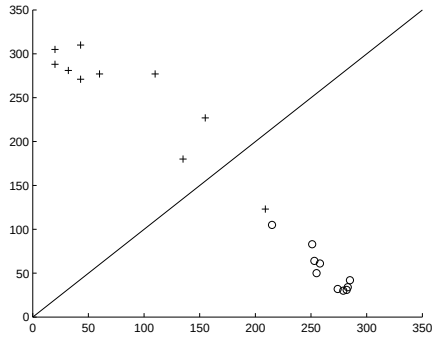


Figure 6.3: *Scatterplot of the classification of 20 patients, using all features, where circle=good prognosis and +=bad prognosis).*

6.1.4 Discussion

With such a small set of observations there will be limitations on how we are going to train and test the data. There really are only two option, i.e., resubstitution or leave-one-out cross-validation. We have chosen to use the latter, because this will keep some sort of independence between training and validation, while resubstitution will train on all the data and then test on the same set. There are problems with both of these methods, the resubstitution often gives too optimistic results and the error estimate of the cross-validation method will have high variance [14]

The real difference in the classification results shown in column 3, 4 and 5 in table 6.2, are between the results by just using the object features and the two other experiments. There is little difference between column 4 and 5 in the table, i.e., it seems that the information separating the classes lies in the cell nuclei features. And when we use all of the features there is little to gain instead of just using the cell nuclei features. One should note that there might be more to gain if an optimal, or suboptimal, selection of the features where used instead. The classification result in the two last trials, with a CCR=95% for the patients, are certainly in the region where we want to be in the end. But with only 20 patients in the data set there obviously will be a lot of uncertainty in the results, and this should only be regarded as a demonstration of concept.

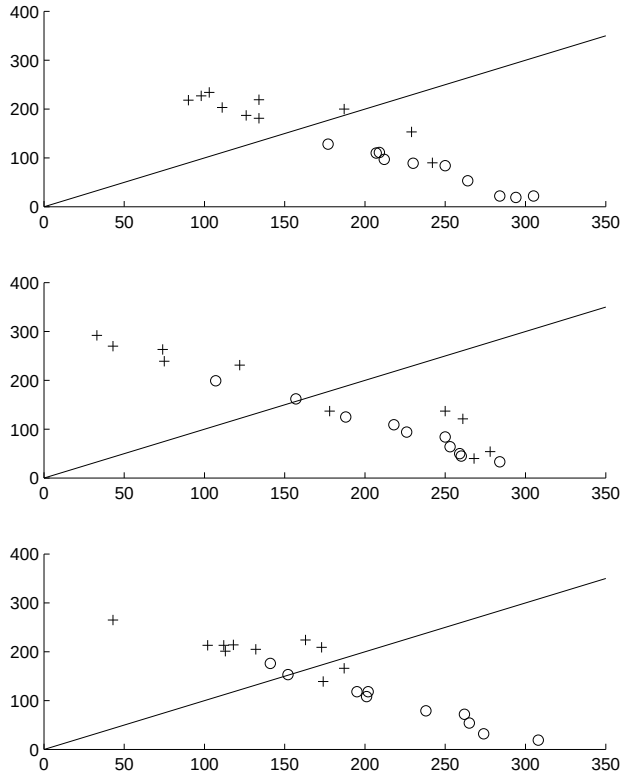


Figure 6.4: *Result of classification by the three different groups of cell features.*

6.1.5 A closer look at the cell features

The results from the analysis of the subset gives rise to some questions, especially the cell features should be more closely investigated. The first question is: which of the cell features is it that holds the information that separates the classes? There are several ways to investigate this, and we start by dividing the cell features into three groups; number of objects, radiometric features and geometric features, and then classify the subset of 20 patients in the same manner as before. The results of the three classification schemes are plotted in figure 6.4. The subfigure on top shows the classification by using only the number of bright and dark objects as feature, and as we can see this gives a decent result with a CCR= 90%, which implies that the number of objects certainly are important features.

The plot in the middle of figure 6.4 shows the classification results using the gray level intensity features, namely the mean and variance of the gray level. This gives a CCR=70%, and it might be that these features do not contribute much

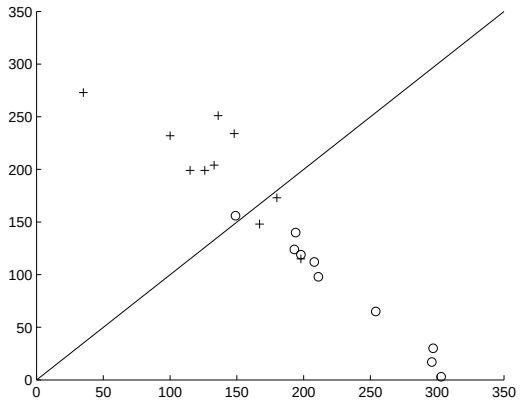


Figure 6.5: *Classification result with only cell nucleus area as feature.*

in the overall results.
Finally, in the bottom row, the result of using the geometric features, i.e., the area, the compactness and the eccentricity of the cell, is shown. Now three of the patients are mis classified, which gives a CCR=80%.

These results are a bit unexpected, a least for the geometric and radiometric groups, where one would expect that the gray level intensities would be more important than the geometric features. Especially since the geometric group includes the features compactness and eccentricity which could be influenced by the cell preparation process. To check this a classification with only the cell nucleus area as feature is done, and as we can see in figure 6.5, this feature alone does the same as when combined with the other geometric features, with a CCR= 80%. Another observation is that the features used in the first row of figure 6.4, the number of dark and bright objects, could arguably be in the object features. By doing so and then classifying the subset, the CCR goes from 70% to 90%, which shows how important these features are. This is shown in figure 6.6.

We could also explore the features in another way, to determine which features that separate the two classes, e.g., by calculating the Mahalanobis distance between the distribution of feature values for the two classes. The Mahalanobis distance is defined as

$$J(\omega) = \sqrt{\frac{2(\mu_{\omega_1} - \mu_{\omega_2})^2}{\sigma^2_{\omega_1} + \sigma^2_{\omega_2}}} \tag{6.1}$$

where ω_i for $i = 1, 2$, indicates the two classes; good and bad prognosis. The results for the 7 cell features are shown in table 6.4. And these results just confirm which features that could separate the two classes well. All features seem to be separable, for the two classes, except compactness and maybe eccentricity.

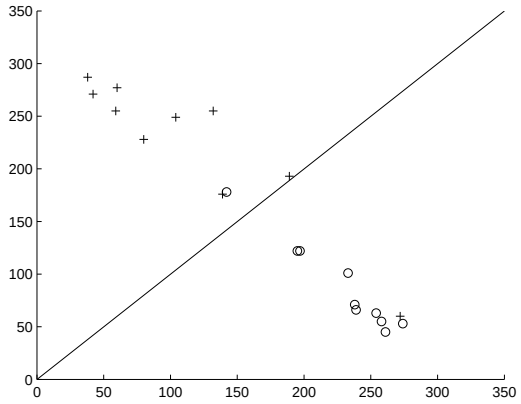


Figure 6.6: *Classification of the subset using the object features including the number of bright and dark objects.*

Feature	J(ω)
area of cell nucleus	0.6642
number of dark objects	0.6097
number of bright objects	0.5402
gray level variance	0.5031
gray level mean	0.4725
eccentricity	0.2563
compactness	0.1262

Table 6.4: *The Mahalanobis distance between the distribution of feature values for the two classes.*

6.2 Simulations with balanced training sets

As pointed out earlier, the data set contains only 40 cases of bad prognosis, while there are data from 94 patients with good prognosis, which in total is not much data from a statistical point of view. In this section the data set has been divided into sets where the training data consist of 40 patients, 20 of each prognosis, and the rest of the data, 94 patients, are used for validation, i.e., a balanced training set.

To estimate the classification error for the patients, we have used a type of bootstrapping method, where the 40 patients used for training are picked out randomly for each iteration and the remaining 94 patients are used for testing. Raudys and Jain [18] recommends using 200-400 simulations when bootstrapping is performed. To assure that every patient is evaluated at least 200 times, we need to use about 500 iterations. The patients with bad prognosis will be in the training and test set half of the times and will, if the random picking is perfect, be validated 250 times. The patients with good prognosis on the other hand, will be in the test set about 400 times during the 500 iterations, since only about 1/5 of the data from this group will be randomly chosen to be in the training set for each iteration.

The mean classification error of the images, for each patient, are reported. Asymptotically this will lead to a more correct error estimation, because of the fact that there are only a small number of patients in the data set. If we had just divided the data in two, and then trained and validated once, there certainly would be a lot of uncertainty in such an error estimate, and by chance we could get either a good or a bad result. There are also some issues with this iterative approach, since we will use patients for both training and testing in some way, but of course not in the same iteration. Then a decision for which class the patients belong to is made by the rule established earlier. i.e., the patients are classified to the class which is most probable on the average, which is based on the majority of the image classification.

Figure 6.7 shows the plot of how the mean error develops as the number of observations increases for 5 randomly chosen patients. It is obvious that 500 iterations is enough, the mean errors converge pretty early, and after 100 error estimations the mean errors are almost stable, and surly 200 observations would be enough. Note the difference in number of observations, which indicates the prognosis of the patients.

Figure 6.8 visualize how the uncertainty in the error estimate decreases and becomes stable as the number of iterations increases, for one patient. The first row shows the plot of the mean error with a 95% confidence interval. Note how

the interval shrink in width, as the number of observations increases. This is better visualized in the plot of the variance alone, which is shown in the second row of figure 6.8, where the decrease is more apparent. These plots really shows the problem of a classification scheme where one divides the data into two sets and train and classify once. The error estimates of such a setting would be very unreliable, i.e., the variance will be large and the results will be randomly good or bad. Especially this will be a problem for patients where the mean error converge close 0.5. With only one evaluation of such a patient will result in a wide confidence interval, because of high variance, and the classification result will be determined by chance.

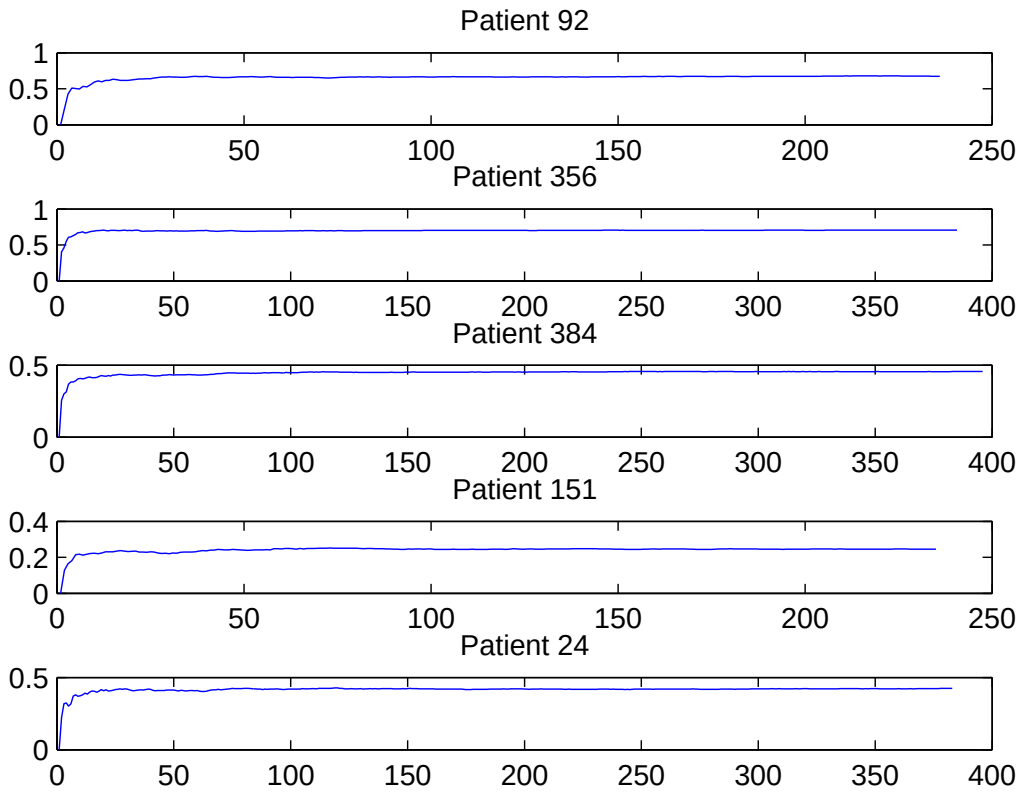


Figure 6.7: *Plot of mean error as number of observations increases.*

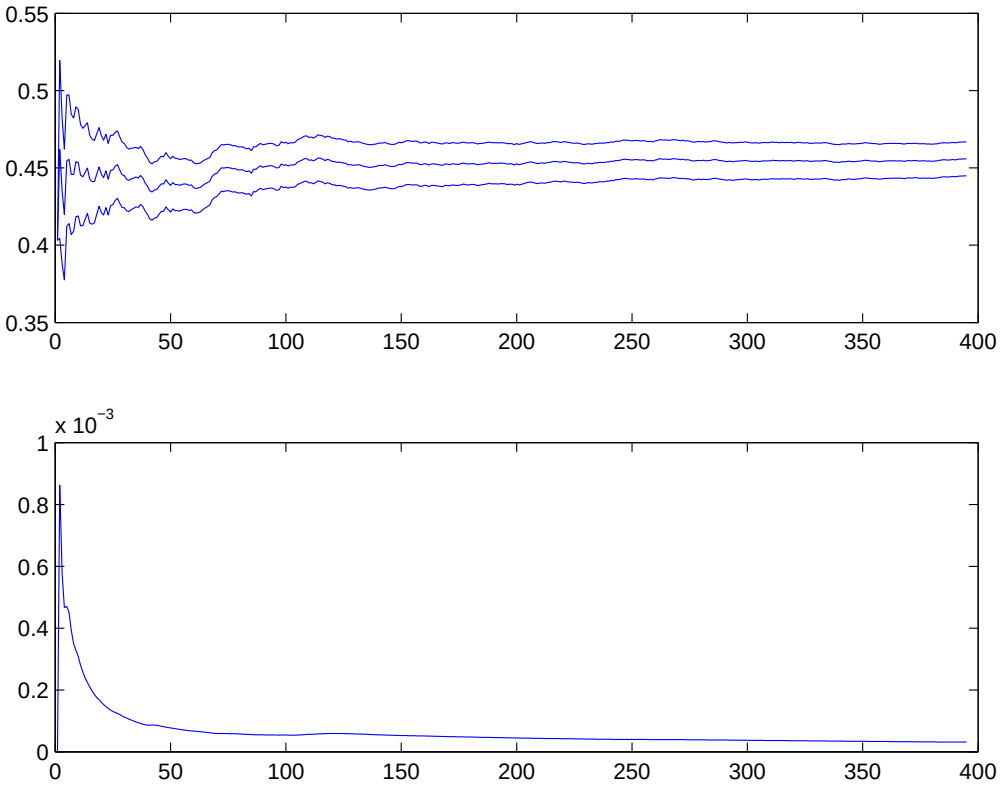


Figure 6.8: *Plot of mean error of patient 384 with a 95% confidence interval in the first row, and a plot of the variance for the mean error as the number of observations increases in the bottom row.*

A balanced subset based on ploidy

From earlier research on this material the experience is that patients with cell nuclei that are tetraploid or polyploid are often classified into the opposite class than the two other types of ploidy. The ploidity are easily determined by the histogram of the images. To check if ploidity is significant in our setting, an experiment without the patients which are classified as tetraploid and polyploid is done. 4 of the patients from bad prognosis and 28 of the patients from good prognosis are removed, and we now have a set with 102 patients, 66 with good prognosis and 34 with bad prognosis.
Following the same convention as in the rest of this section, we then have a training set of 36 patients, i.e., a balanced training set.

6.2.1 Results

As in the last section we will do three different experiments using balanced sets for training and we do this for two different data sets. One for the entire data set and one for the subset based on ploidy. In table 6.5 the results for the different feature combinations are shown for the experiment with all the patients and the experiment for the subset of 102 patients.

For the full data set we get a CCR=53% if we only use the object features. If the cell features are used the correct classification rate is 71.6%. The best classification results are found by using all features, which gives a CCR=72.4%.

In the lower part of table 6.5 the results for the subset of 102 patients are given. The improvement in classification results are significant, and gives a CCR=81.4% as the best result with the use of all features or only the cell features. Again the object features gives a poor classification result with a CCR=60.8%.

6.2.2 Discussion

As we saw with the subset of 20 patients, the cell features again has the most influence on the classification. Even though all features gives a slightly better result, it is obvious that the cell features are the ones that contributes. The classification rates in them self is not that good. Especially the object features gives a poor classification result with a CCR=53.0%, while the cell features and classification with all features gives almost the same with a CCR about 72%.

Removing the 32 patients with tetraploid or polyploid cell nuclei, certainly improves the classification result, and increases the correct classification rate for each of the feature groups with approximately 10%. Since ploidity is easy to establish from the images, this approach opens the possibility of a two-stage classification. Where the first step would be to divide the patients into groups based on ploidy, and then analyse the images from the two groups separatly.

At the end of last section we took a closer look at the cell nuclei features, and found that some of the features were more important than others. In table 6.6 the results of classifications with different subgroups of the cell nuclei features are shown. The results from the subset of 20 patients clearly indicated that the number of objects was essential in separating the two classes, but this is not that clear when all patients are considered.

	Prognosis	Patients	Correct classified	Mis-classified	CCR
All 134 patients:					
Object features:					
	Good	94	53	41	52.1%
	Bad	40	22	18	55.0%
Total:		134	71	63	53.0%
Cell features:					
	Good	94	70	24	74.5%
	bad	40	26	14	65.0%
Total:		134	96	38	71.6%
All features:					
	Good	94	71	23	75.5%
	Bad	40	26	14	65.0 %
Total:		134	97	37	72.4%
102 of 134 patients:					
Object features:					
	Good	66	38	28	57.1%
	Bad	36	24	12	66.7%
Total:		102	62	40	60.8%
Cell features:					
	Good	66	60	6	90.9%
	bad	36	23	13	63.9%
Total:		102	83	19	81.4 %
All features:					
	Good	66	59	7	89.4%
	bad	36	24	12	66.7%
Total:		102	83	19	81.4 %

Table 6.5: *Table with classification result for all patients and the subset of 102 patients, using different features.*

Features	CCR	Mis-classified(good)	Mis-classified(bad)
Number of objects	65.7%	23	23
Geometric features	70.9%	28	11
Radiometric features	64.9%	23	24
Object features + number of objects	68.7%	23	19
Cell features - number of objects	73.1%	22	14

Table 6.6: *Table with results of classification using different features.*

Actually the geometric features are the ones that do best, but if we include the number of objects into the object feature set there is a huge improvement in the classification result and the CCR jumps from 53% to 68.7%. Another interesting note, is that if we subtract the number of objects from the cell nuclei features, it actually improves the classification results, which shows the complexity of feature selection.

6.3 50-50 split of the data set

The next experiment is to split the data into two equal groups, i.e., train with half of the good prognosis and half of the bad prognosis, and then validate on the second half of the data. The training set will then be unbalanced, with 47 patients from good prognosis and 20 patients from bad prognosis. We will use the same approach as in the last section, i.e., choosing training and test sets randomly, do this for 500 iterations, report the mean error and make a decision for each patient based on the mean error.

Since the classification error will be in the area of the percentage of patients with bad prognosis, a classification without saying anything about the apriori probabilities for the two classes, leads to decision boundaries which mis-classify all patients with bad prognosis. The solution to this is to force the classifier to use equal apriori probabilities for the two classes.

The results however, are about the same as for training with a balanced set. Using the object features alone gives a CCR= 55.2%, which is slightly better than in last section, while using only the cell nuclei features gives the same results as with a balanced training set with a CCR=71.6%, and using all features gives a slightly poorer result with a CCR= 70.9.

And as in the last section the subset of patients based on ploidy are checked, and this gives almost the same results as with a balanced set. The CCR for both the cell features and all features are 81.4%, while the CCR for the object feature

Features	CCR	Mis-classified(good)	Mis-classified(bad)
texture	81.4%	8	11
Object	61.8%	26	13
Cell	76.5%	14	10
All	77.5%	13	10
Object+texture	87.3%	0	13
Cell + texture	89.2%	0	11
All + texture	89.2%	0	11

Table 6.7: *Table with results of classification using different structural features combined with a textural feature, for the subset of 102 patients.*

is 61.8%

6.4 A final experiment

Until now we have classified the images individually, based on the features extracted from each of them, and then the patients are classified based on the majority of the images. In this section, however, another approach is considered.

In section 6.1 we introduced a texture feature from a previous study. From this experiment we have got texture features from the 102 patients with cell nuclei that are not tetra- or polyploid. Since this texture measurement is given patients-wise, we will have to find the average feature values per patient, for the features we have extracted. As a final experiment we then can combine the average structural features we found with the textural feature, which will mean that we classify the patients directly.

In this analysis we will use the "plus-l take away-r" method [9] for feature selection, but the texture feature will always be in the evaluated feature set. The Bayesian classification rule is used, with a linear decision boundary [21].

6.4.1 Results

The classification results from this analysis are shown in table 6.7. If we classify the 102 patients and only use the texture feature this leads to a CCR=81.4%. However, if we don't use the texture feature the object features give a CCR=61.8%, while the cell features give a CCR=76.5%, and, at last, using all features will result in a correct classification rate of 77.5%.

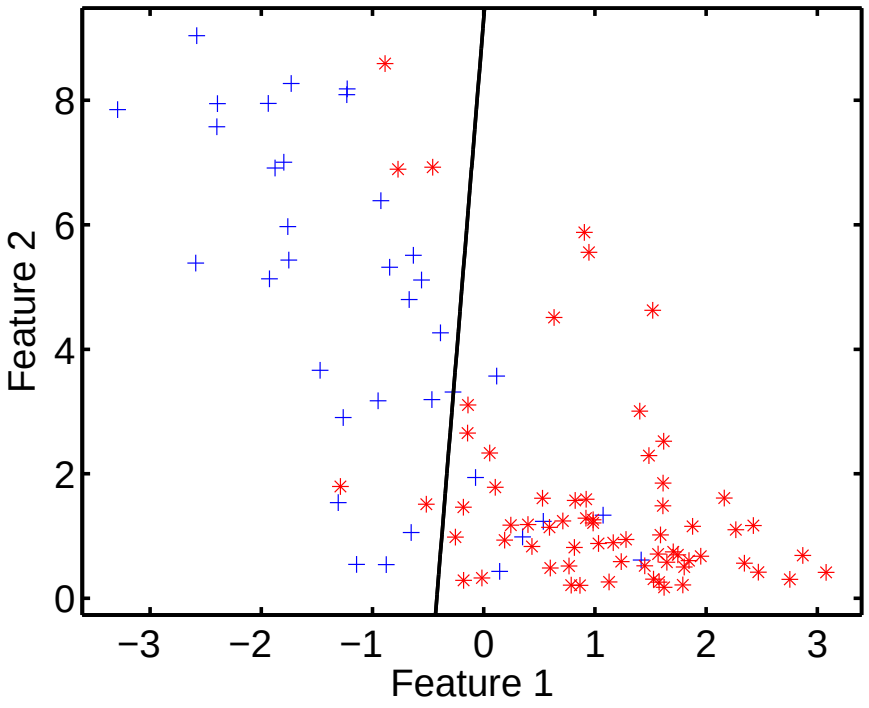


Figure 6.9: *Scatterplot of the texture feature (y-axis) against a FLDA of the structural features (x-axis), for the 102 patients (* = good prognosis and + = bad prognosis).*

The three different groups of features including the texture feature will do approximately the same, with a CCR=89.2% for both the cell nuclei+texture and all structural features + the textural feature, while the object features including the textural features do slightly poorer with a CCR=87.3%.

6.4.2 Discussion

These results only confirm what we saw in the previous sections, that only evaluating the 102 patients that are classified as diploid or aneuploid, improves the classification results. It is noticeable that for the three feature groups, without the texture feature, almost gives the same result if classify the patients directly with average feature values, as if we classify the patients using the scheme in section 6.2 and 6.3.

The texture feature alone results in a classification rate which is slightly better than the best structural features, but when the texture measurement is combined with the structural feature sets, this improves the results significantly. While the results of the texture feature alone, or the structural features alone gave a correct

classification rate around 80%, combining them results in a CCR close to 90%.

In figure 6.9 a scatterplot of the texture feature against a Fisher linear discriminant mapping of the structural features is shown. Even though the classes are overlapping in their projections onto each of the two axes, it shows that a linear decision line could separate the classes pretty well, with 12 patients mis-classified, this would give a CCR=88.2%. But remember that this is just an illustration in the 2-D space.

6.5 Discussion

In this section some general observations from the analysis performed are discussed. As pointed out several times in this thesis, for most of the experiments done, it is the images that are classified into one of the prognostic classes and then by a majority vote each patient is classified into one of the two classes. A classification rule minimizes the classification error, but this is performed with respect to all images in the different classes, and this might not be the optimal classification rule to give the best classification of the patients. An evenly spread error amongst all patients would certainly be the best, even if it means that the overall error is higher. So it could be that finding the average feature values as we did in section 6.4, and classify the patients directly is a more proper way of analysing the data.

Another issue not yet discussed, is the classification of patients from a medical point of view. For which group of patients would it be preferable to have the smallest classification error? This is not a trivial question to answer. One could argue that classifying patients with bad prognosis with lower error than in the opposite case, would lead to better care and follow up for this group. But this will mean that some patients with good prognosis are put through heavily treatment and the consequences such treatment leads too. We have not considered this when we have done our analysis.

When we started this project our hypothesis was that the structure inside the cell nuclei holds the information that could distinguish an images from a patient with good prognosis and a patient with bad prognosis. And as the project evolved it seemed reasonable, and easy, to include global information of each cell nuclei, such as area and mean gray-level.

From the results in this chapter this hypothesis is partially rejected, and the classification results which were based on the object features almost always gave a poor result. But one should notice that some of the best features, the number of objects, is a result of analysis of the structures within the cell nuclei and

would not been found without segmenting the images. And we have shown that if the number of objects are included in the object feature group this improves the classification results significantly.

The conclusion of these results is certainly that with such a complex data material, it is essential to use as much of the information and experience that is obtained from former studies. For this project, and perhaps in general, the problem that is being solved probably should be done in several steps. It is obvious that the first step in analysing the ovarian cancer material, is to partition the patients into two groups based on DNA-ploidy and then analyse the two groups separatly.

The results from the experiment with the 102 patients, that have been classified as either diploid or aneuploid, are much better than the analyses for all the patients.

Chapter 7

Summary and Conclusion

The main aim of this thesis has been to segment and separate structures inside cell nuclei images and develop and evaluate structural features with a potential prognostic value for early ovarian cancer. I this thesis, we wanted to take a different approach to the analysis of the images of the cell nuclei, than what has been done before. Instead of using features from statistical gray level texture analysis methods, we have aimed at using features that describe the structures inside the cell nuclei.

Our first problem was to find a segmentation algorithm that could handle the variations in the different images and especially the variations between the patients. A modification of Niblack's adaptive segmentation algorithm was developed. The refined thresholding method is spatially adaptive within each image and the parameters are adapted to the histogram of each image.

To separate the regions that were segmented, we had to be careful, because the low resolution in the images resulted in objects with few pixels. Using mathematical morphology, a rotated linear structure element was used iteratively to separate the objects.

Then a set of simple structural features was extracted from the objects, and the images were classified, usually with Bayes classification rule and a linear decision boundary. The problem with few observations in the data set, 134 patients, was handled by using a classification scheme which is based on statistical bootstrapping, and the patients were classified based on the average mean error. Rather than training and evaluating once, which results in a very unreliable classification based on pure coincidence, our proposed method leads to robust classification rates.

These methods resulted in disappointing correct classification rates around 70% for the best features. However, introducing a simple two-step classification scheme, where the patients first are divided into two groups based on DNA-ploidy, increased the CCR by 10%.

As a final experiment, the average structural features extracted from the object were combined with a texture feature from a previous study and used on the 102 patients with similar ploidy. Since the average feature values were used, this leads to a direct classification of the patients. This experiment resulted in a CCR close to 90%.

7.1 Suggestion for further study

The obvious extension to the work done in this thesis would be to include the texture feature value of each of the images, instead of doing it for each patient. This would probably improve the classification results for the analysis done in section 6.2 and 6.3, and a CCR above 90% might be reachable.

The material used in this project is just a subset of a larger data set, which is going to be used in a final test of the best method for prognostic classification. The best of the gray level texture features combined with the best structural features could be a candidate for such a test.

If time had not been an issue during the study, a closer look at the 32 patients in the group with tetraploid and polyploid patients would also be interesting. However, in this group there are only 4 patients with bad prognosis and the analysis of such a set would probably be difficult, and just by classifying all patients in this group as good prognosis would be tempting.

As mentioned, the vital step is the segmentation, and it is possible that another approach to could do better, but we did try out different segmentation approaches and they gave similar results.

The improved Niblack's method can also be useful in other applications. And with a further extension, our proposed algorithm could also handle estimation of (k,w) values within subsets of an image, but this was not relevant in our setting, since the images are rather small.

Bibliography

[1] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986.

[2] T.M. Cover. The best two independent measurements are not the two best. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-4(1):116 –117, 1974.

[3] O. Demirkaya and M.H. Asyali. Determination of image bimodality thresholds for different intensity distributions. *Signal Processing: Image Communication*, 19(6):507 – 516, 2004.

[4] M.M. Galloway. Texture analysis using gray level run lengths. *Computer Graphics and Image Processing*, 4(2):172 – 179, 1975.

[5] R. C. Gonzalez and R. E. Woods. *Digital Image Processing, third edition.* Pearson Education, Inc., 2008.

[6] R. M. Haralick, S. R. Sternberg, and X. Zhuang. Image analysis using mathematical morphology. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 9(4):532–550, 1987.

[7] R.M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, 3(6):610 –621, 1973.

[8] R.M. Haralick and L.G. Shapiro. *Computer and Robot Vision.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1992.

[9] A.K. Jain, R.P.W. Duin, and J. Mao. Statistical pattern recognition: a review. *IEEE Transactions on,Pattern Analysis and Machine Intelligence*, 22(1):4 –37, jan. 2000.

[10] J. Kittler and J. Illingworth. Minimum error thresholding. *Pattern Recognition*, 19(1):41–47, 1986.

[11] B. Nielsen, F. Albregtsen, and H.E. Danielsen. Prognostic classification of early ovarian cancer based on very low dimensionality adaptive texture feature vectors from cell nuclei from monolayers and histological sections. *Analytical Cellular Pathology*, 23(2):75–88, 2001.

[12] B. Nielsen, F. Albregtsen, and H.E. Danielsen. Low dimensional adaptive texture feature vectors from class distance and class difference matrices. *IEEE Transactions on,Medical Imaging*, 23(1):73 –84, 2004.

[13] B. Nielsen, F. Albregtsen, and H.E. Danielsen. Fractal analysis of monolayer cell nuclei from two different prognostic classes of early ovarian cancer. In *Fractals in Biology and Medicine*, Mathematics and Biosciences in Interaction, pages 175–186. Birkhäuser Basel, 2005.

[14] B. Nielsen, F. Albregtsen, and H.E. Danielsen. Statistical nuclear texture analysis in cancer research: A review of methods and applications. *Critical Reviews in Oncogenesis*, 14(2):89 –164, 2008.

[15] B. Nielsen and H.E. Danielsen. Prognostic value of adaptive features - the effect of standardizing nuclear first-order statistics and mixing information from nuclei having different area. *Cellular Oncology*, 28(3):85 – 95, 2006.

[16] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62 –66, jan. 1979.

[17] P. Pudil, J. Novovičová, and J. Kittler. Floating search methods in feature selection. *Pattern Recogn. Lett.*, 15:1119–1125, November 1994.

[18] S.J. Raudys and A.K. Jain. Small sample size effects in statistical pattern recognition: recommendations for practitioners. *IEEE Transactions on, Pattern Analysis and Machine Intelligence*, 13(3):252 –264, mar. 1991.

[19] T.W. Ridler and S. Calvard. Picture thresholding using an iterative selection method. *IEEE Transactions on Systems, Man and Cybernetics*, 8(8):630 – 632, aug. 1978.

[20] H. Schulerud and F. Albregtsen. Effects of many feature candidates in feature selection and classification. In *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pages 480–487, London, UK, 2002. Springer-Verlag.

[21] S. Theodoridis and K. Koutroumbas. *Pattern Recognition, Third Edition*. Academic Press, Inc., Orlando, FL, USA, 2006.

[22] Ø.D. Trier and A.K. Jain. Goal-directed evaluation of binarization methods. *IEEE Transaction on, Pattern Analysis and Machine Intelligence*, 17(12):1191–1201, 1995.

[23] Ø.D. Trier, A.K. Jain, and T. Taxt. Feature extraction methods for character recognition-a survey. *Pattern Recognition*, 29(4):641 – 662, 1996.

[24] A.W. Whitney. A direct method of nonparametric measurement selection. *IEEE Transactions on Computers*, C-20(9):1100 – 1103, 1971.

[25] S. D. Yanowitz and A. M. Bruckstein. A new method for image segmentation. *Comput. Vision Graph. Image Process.*, 46(1):82–95, 1989.