KU LEUVEN

Classification of human epithelial cell's staining patterns

Sebastijan Dumancic

Thesis submitted for the degree of
Master of Science in Artificial
Intelligence

Thesis supervisor:
Prof. dr. Hendrik Blockeel

Mentor:
Antoine Adam

Academic year 2013 – 2014

Preface

I would like to thank everybody who kept me busy the last year, especially my promotor and my assistants. I would also like to thank the jury for reading the text. My sincere gratitude also goes to my wive and the rest of my family.

Sebastijan Dumancic

Contents

Preface i

Abstract iv

List of Figures and Tables v

List of Abbreviations and Symbols vi

1 Introduction 1
- 1.1 Motivational Scenario . . . 2
- 1.2 Detecting the autoimmune diseases . . . 2
- 1.3 Problem statement . . . 3

2 Background 5
- 2.1 Understanding the domain . . . 5
- 2.2 Machine learning . . . 8

3 Related work 9
- 3.1 Cell segmentation . . . 9
- 3.2 Intesity level classification . . . 10
- 3.3 Staining pattern classification . . . 10

4 Cell segmentation 13
- 4.1 Background segmentation . . . 13
- 4.2 Positioning a prior . . . 21
- 4.3 Precise contours with Morphological snakes . . . 22

5 Fluorescence intensity classification 23
- 5.1 Classifying intensity . . . 23

6 Describing cells 27
- 6.1 Interesting features . . . 27
- 6.2 Deep learning . . . 27
- 6.3 Deep belief networks . . . 27
- 6.4 Evaluation . . . 27

7 Rule mining 29
- 7.1 Inductive logic programming . . . 29
- 7.2 Aleph . . . 29
- 7.3 FOIL . . . 29

Bibliography **33**

Abstract

The `abstract` environment contains a more extensive overview of the work. But it should be limited to one page.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

List of Figures and Tables

List of Figures

1.1 Cell examples 3

2.1 Illustration of antibody structure 6

3.1 Bad segmentation example 9

4.1 Example of an image histogram 15
4.2 Approximation of an image histogram 20
4.3 A calculation of a threshold 20
4.4 Hough transformation with all detected circles (a) and after removing background circles (b) 22

List of Tables

List of Abbreviations and Symbols

Abbreviations

LoG	Laplacian-of-Gaussian
MSE	Mean Square error
PSNR	Peak Signal-to-Noise ratio

Symbols

42	"The Answer to the Ultimate Question of Life, the Universe, and Everything" according to [1]
c	Speed of light
E	Energy
m	Mass
π	The number pi

CHAPTER 1

Introduction

In the last couple of decades, computers have found a number of applications in biology and medicine. We may even say they have become an essential tool in revealing the questions of life. A significant role in those problems was played by machine learning, a branch of artificial intelligence concerned with the construction of models capable of learning from data . Probably the most inspiring example comes from the field of bioinformatics where scientists have used the methods from statistics and artificial intelligence to sequence the human genome for the first time. If that was one of the first results, then, what can we expect from the future?

Besides the genomic data which is represented as a sequence of nucleotides, a great amount of biological data can be aquired by different imaging methods such as microscopy, PET or CT imaging and others. That is were the medical imaging and bioimage analysis fields come from. The field of bioimage analysis studies the biological problems by examining an image, or image sequence of a process of interest, while the medical image analysis is concerned with developing of methods that will help in a medical diagnosis process based on imaging.

The above mentioned field of medical image analysis overlaps with a field of computer aided diagnosis (CAD) which collects a broader range of methods used as an assistance to the doctors. An application area of this Thesis would fit the best in that field. This thesis will take a direction in a specific task of assistance to an autoimmune disease diagnosis, with a special emphasis to human interpretable models. During the recent few years, this specific problem has been solved very efficiently. If the problem has been solved, why is this thesis taking another look at it?

The goal of the Thesis is to provide a CAD method capable of making a decision based on a microscopic image of HEp-2 cell in a human intepretable way. To demonstrate the importance of intepretable model in CAD systems, I consider the following scenario.

1.1 Motivational Scenario

Imagine a following scenario. *Peter* is a doctor, an immunologist. Being an immunologist, his job is to find out which autoimmune disease a patient has. As that is an extremly hard task, Jules uses computer assistance - an artifical intelligence program named *Hal*. Hal's role is to confirm Peter's diagnosis, or to provide an additional insight if the decision is hard to make.

We are interested in the situation where Peter and Hal have made a conflicted decision. There are two cases representing the insight Hal can provide. Each case reflects Hal's interior structure : a **black-box** model[1] or an **interpretable** one. If Hal is the black-box model, it is not much of an assistance, as Hal can't elaborate it's decision. We can only agree we disagree. In this case, Hal is not much of a help.

On the other hand, if Hal is an interpretable model, it can elaborate it's decision and help Peter. If a wrong decision was made by Hal, Peter can observe it's model parameters, correct the wrong one(s), query again and see if new result supports it's decision. If a wrong decision was made by Peter, Peter can observe Hal's model, find the parameters he might have overseen and correct his diagnosis.

The example clearly demonstrates one thing – the importance of interpretable models for CAD systems. Computers have proven their ability of inferring from a large amount of data, usually outperforming humans, but in specific situations, a good and accurate model is not enough. We also need to interpret results if we are going to use them.

1.2 Detecting the autoimmune diseases

As mentioned already, this Thesis will address the specific topic of autoimmune diseases classification. The human immune system creates antibodies to fight against infections whereas antinuclear antibodies affect healthy tissues. The Antinuclear Autoantibodies test (ANA) is widely used to determine whether the immune system is developing antibodies or not. Indirect immunofluorescence (IIF) with HEp-2 cells is the recommended method to diagnose the presence of antinuclear autoantibodies in patient serum.

IIF method consists of four consecutive steps:

- cell segmentation
- intensity level classification
- mitosis detection

[1] By black-box model we assume a model for which we can only observe it's output, not a decision process

- staining pattern classification

The final step usually classifies cell images into one of following patterns: homoge neous, speckled, nucleolar, cytoplasmic, centromere (see figure 1.1). Some variations may be introduced in different datasets due to large number of possible patterns. Those staining patterns further have a clinical associations with a specific autoimmune disease such as Scleroderma, malignant tumor, Lupus nephritis and others.

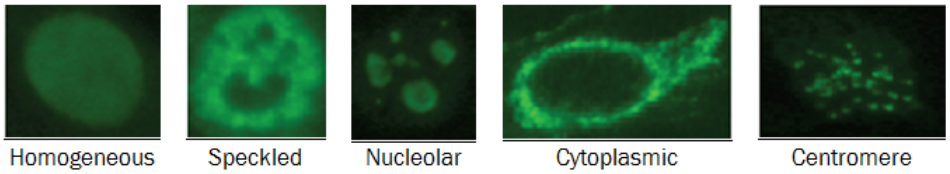


FIGURE 1.1: HEp-2 cell patterns

Although IIF posses qualities such as high sensitivity and a large number of antigens that can be detected, it suffers from numerous shortcomings. The most important ones are liability to subjectivity and time and labour consuming.

In order to avoid any kind of subjectivity , there is a great need for standardization and formalization of the mentioned procedure. Addressing this problem calls for CAD techniques which combines methods from machine learning and image analysis.

1.3 Problem statement

The content of the thesis proposes solutions for three major steps in the IFF process.

First of all, in order to find out a cell's pattern, cells should be isolated from an image aquired by IFF. As it is going to be explained (see chapter 3), although the problem of cell segmentation has been studied for more than 50 years, segmenting HEp-2 cells still suffers from certain problems, mostly due to different green floures-cent protein apsorbtion accross the cells. This thesis proposes a method based on a region growing algorithm that demonstrates an encouraging result for overcoming mentioned problems.

Once we have isolated the cells, the next step is to determine the fluorescent intensity of an image. The fluorescent intensity level can take three different values, namely *positive*, *intermediate* and *negative*. A negative value means there are no observable cells in an image, while positive marks easily observable cells.

The final step presents a novel approach to the staining pattern classification. The current state of the art solution solves the problem very successfully, but acts like black-box solutions not providing any explanation for the decision. This thesis tends to develop a method based on the human interpretable representation and reasoning. As IF-THEN rules are the most natural way of representing human knowledge, the thesis will follow a rule mining approach, such as Inductive Logic Programming (ILP).

CHAPTER 2

Background

2.1 Understanding the domain

2.1.1 Immune system and antibodies

The immune system is the central part of the human body responsible for protection against infections. In order to function properly, the immune system has to detect a wide range of threats, and at the same time distinguish them from healthy tissue. The main weapon the immune system can use are antibodies.

The antibody is a protein complex produced by *B cells* [1] that initiates an immune response against a target antigen [2]. Their primal role is to recognize the unique part of the foreign target and protect the body from infections. The basic organization of the antibody includes two functional domains that, together, resemble the letter Y (Figure 2.1, left). The *Fab* part makes up the arms of the Y, and it contains the antigen-binding site - the region responsible for antigen binding. The *Fc* part comprises the tail of the Y and effects other cells, proteins and antibodies.

This unique structure allows detection of antigens, in direct or indirect manner. By direct detection we assume detection using a single fluorophore-labeled antibody, and by indirect detection we assume detection through binding of a fluorophore-labeled secondary antibody raised against the Fc part of an unlabeled primary antibody (as illustrated in Figure 2.1, right). This system is versatile and cost-effective because few labeled antibodies are required to detect many possible primary antibodies.

[1] a subgroup of white blood cells, a viral part of the immune system

[2] Foreign substance that, when introduced into the body, is capable of stimulating an immune response

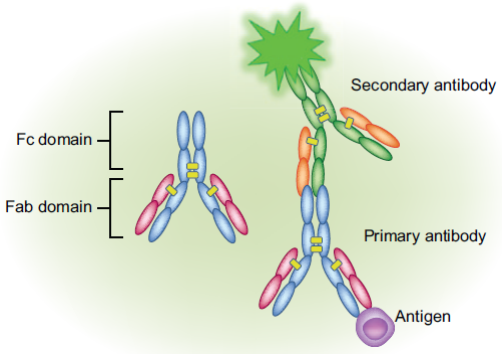


FIGURE 2.1: Illustration of antibody structure (from [6])

The immune system can sometimes suffer from different disorders. A disorder of special importance to this Thesis is *autoimmunity*. Autoimmunity results in the disability of the immune system to recognize an organism's healthy tissue, and therefore attacks normal tissues as if they were foreign organisms. In the case of autoimmunity, antibodies are called antinuclear antibodies. We can observe those antibodies by using indirect immunofluorescence.

2.1.2 Indirect Immunofluorescence

As it was already mentioned, the indirect detection is the main focus of the thesis. Indirect immunofluorescence is a diagnostic methodology based on image analysis that reveals the presence of autoimmune diseases by searching for antibodies in the patient serum. Indirect immunofluorescence is a two-step technique, in which a primary, unlabeled antibody binds to the target, after which a fluorophore-labeled[3] second antibody is used to detect the first antibody (figure 2.1, right). Indirect immunofluorscence is more sensitive than a direct one because more than one secondary antibody can bind to each primary antibody, which amplifies the fluorescence signal.

As a result of its effectiveness,there has been a growing demand for diagnostic tests for systemic autoimmune diseases. Unfortunately, IIF still remains a subjective method that depends too heavily on the experience and expertise of the physician. The main reasons causing the problems are:

- the lack of quantitative information supplied to physicians
- varieties of reading systems and optics

[3]fluorophore-labeling is a method to color the antibodies so they can be observed under the microscope

- the photo-bleaching effect caused by a light source irradiating the cells over a short period of time
- the low reproducibility of the diagnostic protocol.

2.1.3 Putting it all together

The focus of this thesis is on the Antinuclear antibodies test (ANA), which plays the main role in the serological[4] diagnosis of autoimmune disease. ANAs are directed against a variety of antigens and can be detected in patient serum through laboratory tests. IIF uses the human epithelial (HEp-2) substrate, which bonds with serum antibodies forming a molecular complex. This complex then reacts with human immunoglobulin [5] and becomes visible under a fluorescence microscope which reveals the antigen-antibody reaction.

The procedure of ANA starts with fluorescence intensity classification, a segmentation step is not a part of ANA procedure. The Center for Disease Control and Prevention in Atlanta,USA have published guidelines [8] for scoring the intensity. The score ranges from 0 to 4+ as follows:

- 4+ : brilliant green (maximal fluorescence)
- 3+ : less brilliant green fluorescence
- 2+ : defined pattern but diminished fluorescence
- 1+ : very subdued fluorescence
- 0 : negative.

Although the guidelines provide very detailed instructions, in [12] Rigon et al. analyzed the variability between a set of physician's fluorescence intensity classifications. Their work has shown a big variance of classifications made by physicians on the same dataset, so they suggested to classify the fluorescence intensity into three classes, namely negative, intermediate and positive. This work follows the protocol.

The final step consists of staining pattern recognition. As shown in figure 1.1, there are several patterns that may be observed. [3] and [11] provide a description of all staining patterns which is a valuable input taking into consideration a human interpretable perspective of this step. A summary is presented here:

- **Centromere**: characterized by several discrete speckles ($\sim$ 40 - 60) distributed throughout the interphase[6] nuclei and characteristically found in the condensed nuclear chromatin during mitosis as a bar of closely associated speckles.

[4] Further explanation
[5] Immunoglobulin is a specific type of antibody created by plasma cells
[6] The interphase is the nonmitotic phase of the cell cycle in which the cell spends the majority of its time and performs the majority of its purposes

- **Nucleolar**: characterized by clustered large granules in the nucleoli of interphase cells which tend towards homogeneity, with less than six granules per cell.
- **Homogeneous**: characterized by a diffuse staining of the interphase nuclei and staining of the chromatin of mitotic cells.
- **Fine Speckled**: characterized by a fine granular nuclear staining of the interphase cell nuclei.
- **Coarse Speckled**: characterized by a coarse granular nuclear staining of the interphase cell nuclei.
- **Cytoplasmatic**: characterized by a highly irregular shape and large granule in the nucleoli

2.2 Machine learning

CHAPTER 3

Related work

3.1 Cell segmentation

Several studies have been proposed to classify autoantibody fluorescence patterns by using an automatic thresholding method, i.e. Otsu's method, to segment the cells. The thresholding method can choose the threshold to minimize the intraclass variance of the black and white pixels automatically. Due to the variety of ANA patterns, Otsu's algorithm always failed to segment cells of speckled and nucleolar patterns, such as cases of a very blury image of low intensity. Figure 3.1 shows the over-segmentation results by using Otsu's algorithm. Additional challenge for the segmentation here is to separate overlapping cells which are quite common in the process.

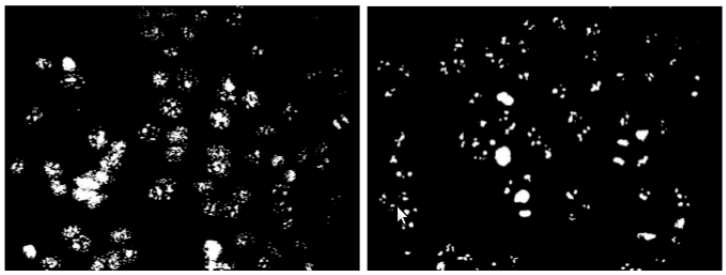


FIGURE 3.1: Segmentation results of the Otsu method (from [5])

In [5], Huang et all. present an adaptive edged-based segmentation method for automatically detecting outlines of fluorescence cells in IIF images. Their approach is based on specific properties of the images regarding the pattern class. They have divided the images in two groups : sparse region and mass region cells. The mass region cells are those ones which have a *compact* appearance, that look like a smooth object, while the sparse region cells are those ones for which we can detect multiple

object in a cell. The approach trains a classifier to classify each image in the groups and applies different segmentation procedure for each group. In the case of the mass region cells, the cells are segmented using Otsu segmentation method, while in the case of the sparse region cell segmentation is performed by an edge detection. Their approach resulted in better segmentation results, but approximately 10% of the cells remained undetected.

In [4], same authors further improve their method by incorporating watershed segmentation. The second approach also includes segmentation in two stages, depending on defined criteria. After a segmentation step performed by the watershed, the approach merges parts located relatively close and eliminates parts not large enough to represent a cell. If the retrieved number of regions doesn't safisfy the defined criteria, the segmentation step is performed again with different parameter settings determinated by Otsu's thresholding.

All forementioned approaches report similar shortcomings : approximately 10% of cells remained undetected and the inability to separate overlapping cells. The focus of the segmentation part of the Thesis will be on overcoming those problems.

3.2 Intesity level classification

The following step, the intensity level classification, hasn't attracted a lot of scientific research, but has demonstrated remarkable results so far.

In [13], authors propose a system based on *Multi-Layer Perceptrons* and a *Radial Basis Network* for the intensity classification step. That system, which makes use of features inspired from medical practice, shows error rates up to 1%, but it uses a reject option and it does not cast a result in about 50% of cases. In [14] the authors further refine their system. They train three experts, one specialized for each class, with a different set of features. They threat the classifiers similar to the *one-vs-all* approach, so the final decision is made by a classifier most certain in it's decision. The authors report a success rate of 92,6% accuracy.

3.3 Staining pattern classification

As this problem was emphasized on the *International Conference on Pattern Recognition 2012* as a contest, this step has been well researched and several very successful methods have been proposed. In [3], Foggia et al. provides a detailed overview of the methods submitted for the contest. The three most successful ones are presented here.

In [7], Kuan presents a method based on four texture descriptors: a rotation invariant form of local binary patterns (LBP) with multi-scale analysis,discrete cosine transformation, the mean values and standard variances of 2-D Gabor wavelets, and some global appearance based statistical features. A multiclass SVM was trained on each class of the four feature sets. The SVMs are then integrated in one classifier by using the AdaBoost.M1 algorithm.

In [10], Nosaka presents a similar approach on an extension of LBP, namely CoALBP [9]. The advantage of this method is that the method can observe not only locals LBP, but also the spatial relations among adjacent LBP. The classifier is a linear SVM trained on an extended dataset including the rotated patterns of the original images.

Xianfeng et al. proposed a system based on MR8 method [15] to extract statistical intensity features. The method calculates filter responses locally on the image, and then trains a global texton dictionary using *K*-means clustering. In that way, each image is represented by the frequency histogram of textons. The decision is made by a *k*-NN classifier.

Although there are many more papers in existance covering this problem, they are not presented here due to different, and not so rigorous, evaluation. Most of the early work was done on private datasets not available to public, which makes them not comparable to the new research results. Other papers with a more recent date do not follow the evaluation procedure, so it is very hard to compare their efficiency with those ones in the overview.

More recently, a new overview of the method was presented by Agrawal et al. in [2]. The committee of *ICPR'13* has released a new, much bigger dataset for the same problem. The authors experimented with the most commonly used features in the previous contest - statistical features, histograms of oriented gradients, shape and size descriptors and texture descriptors. They have chosen the most typical representatives of classifiers, namely Naive Bayes, k-NN, SVM and Random forest. The SVM with Law's textural representation significantly outperformed other classifiers and feature representations.

CHAPTER 4

Cell segmentation

Finding the cells in images is the first and crucial step in the procedure. The problems encountered in the cell segmentation are discussed in Chapter 3. All of the problems occur due to coloring procedure that is time dependent, so cells expose different intensity levels over an image. Because of that, part of cells remains undetected and overlapping. The challenge here is to find a method that can overcome different illumination of objects. Proposed solution deals with mentioned problems in separate steps.

The proposed solution is based on an observation that, although cells exhibit different properties across image due to different illumination levels, the background is uniform over a whole image and exhibit constant properties. Following the observations, as summarized in algorithm X, the method first segments the background to find locations of the cells and then uses second segmentation step to refine the borders of cells.

4.1 Background segmentation

The first step of background segmentation is intended to overcome the problem of undetected cells. A desirable property of segmentation algorithm for this task is the capability of capturing the global properties of the background region across an image. One such algorithm is *Region growing*. Region growing is a simple method that extends a region based on a similarity in intensity levels - the intensity of each candidate pixel is compared with a mean intensity of a current region. If a difference is less than a given threshold, the pixel is added to region. The algorithm is summarized in algorithm 1.

The bottleneck of the method is the evaluation of candidate pixels. If target regions are big, as in this case, the method could be very slow. To overcome this bottle

Algorithm 1 Region Growing

```
1: function REGIONGROWING(starting point, criteria)
2:     candidates ← {neighborhood(starting point)}
3:     region ← {starting point}
4:     currentpoint ← starting point
5:     while no new candidates do
6:         for each candidate of current point do
7:             if candidate satisfies criteria then
8:                 add it to region
9:                 update parameters of region
10:            end if
11:        end for
12:        currentpoint ← next point from border
13:    end while
14: end function
```

neck, the method is extended with a pre-filling step in which a part of an image is assigned to belong to the background by default. As it can be noticed from images, the majority of image is a dark background which can be observed in an image histogram as the highest peak. To speed up the segmentation, every pixel with intensity lower or equal to the highest peak is automatically assigned to background.

The critical issue here is the estimation of a threshold for comparison. It is a parameter that depends on a distribution of intensities over an image. In a sense, this parameter models a variance in intensity of the background region. Motivated by a variance fitting, the proposed solution is based on an assumption that an image histogram should consist of two regions - one modeling the background and a second one modeling cells, as illustrated in image 4.1. The goal now is to find those two regions in a histogram. One way to accomplish that is to approximate a histogram with a mixture of Gaussian functions. This approach is taken here.

4.1.1 Gaussian mixture model

Mixture density is a linear combination of K probabilistic density functions:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k p(\mathbf{x}|\theta_k) \tag{4.1}$$

where $p(\mathbf{x}|\theta_k)$ represents mixture components with their parameters θ_k. In out case, those mixture components are Gaussian functions $\mathcal{N}(\mu_k, \Sigma_k)$. Parameters π_k are mixture coefficients that satisfy $1 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$. As they satisfy given criteria, those coefficient can be treated as prior probability of a component k.

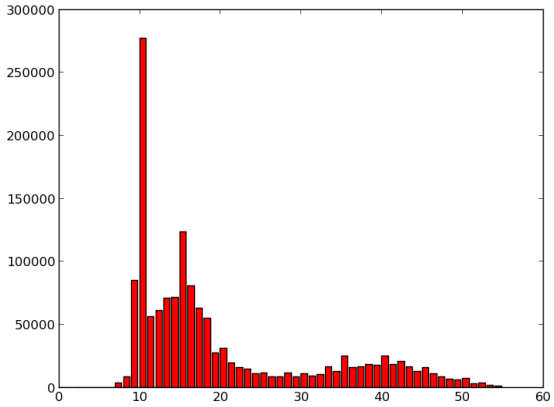


FIGURE 4.1: Example of an image histogram

Equation 4.1 can be then rewritten as :

$$p(\mathbf{x}) = \sum_{k=1}^{K} P(\mathcal{G}_k) p(\mathbf{x}|\mathcal{G}_k), \tag{4.2}$$

where $P(\mathcal{G}) = \pi_k$ and $p(\mathbf{x}|\theta_k) = p(\mathbf{x}|\mathcal{G}_k)$. Out task is to determine the parameters

$$\theta = \{P(\mathcal{G}_k), \theta_k\}_{k=1}^{K}, \tag{4.3}$$

or more precisely for the Gaussian function

$$\theta = \{P(\mathcal{G}_k), \mu_k, \Sigma_k\}_{k=1}^{K}. \tag{4.4}$$

An efficient algorithm to determine those values is the Expectation Maximization algorithm.

4.1.2 Expectation Maximization Algorithm

The goal of the Expectation Maximization algorithm is to find parameters θ that maximize the log-likelihood

$$\mathcal{L}(\theta|\mathcal{D}) = \ln \prod_{i=1}^{N} p(\mathbf{x}^{(i)}) = \ln \prod_{i=1}^{N} \sum_{k=1}^{K} p(\mathbf{x}^{(i)}|\theta_k) = \sum_{i=1}^{N} \ln \sum_{k=1}^{K} p(\mathbf{x}^{(i)}|\theta_k). \tag{4.5}$$

Unfortunately, there is no closed-form solution for this formulation. Model $p(\mathbf{X}|\boldsymbol{\theta})$ is extended with latent variable $\mathbf{Z}$ which determines to which cluster belongs every x_i. The density function is now described with $p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$. Marginal density $p(\mathbf{X}|\theta)$, which is density we want to estimate, can always be reconstructed from joint probability by marginalization :

$$p(\mathbf{X}|\theta) = \sum_{\mathbf{Z}} p(\mathbb{X}, \mathbf{Z}|\boldsymbol{\theta}).$$

Log-likelihood we try to maximize is now

$$\ln\mathcal{L}(\boldsymbol{\theta}|\mathbf{X}) = \ln p(\mathbf{X}|\boldsymbol{\theta}) = \ln \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}). \tag{4.6}$$

As values of the latent variable $\mathbf{Z}$ are not yet known, we still can't work with log-likelihood directly. Instead, we will work with the expectation of the log-likelihood $\mathbb{E}[\ln\mathcal{L}(\boldsymbol{\theta}|\mathbf{X}, \mathbf{Z})]$. The main idea behind the EM-algorithm is to iteratively adjust parameters $\boldsymbol{\theta}$ to maximize the expectation.

The maximization of $\mathbb{E}[\ln\mathcal{L}(\boldsymbol{\theta}|\mathbf{X}, \mathbf{Z})]$ is now done by switching between two steps – E-step and M-step. E-step (*expectation step*) calculates the expectation of the log-likelihood with respect to current values of the parameters $\boldsymbol{\theta}^{(t)}$. That expectation $\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ we can calculate as

$$\begin{aligned} \mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) &= \mathbb{E}_{\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{(t)}}[\ln\mathcal{L}(\theta|\mathbf{X}, \mathbf{Z})] \\ &= \mathbb{E}_{\mathbf{Z}|\mathbf{X},\theta^{(t)}}[\ln p(\mathbf{X}, \mathbf{Z}|\theta)] \\ &= \sum_{\mathbf{Z}} P(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(t)}) \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}). \end{aligned} \tag{4.7}$$

The expectation is now expressed on variable $\mathbf{Z}$ with fixed values of $\mathbf{X}$ and θ. Probability $P(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(t)})$ is the a posteriori probability of the latent variable $\mathbf{Z}$ with fixed parameters which can be evaluated using Bayes rule.

Only thing left is to optimize the parameters $\boldsymbol{\theta}$. M-step (*maximization step*) chooses new parameters $\boldsymbol{\theta}^{(t+1)}$ by maximizing the expression

$$\boldsymbol{\theta}^{(t+1)} = \arg\max_{\theta} \mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}). \tag{4.8}$$

EM-algorithm is briefly summarized in Algorithm 4.1.2. Starting with the initial parameters $\boldsymbol{\theta}^{(0)}$, it iteratively switches between E and M step until converged. The convergence is guaranteed as algorithm maximizes the expectation in every iteration. On the other hand, found solution might not be the global optimum.

Algorithm 2 Expectation-maximization algorithm

```
1: function EM
2:     Initialize parameters θ^(0)
3:     t ← 0
4:     repeat
5:         E-step: calculate P(Z|X, θ^(t))
6:         M-step: θ^(t+1) ← arg max_θ Q(θ|θ^(t))
             where Q(θ|θ^(t)) = Σ_Z P(Z|X, θ^(t)) ln p(X, Z|θ)
7:         t ← t + 1
8:     until converged
9: end function
```

4.1.3 EM algorithm for mixture models

The EM-algorithm for mixture models works as follows. Let $\mathbf{z} = (z_1, \ldots, z_K)$ be a latent variable vector indicating a cluster to which an example belong to. $z_k = 1$ if an example belongs to a cluster $\mathcal{G}_k$, $z_k = 0$ otherwise. Each cluster is assigned with a prior probability

$$P(z_k = 1) = \pi_k,$$

such that $\sum_k \pi_k = 1$. Distribution over a latent variable $\mathbf{z}$ hence can be expressed as

$$P(\mathbf{z}) = \prod_{k=1}^{K} \pi_k^{z_k}. \tag{4.9}$$

To evaluate the probability $p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})$ we are still missing the factor $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$. It can be expressed as

$$p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = \prod_{k=1}^{K} p(\mathbf{x}|\boldsymbol{\theta}_k)^{z_k}. \tag{4.10}$$

Joint probability $p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})$ can now be expressed as

$$p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) = P(\mathbf{z})p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = \prod_{k=1}^{K} \pi_k^{z_k} \prod_{k=1}^{K} p(\mathbf{x}|\boldsymbol{\theta}_k)^{z_k}. \tag{4.11}$$

Using the obtained model 4.11, the log-likelihood $\ln\mathcal{L}(\boldsymbol{\theta}|\mathcal{D}, \mathcal{Z})$ can be expressed. Take $\mathcal{D} = \{\mathbf{x}(i)\}_{i=1}^{N}$ as a set of examples and $\mathcal{Z} = \{\mathbf{z}^{(i)}\}_{i=1}^{N}$ as a set of latent variables describing the relationship between examples and model. Log-likelihood of the model 4.11 is

$$\begin{aligned} \ln\mathcal{L}(\boldsymbol{\theta}|\mathcal{D}, \mathcal{Z}) &= \ln \prod_{i=1}^{N} p(\mathbf{x}^{(i)}, \mathbf{z}^{(i)}|\boldsymbol{\theta}) = \ln \prod_{i=1}^{N} \prod_{k=1}^{K} \pi_k^{z_k^{(i)}} p(\mathbf{x}^{(i)}|\boldsymbol{\theta}_k)^{z_k^{(i)}} \\ &= \sum_{i=1}^{N} \sum_{k=1}^{K} z_k^{(i)} \left(\ln\pi_k + \ln p(\mathbf{x}^{(i)}|\boldsymbol{\theta}_k) \right). \end{aligned} \tag{4.12}$$

E-step

In E-step of the algorithm we need to calculate the expectation $\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$:

$$\begin{aligned}\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) &= \mathbb{E}_{Z|\mathcal{D},\theta^{(t)}}[\ln\mathcal{L}(\boldsymbol{\theta}|\mathcal{D},\mathcal{Z})] \\ &= \mathbb{E}_{Z|\mathcal{D},\theta^{(t)}}\left[\sum_{i=1}^{N}\sum_{k=1}^{K} z_k^{(i)}\left(\ln\pi_k + \ln p(\mathbf{x}^{(i)}|\boldsymbol{\theta}_k)\right)\right] \\ &= \sum_{i=1}^{N}\sum_{k=1}^{K}\mathbb{E}\left[z_k^{(i)}|\mathcal{D},\boldsymbol{\theta}^{(i)}\right]\left(\ln\pi_k + \ln p(\mathbf{x}^{(i)}|\boldsymbol{\theta}_k)\right).\end{aligned} \tag{4.13}$$

As z_k^i is a Bernoulli variable (as only one component of $\mathbf{z}^{(i)}$ can be 1) which depends only on one example from $\mathcal{D}$, $\mathbf{x}^{(i)}$, it holds

$$\mathbb{E}\left[z_k^{(i)}|\mathcal{D},\boldsymbol{\theta}^{(t)}\right] = \mathbb{E}\left[z_k^{(i)}|\mathbf{x}^{(i)},\boldsymbol{\theta}^{(t)}\right] = P(z_k^{(i)}=1|\mathbf{x}^{(i)},\boldsymbol{\theta}^{(i)}). \tag{4.14}$$

The expectation of the latent variable equals to its *a posteriori* probability and it can be evaluated by applying the Bayes rule:

$$\begin{aligned}P(z_k^{(i)}=1|\mathbf{x}^{(i)},\boldsymbol{\theta}^{(i)}) &= \frac{p(\mathbf{x}^{(i)}|z_k^{(i)}=1,\boldsymbol{\theta}^{(t)})P(z_k^{(i)}=1)}{\sum_{j=1}^{K}p(\mathbf{x}^{(i)}|z_j^{(i)}=1,\boldsymbol{\theta}^{(t)})P(z_j^{(i)}=1)} \\ &= \frac{p(\mathbf{x}^{(i)}|z_k^{(i)}=1,\boldsymbol{\theta}^{(t)})\pi_k^t}{\sum_{j=1}^{K}p(\mathbf{x}^{(i)}|z_j^{(i)}=1,\boldsymbol{\theta}^{(t)})\pi_j^t} \equiv h_k^{(i)}.\end{aligned} \tag{4.15}$$

Hence, the expectation of the log-likelihood equals

$$\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \sum_{i=1}^{N}\sum_{k=1}^{K} h_k^{(i)}\ln\pi_k + \sum_{i=1}^{N}\sum_{k=1}^{K} h_k^{(i)}\ln p(\mathbf{x}^{(i)}|\boldsymbol{\theta}_k). \tag{4.16}$$

M-step

M-step maximizes the log-likelihood obtained in 4.16. The solution is found analytically by solving $\nabla_{\boldsymbol{\theta}}\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$. To obtain the mixture components, $\pi_k^{(t+1)}$, the equation is $\nabla_{\pi_k}\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$. As the second term in 4.16 does not depend on π_k, so it can be ignored. Starting from

$$\nabla_{\pi_k}\left(\sum_{i=1}^{N}\sum_{k=1}^{K} h_k^{(i)}\ln\pi_k + \lambda\left(\sum_k \pi_k - 1\right)\right) = 0, \tag{4.17}$$

where *Langrange multiplier* is used to satisfy the condition $\sum_{k=1}^{K} = 1$, it can be easily obtained

$$\pi_k^{(t+1)} = \frac{1}{N} \sum_{i=1}^{N} h_k^{(i)}. \tag{4.18}$$

To find parameters of each component, $\boldsymbol{\theta}_k^{(i)}$, the equation $\nabla_{\theta_k} \mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = 0$ has to be solved. Now first term in the equation 4.16 does not depend on $\boldsymbol{\theta}_k$ so it can be ignored

$$\nabla_{\theta_k} \sum_{i=1}^{N} \sum_{k=1}^{K} h_k^{(i)} \ln p(\mathbf{x}^{(i)}|\boldsymbol{\theta}_k). \tag{4.19}$$

Substituting $p(\mathbf{x}^{(i)}|\boldsymbol{\theta}_k)$ with a multidimensional Gaussian distribution and deriving with regards to $\boldsymbol{\mu}_k$ and σ_k^2 new parameters are calculated

$$\boldsymbol{\mu}_k^{(t+1)} = \frac{\sum_i h_k^{(i)} x^{(i)}}{\sum_i h_k^{(i)}} \tag{4.20}$$

$$(\sigma^2)_k^{(i)} = \frac{\sum_i h_k^{(i)} (x^{(i)} - \mu_k^{(t+1)})(x^{(i)} - \mu_k^{(t+1)})^T}{\sum_i h_k^{(i)}}. \tag{4.21}$$

4.1.4 Threshold estimation

Once histogram has been approximated with two Gaussian functions, obtained functions are used to find a threshold. Such approximation is illustrated in figure 4.1.4. A Gaussian with a lower mean is assumed to model the background. By examining its variance, the threshold is estimated. The threshold is now defined as a distance from the mean to a point where two Gaussian are equally probable. Additional restriction that has to be satisfies is that this point should be placed between the means of Gaussians. The principle used for calculating the threshold is illustrated in figure 4.1.4.

The point of using the background segmentation step is to overcome a problem of undetected cells that expose a low intensity. To evaluate this step, the optimal threshold for each image in dataset was determined manually. The evaluation was perform in two ways - on object and pixel level. First, the number of detected cells was used as the number of cells in each image is known in advance. Second, segmented region were compared to the ground truth. As a measure of similarity *precision* is used - number of pixels in segmented regions that is contained in the ground truth segmentation divided by the number of pixels in the ground truth

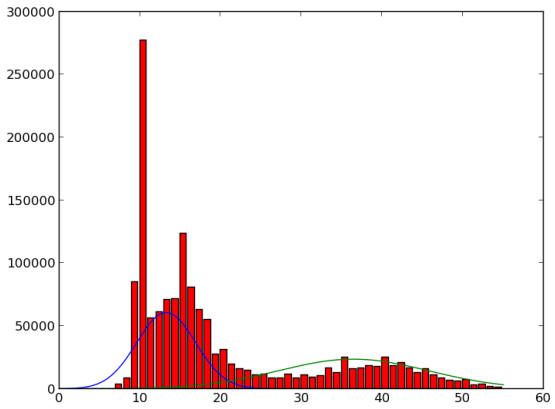


FIGURE 4.2: Approximation of an image histogram

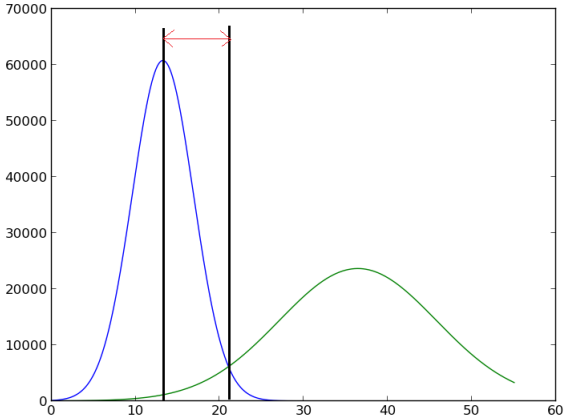


FIGURE 4.3: A calculation of a threshold

segmentation. This measure should indicate a similarity of obtained segmentation to a manually segmented cells.

Table **??** summarizes the object level results. The results show that the Region growing approach usually detects more objects than contained in the ground truth segmentation. Those extra object detected by the proposed approach are actually partial cells that were removed from the ground truth segmentation. The presences of those partial cells demonstrates the advantages and sensitivity of region growing, but also raises a question should those cells be eliminated from further process or used as a *fully observed* cells.

4.2 Positioning a prior

Once the background has been found, we have a rough estimate of cell positions. Still, a lot of cells overlap. In order to split them, we will make use of circularity properties of the cells. If circles of cells could be detected, or at least circular parts of cells, they might be split. To detect circles Hough transformation is used.

4.2.1 Hough transformation for circles

Hough transform is a general voting procedure used in Computer vision. The outline of the method is given in algorithm 3. It discretize a parameter space and assigns every bin a number of votes proportional to the number of edges in an image that could be generated with that specific bin parameters. In our case, each bin represents one candidate circle in an image with equation

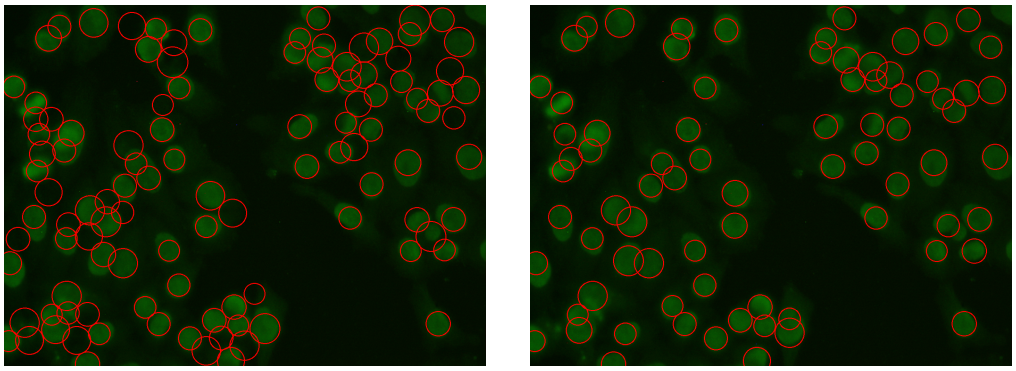
$$\mathcal{H}(\hat{x}, \hat{y}, r) = (\hat{x} - x)^2 + (\hat{y} - y)^2 = r^2, \tag{4.22}$$

where $\hat{x}$ and $\hat{y}$ represent the origin of a circle and r its radius. Now every edge point in an image *votes* for every bin that could have generated it, under assumption that the edge is a part of a circle.

Algorithm 3 Hough Transform

```
1: function HOUGH TRANSFORMATION
2:     initialize accumulator H to all zeros
3:     for each edge in image do
4:         increment every cell H(x, y, r) which could be the center of a circle
5:     end for
6:     search for local maxima cells of H
7: end function
```

After voting, every local maximum is selected as a circle. Additionally, every vote could be weighted proportionally to its magnitude. Note that not all local



(a) (b)

FIGURE 4.4: Hough transformation with all detected circles (a) and after removing background circles (b)

maxima are actual circles. Although quite straightforward, the method can easily become unfeasible. Searching for all radius's is obviously unfeasible for large images, so any restriction on circle size range is most certainly helpful. To help the search, radius is limited to a range of minimal to the largest radius found in the data set.

Another problem with Hough transformation is its sensitivity to noise. If we assume there is an uniform noise in an image, it is easy to conclude that might imply false circles over an image. Methods to overcome that problem usually include better discretization of parameters or smoothing in the accumulator by incrementing neighboring bins, but for fully automated application this is not suitable solution. Instead, information about background extracted in the previous step could be used. As we can obtain background quite successfully, every circle with a center in the background region can be removed. so as every other circle which has similar properties as the background circles. As a criteria here, a mean intensity in the green channel is used - every circle with a mean intensity lower than the highest mean intensity found in the background is removed. Figure 4.4 illustrates that selection. Now each circle serves as a seed point for precise segmentation of contours and splitting overlapping cells.

4.3 Precise contours with Morphological snakes

CHAPTER 5

Fluorescence intensity classification

The fluorescence intensity is an parameter that describes the *clarity* of cells in image. It is a subjective parameter which, unfortunately, doesn't have a strong theoretical explanation. The fluorescence intensity is described with three values - positive, intermediate and negative. Positive value defines images in which cells are perfectly separated from background, while negative value defines images in which cells in which cells cannot be identified. The intermediate value covers images that are not positive nor negative.

In [12] Rigon studied the variability of decision made by doctors and showed that it is hard to achieve a consensus about unique determination of the fluorescence intensity value. The lack of underlying model is making this problem hard to formulate. The intuition behind the suggested approach is an assumption that intensity level could be observed in the histogram of an image. The fluorescence intensity should correspond to the difference to a region describing the background and a region describing cells. Following the intuition, an image histogram is approximated with the Gaussian mixture model.

The idea is to approximate the histogram with a mixture of 2 Gaussian functions - one representing the background and second one to model the cells. The intuition is that images with positive intensity should have Gaussians with higher means and more further apart.

5.1 Classifying intensity

Once histogram has been approximated with two Gaussian functions, the estimated means and variances have taken as features for the classification. SVM with radial

basis functions as kernel function has been trained for the task. Evaluation is performed using a 10-fold cross validation.

5.1.1 Support Vector Machine

Support vector machine is a very popular machine learning technique. It is a representative of a more general class of *kernel methods*. The most interesting property of the support vector machine is that it tries to find the *optimal hyperplane* that separates classes. Consider a two-class case. The sample is $\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{N}$ where $y^{(i)} = 1$ if $\mathbf{x}^{(i)} \in \mathcal{C}_1$ and $y^{(i)} = -1$ if $\mathbf{x}^{(i)} \in \mathcal{C}_2$. We want to find a margin $\mathbf{w}$ so that

$$\mathbf{w}^T\mathbf{x}^{(i)} + w_o \geq +1 \quad \texttt{for} \quad y^{(i)} = 1$$

$$\mathbf{w}^T\mathbf{x}^{(i)} + w_o \leq -1 \quad \texttt{for} \quad y^{(i)} = -1$$

or simplified

$$y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + w_0) \geq 1.$$

The support vector machine extends the *basic* hyperplane requirement - setting the instances on the right side of the hyperplane - by trying to maximize the margin - the distance from the hyperplane to the instances closest to it. So, the *optimal separating hyperplane* is the one that maximizes the margin.

The distance of $\mathbf{x}^{(i)}$ to the hyperplane equals to

$$\frac{|\mathbf{w}^T\mathbf{x}^{(i)} - w_0|}{\|\mathbf{w}\|}. \tag{5.1}$$

Finding the maximal margin can be expressed as

$$\frac{y^{(i)}(\mathbf{w}^T\mathbf{x} - w_0)}{\|\mathbf{w}\|} \geq \rho, \forall i, \tag{5.2}$$

i.e., we want the margin to be at least ρ. As there are infinitely many solutions that can be obtained by scaling $\mathbf{w}$, we fix the $\rho\|\mathbf{w}\| = 1$ and minimize $\|\mathbf{w}\|$ to maximize the margin. The problem can now be written in a form of quadratic optimization problem

$$\begin{aligned} \min_{\mathbf{w}, w_0} \quad & \frac{1}{2}\|\mathbf{w}\| \\ \texttt{subject to} \quad & y^{(i)}\left(\mathbf{w}^T\mathbf{x} + w_0\right) \geq 1. \end{aligned}$$

Now there will be instances that are $\frac{1}{\|\mathbf{w}\|}$ away from the hyperplane and the total margin equals to $\frac{2}{\|\mathbf{w}\|}$.

If the problem is not linearly separable, one trick is to map the problem to a new space, with more dimension that the original space, by using nonlinear basis functions. It is generally the case that higher dimensional representation is easier to separate. One such mapping function is the *radial basis function or gps or google maps*.

5.1.2 Radial basis functions

Radial basis functions are an instance of a *local representation*, where fora given input, only a few factors are *active*. It is in a way a partition of a space so that *locally tuned* partitions are selective to only certain inputs.

With the concept of local partitioning we need to define a measure of similarity between an input $\mathbf{x}^{(i)}$ and the local clusters $\boldsymbol{\mu}^1, \ldots, \boldsymbol{\mu}^n$. The radial basis functions are defined as

$$\mathbf{r}(\mathbf{x}^{(i)}, \boldsymbol{\mu}^k) = exp\left(-\frac{\|\mathbf{x}^{(i)} - \boldsymbol{\mu}^k\|}{2\sigma_k^2}\right), \tag{5.3}$$

that is, it uses the Euclidean distance as a measure of similarity and Gaussian function as a response function. The response function express a property of having a maximum where $\mathbf{x}^{(i)} = \boldsymbol{\mu}^k$ and decreasing as they get less similar.

5.1.3 Results

One issue that might occur is a bad estimation of the background or cell body as some cells takes a very small portion of an image or, on the other size, takes almost a whole image when segmented as shown in image X. To see how this influences the problem, histograms are approximated with Gaussian functions in two ways. First, the histogram is as in Chapter 4, without any restrictions. Second, the background and cells part of an image are separated and each part is approximated with a Gaussian individually.

CHAPTER 6

Describing cells

6.1 Interesting features

6.2 Deep learning

6.3 Deep belief networks

6.4 Evaluation

CHAPTER 7

Rule mining

7.1 Inductive logic programming

7.2 Aleph

7.3 FOIL

Appendices

Bibliography

[1] D. Adams. *The Hitchhiker's Guide to the Galaxy*. Del Rey (reprint), 1995. ISBN-13: 978-0345391803.

[2] P. Agrawal, M. Vatsa, and R. Singh. Hep-2 cell image classification: A comparative analysis. In G. Wu, D. Zhang, D. Shen, P. Yan, K. Suzuki, and F. Wang, editors, *Machine Learning in Medical Imaging*, volume 8184 of *Lecture Notes in Computer Science*, pages 195–202. Springer International Publishing, 2013.

[3] P. Foggia, G. Percannella, P. Soda, and M. Vento. Benchmarking hep-2 cells classification methods. *IEEE Trans. Med. Imaging*, 32:1878–1889, 2013.

[4] Y.-L. Huang, C.-W. Chung, T.-Y. Hsieh, and Y.-L. Jao. Outline detection for the hep-2 cell in indirect immunofluorescence images using watershed segmentation. In *Sensor Networks, Ubiquitous and Trustworthy Computing, 2008. SUTC '08. IEEE International Conference on*, pages 423–427, 2008.

[5] Y.-L. Huang, Y.-L. Jao, T.-Y. Hsieh, and C.-W. Chung. Adaptive automatic segmentation of hep-2 cells in indirect immunofluorescence images. In *Proceedings of the 2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (Sutc 2008)*, SUTC '08, pages 418–422. IEEE Computer Society, 2008.

[6] D. C. Ian D Odell. Immunofluorescence techniques, 2013.

[7] K. Li, J. Yin, Z. Lu, X. Kong, R. Zhang, and W. Liu. Multiclass boosting svm using different texture features in hep-2 cell staining pattern classification. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 170–173, 2012.

[8] R. Nakamura. *Quality Assurance for the Indirect Immunofluorescence Test for Autoantibodies to Nuclear Antigen (IF-ANA): Approved Guideline (1996)*. NCCLS document. NCCLC, 1996.

[9] R. Nosaka, Y. Ohkawa, and K. Fukui. Feature extraction based on co-occurrence of adjacent local binary patterns. In *Proceedings of the 5th Pacific Rim Conference on Advances in Image and Video Technology - Volume Part II*, PSIVT'11, pages 82–91. Springer-Verlag, 2012.

[10] R. Nosaka, C. H. Suryanto, and K. Fukui. Rotation invariant co-occurrence among adjacent lbps. In *Proceedings of the 11th International Conference on Computer Vision - Volume Part I*, ACCV'12, pages 15–25. Springer-Verlag, 2013.

[11] P. Perner, H. Perner, and B. Mueller. Mining knowledge for hep-2 cell image classification. *Artificial Intelligence in Medicine*, 26:161–173, 2002.

[12] A. Rigon, P. Soda, D. Zennaro, G. Iannello, and A. Afeltra. Indirect immunofluorescence in autoimmune diseases: Assessment of digital images for diagnostic purpose. *Cytometry Part B-clinical Cytometry*, 72B:472–477, 2007.

[13] P. Soda and G. Iannello. A multi-expert system to classify fluorescent intensity in antinuclear autoantibodies testing. In *Computer-Based Medical Systems, 2006. CBMS 2006. 19th IEEE International Symposium on*, pages 219–224, 2006.

[14] P. Soda, G. Iannello, and M. Vento. A multiple expert system for classifying fluorescent intensity in antinuclear autoantibodies analysis. *Pattern Anal. Appl.*, 12(3):215–226, Sept. 2009.

[15] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *Int. J. Comput. Vision*, 62(1-2):61–81, Apr. 2005.