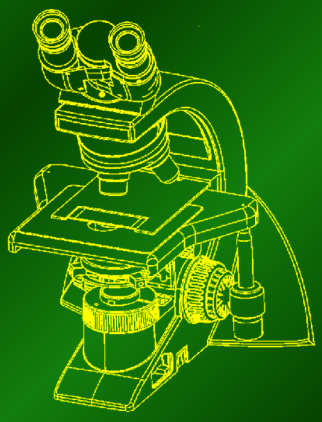
Image analysis for gene expression based phenotype characterization in yeast cells

Tleis, M.

Image Analysis for Gene Expression based Phenotype Characterization in Yeast Cells



Mohamed Tleis

Image Analysis for Gene Expression based Phenotype Characterization in Yeast Cells

Mohamed Tleis

July 2016

ISBN: 978-94-028-0236-8

Image on the back cover is from a GFP yeast cell-line expressing Bmh1 made with Zeiss Confocal Microscope.

This research was partly supported by:

- *Erasmus Mundus JoSyLeEN program*
- *Raymond-Sackler* organisation.
- *Landelijke Stitching voor Blinden en Slechtzienden (LSBS)* organisation.

Typeset by LaTeX printed by Ipskamp Drukkers.

Image Analysis for Gene Expression based Phenotype Characterization in Yeast Cells

Proefschrift

te verkrijging van

de graad van Doctor aan de Universiteit Leiden,

op graag van Rector Magnificus Prof. Mr. C.J.J.M Stolker

volgens besluit van het College voor Promoties

te verdedigen op woensdag 6 juli 2016

klokke 10.00 uur

door

Mohamed Tleis

geboren te Brital, Libanon, in 1982

Promotiecommissie

Promotoren:	Prof. dr. T.H.W. Bäck	
	Prof. dr. A. Plaat	
Co-Promotor:	Dr. ir. F.J. Verbeek	
Overige leden:	Prof. dr. H. Blockeel	Katholieke Universiteit Leuven, Belgium
	Prof. dr. T. Gevers	University of Amsterdam
	Dr. G.P.H van Heusden	

Dedicated to humanity ...

Contents

List of Figures

List of Tables

1

Introduction

“This chapter presents the reader with our research objective, the problem that we address and the goals of this research. In addition, it offers a basic background and definitions necessary to follow up in this thesis. Specifically, it introduces the necessary background about cytomic studies and pattern recognition methods followed within our research. The final section illustrates the scope or structure of the thesis.”

1.1 Research Objective

THIS thesis addresses the main research problem on how pattern recognition systems can support objective analysis and phenotype characterization of single-cell in image-based gene expression experiments. Hence, we address the crucial research questions directly connected to this problem. Our starting point is spherical/ovoid shape cells. First we address the components and processes required to build a comprehensive image analysis pipeline for single-cell image based gene expression. Second we address the best approach to segment ovoid-shaped cells that are common in micro/cell biology such as *Saccharomyces cerevisiae*. In addition, we address how a machine learning approach can aid in the object recognition process, and how it can improve the identification of subtle patterns residing within the measurement data? i.e. recognizing the patterns that are not obvious and hard to notice within standard measurement methods. Another directly related research question is about the features used by the machine learning approach, where we address the extraction of relevant and meaningful feature sets. Finally, we address the question on whether the recognition system can be validated on a yeast study experiment to discriminate various cell groups.

1.2 Cytomics and *Saccharomyces cerevisiae*

Cytomics is the study of cell systems (cytomes) at a single cell level. It combines all the bioinformatics knowledge to attempt to understand the molecular architecture and functionality of the cell system. Much of this is achieved by using molecular and microscopic techniques that allow the various components of a cell to be visualized as they interact in vivo [Bra11]. In this section, we define and discuss the gene expression and measurement followed within cytomics. Subsequently, we define and discuss fluorescent proteins used to study cell behaviours. Finally we present and discuss the eukaryote model organism used commonly in cell/micro biological studies, and which we take as a case study in our research, i.e. the *Saccharomyces cerevisiae* yeast cells.

1.2.1 Gene Expression and Measurement

In genetics, gene expression is the most fundamental level at which the genotype gives rise to the phenotype, i.e. observable trait. The genetic code stored in DNA is "interpreted" by gene expression, and the properties of the expression give rise to the organism's phenotype. Such phenotypes are often expressed by the synthesis of proteins that control the organism's shape, or that act as enzymes catalysing specific metabolic pathways characterising the organism. Protein-coding genes are transcribed into messenger RNA (mRNA), which is an information carrier coding for the synthesis of one or more proteins.

Measuring gene expression is an important part of many life sciences. The ability to quantify the level at which a particular gene is expressed within a cell or organism can provide a huge amount of information. Similarly, the analysis of the location of expression protein is a powerful

tool and this can be done on an organism or cellular scale. Investigation of localisation is particularly important for study of development in multicellular organisms and as an indicator of protein function in single cells. Ideally, measurement of expression is done by detecting the final gene product (for many genes this is the protein); however, it is often easier to detect one of the precursors, typically mRNA, and infer gene expression level. Levels of *mRNA* can be quantitatively measured by northern blotting, which provides size and sequence information about the *mRNA* molecules.

For expression profiling or high-throughput analysis of many genes within a sample, quantitative *PCR* may be performed for hundreds of genes simultaneously in the case of low-density arrays. A second approach is the hybridization microarray, which is a popular approach in gene expression studies. Microarrays reveal expression profiles for a large number of genes at different time points. A single array or "chip" may contain probes to determine transcript levels for every known gene in the genome of one or more organisms [Wel07]. Alternatively, "tag based" technologies can be used, such as Serial analysis of gene expression (*SAGE*), which can provide a relative measure of the cellular concentration of different mRNAs. Next-generation sequencing (*NGS*) such as *RNA-Seq* is another approach, producing vast quantities of sequence data that can be matched to a reference genome. Although *NGS* is comparatively expensive, and resource-intensive, it can identify single-nucleotide polymorphisms, splice-variants, and novel genes, and can also be used to profile expression in organisms for which little or no sequence information is available.

For genes encoding proteins, the expression level can be directly assessed by a number of means with some clear analogies to the techniques for *mRNA* quantification. The most commonly used method is to perform a Western blot against the protein of interest. The gel-based nature of this method makes quantification less accurate although it has the advantage of being able to identify later modifications to the protein [Nei00, Ama08]. Moreover, Mass spectroscopy is developing fast and allows the quantification of a large part of the proteome, which directly addresses the level of gene products present in a given cell state and can further characterize protein activities, interactions and subcellular distributions [Ong05]. Mass spectrometry (MS) is an analytical technique that ionizes chemical species and sorts the ions based on their mass to charge ratio. In simpler terms, a mass spectrum measures the masses within a sample.

Analysis of expression is not limited to only quantification; localisation can also be determined. *mRNA* can be detected with a suitably labelled complementary *mRNA* strand and protein can be detected via labelled antibodies. The probed sample is then observed by microscopy to identify where the *mRNA* or protein is.

By tagging the gene with a reporter gene, i.e. by replacing the gene with a new version fused to the reporter gene expressing fluorescent proteins as markers, expression may be directly quantified in live cells. It is very difficult to clone a reporter gene into its native location in the genome without affecting expression levels so this method often cannot be used to measure endogenous gene expression. It is, however, widely used to measure the expression of a gene artificially introduced into the cell; for example, via an expression vector. It is important to note that, in some cases, by fusing a gene to a fluorescent reporter the expressed protein's behaviour, including its cellular localization and expression level might change. The analysis

of reporters is initiated by imaging using a confocal laser scanning microscope and by flow cytometry, discussed hereafter.

1.2.2 Fluorescent Microscopy and Flow Cytometry

There are two popular cell analysis techniques including fluorescent microscopy and flow cytometry, these are discussed and subsequently their complementarity is considered.

Microscopy

Proteins that are tagged with fluorescent molecules can be studied through imaging. The most common techniques is by using fluorescent microscopy. Laser scanning microscopy in general is preferred for fluorescence imaging because of their resolution, higher contrast, ability to reconstruct 3-D images, the absence of artefacts induced by conventional microscopy, and most importantly its ability to penetrate into the specimen and obtaining an image of a specific focal plane [Mas01]. The multi-photon photo-luminescence microscopes have high spatial resolution and reduced background [Zho10] and also permit additional structures to be observed [Mas01]. However, multi-photon microscopes do not contain pinhole apertures, which give confocal microscopes their optical sectioning quality [Kam13]. In addition, modern confocal laser scanning microscope *CLSM* can do fast scanning that prevents photo-toxicity due to thermal damage [Paw10], as well as minimizing photo-bleaching. In confocal microscopy, the focus plane of illumination is the same as the focal plane of detection. In other words, the focus plane of illumination and the focal plane of detection are confocal [Row00].

Flow Cytometry

In cell biology, flow cytometry is a laser-based, biophysical technology employed in cell counting, cell sorting, biomarker detection and protein engineering. It suspends cells in a stream of fluid and passes them by an electronic detector. Flow cytometry allows simultaneous multi-parametric analysis of the physical and chemical characteristics of up-to thousands of particles per second [Yan15]. A flow cytometer is similar to a microscope; however, it doesn't produce an image of the cell but offers high-throughput automated quantification of the set parameters for a high number of single cells during each analysis session [Tho06].

Complementarity of Flow Cytometry and Fluorescence Microscopy

Flow cytometry and fluorescence microscopy both provide single-cell analysis using different but complementary sets of data, essentially population-based target intensities versus target morphology in relatively small sample sizes. Both approaches employ optical filters to analyze fluorescence emissions and have to overcome some of the same physical limitations including spectral overlap of dyes and the dynamic range limits of measuring systems. Hence, flow cytometry and confocal fluorescence microscopy technologies both have specific characteristics and limitations. In microscopy, photostability is a more critical issue. Flow cytometry is

limited by its requirement that analyzed cells are in suspension, making information on tissue architecture and cell-cell interactions inapplicable. On the other side, fluorescence microscopy is well suited to the resolution of cell and tissue architecture, and to following kinetic and trophic responses in single cells. In flow cytometry, cell subpopulations with similar marker expression are difficult to differentiate, and analyses that employ more fluorophores are subject to signal spillover [Jah12]. In addition, there is the inability of flow cytometry to recognize morphologically analyzed cells [Mur08]. Flow cytometry rapidly quantifies small differences between cell populations using statistically significant numbers of events. Flow cytometry can represent a “black box” when looking at the magnitude of a population response; fluorescence microscopy can help verify that measured results represent meaningful biological effects.

In general, the microscopist may arrive at quantitative data. However, the cooperative use of both flow cytometry and microscopy can provide more robust numerical description of biological phenomena [God05].

1.2.3 Cytomics and Fluorescent Proteins

Proteins are vital parts in living organisms. Many important proteins in human biology were understood by studying their homologs in yeast; such proteins include cell cycle proteins, signalling proteins, and protein-processing enzymes [Wal04]. The large-scale study of the structures and functions of such proteins is called Proteomics [Bla99]. In Proteomics, Fluorescent proteins such as green fluorescent protein (*GFP*) and its derived variants are widely used. One of the most exciting applications is the generation of a library of *S. cerevisiae* strains in which each coding sequence is tagged with green fluorescent protein (*GFP*) [Huh03]. This library enables the determination of the localization of more than 70 percent of the *S. cerevisiae* proteins. In addition, levels of these proteins can be quantified after cultivation under different conditions.

Tagging genes with the reporter expressing green fluorescent protein (*GFP*) is a highly specific and sensitive technique for studying the inter-cellular dynamics of proteins and organelles [Sha97]. *GFP* expression is an excellent marker to monitor the gene expression [Phi01] and protein localization in the living yeast cells. The biggest advantage of the intracellular *GFP* is that it is heritable, since it can be transformed with the use of DNA-encoding *GFP*. Additionally, visualizing *GFP* is non-invasive as it is detectable by just shining light on it. Furthermore, it is a relatively small and inert molecule that does not appear to interfere with cell growth and function. Moreover, if *GFP* is used with a monomer it can diffuse readily throughout cells [Cha09].

The green fluorescent protein (GFP) was first isolated from the jellyfish *Aequorea victoria* [Sha97, Phi01]. It is a protein composed of 238 amino acid residues (26.9 kDa) that exhibits bright green fluorescent when exposed to light in the blue to ultraviolet range [Wal04, Moy08, Cha94]. It has a beta-barrel structure consisting of eleven β-strands. The beta barrel structure is a nearly perfect cylinder, 42 Å long and 24 Å in diameter, creating what is referred to as a "β-can" formation, which is unique to the GFP-like family [Yan97]. In GFP the fluorophore is formed inside the protein globule by modification of amino acids [Shi79].

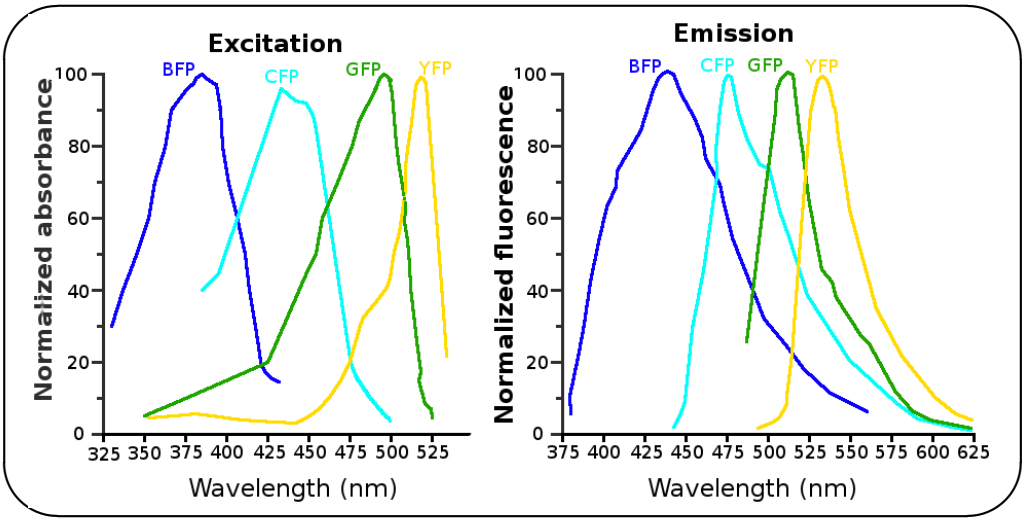


Figure 1.1: *Excitation and Emission of GFP and some variants. Data from [Hei96].*

The *GFP* gene is widely used as a reporter gene and can be introduced into organisms and maintained in their genome through breeding, injection with a viral vector, or cell transfection. The *GFP* gene has been introduced and expressed in the *S. cerevisiae* yeast cells as well as other types of yeast cells, bacteria, fungi, fish (such as zebrafish), plant, fly, and mammalian cells, including human [Cha09].

GFP requires low excitation light intensity to prevent photo-bleaching and photo-toxicity at a light wavelength of 490 nm or less [Sha97]. The *GFP* from *A. victoria* has a major excitation peak at a wavelength of 395 nm and a minor one at 475 nm. Its emission peak is at 509 nm, which is in the lower green portion of the visible spectrum [Phi01]. This fluorescence is very stable, and virtually no photo-bleaching is observed [Cha94].

Many different mutants (variants) of the *GFP* have been engineered, with improved spectral characteristics of *GFP*, resulting in increased fluorescence, photo-stability, and a shift of the major excitation peak to 488 nm. *EGFP* and *Superfolder GFP* are examples of such variants. Many other mutations have been made as well including color mutants; in particular, blue fluorescent protein (*EBFP, EBFP2, Azurite, mKalama1*), cyan fluorescent protein (*ECFP, Cerulean, CyPet, mTurquoise2*), and yellow fluorescent protein derivatives (*YFP, Citrine, Venus, YPet*). They exhibit a broad absorption band in the ultraviolet spectrum (cf. Fig. 1.1).

Knowing how much of a protein is expressed is not sufficient to understanding its behaviour. It is particularly important to also know its subcellular location because changes in protein subcellular location can cause dramatic effects on cell behaviour. Changes in location within a cell type may also cause or result from disease [Mur05]; however, subcellular location has received less attention than many other aspects of gene and protein behaviour. The major exception is in yeast, in which almost all proteins have been assigned to a set of major subcellular structures using fusion of DNA, with the coding sequence of fluorescent proteins such as the

green fluorescent protein. For example, Huh et al [Huh03] used green fluorescent protein tagging of DNAs and visual examination to assign proteins to 12 categories: cell periphery, bud, bud neck, cytoskeleton, microtubule, cytoplasm, nucleus, mitochondrion, endoplasmic reticulum, vacuole, vacuolar membrane, and punctate. They then used colocalization with red fluorescent protein markers to divide the cytoskeleton class into two classes, actin cytoskeleton and spindle pole, and to add nine new categories: nucleolus, nuclear periphery, golgi apparatus, three types of transport vesicles, endosome, peroxisome, and lipid particle. In all, 4,156 proteins were assigned to these 22 categories in their study [Huh03, Mur05].

In our experiments on *S. cerevisiae*, the model organism is tagged with fluorescent proteins and is visualized by confocal laser scanning microscope. The subsequent sub-section introduces this organism that is used to understand gene expression and genetic networks in cytomics.

1.2.4 Saccharomyces cerevisiae in cytomics

Saccharomyces derives from Latinized Greek and means "sugar-mold" or "sugar-fungus", *saccharo* being the combining form "sugar" and *myces* being "fungus". *Cerevisiae* comes from Latin and means "of beer". This organism is also known as Baker's yeast, Brewer's yeast, Ale yeast, Top-fermenting yeast and Budding yeast [Stă13].

S. cerevisiae is a well-known yeast species and used since ancient times in wine-making, brewing and baking. It was originally isolated from the skin of grapes and has been one of the most intensively studied eukaryote model organisms in molecular and cell biology. [Fel10]. *S. cerevisiae* cells are round to ovoid with a diameter between 2 to 10 micrometers. [Par97].

Many cell processes in the yeast model eukaryote cell are similar to that in plants and mammalians including humans. This fact makes yeast an excellent model organism to understand the behaviour of proteins involved in such processes. Several traits in the *S. cerevisiae* drive researchers to look for this organism. Among these traits is its size, generation time, accessibility, manipulation, genetics, conservation of mechanisms, potential economic benefits [Gov11], cell's transparency, the fact that its genome sequence was completed in 1996, and the availability of a library of strains in which each individual coding sequence is tagged with green fluorescent protein (GFP) [Huh03] in addition to a library of knock-out strains [Win99] and a library of TAP-tagged strains [Gha03]. These traits make *S. cerevisiae* a significant tool in biological research. Studying DNA damage and repair mechanisms is one example [Nic01].

S. cerevisiae yeast cells can survive and grow in two forms; as haploid or diploid cells where they undergo a simple life-cycle of mitosis and growth. Haploid cells usually die under stress conditions, while diploid can undergo sporulation, entering meiosis and producing four haploid spores, which can proceed to mate. This reproduction process is known as budding and there where Budding yeast get their name from. With adequate nutrients, yeast cells can double in numbers within 100 minutes [Her88]. The mean replicative life span of the *S. cerevisiae* is about 26 cell divisions [Kae05, Kae10].

In molecular genetics, *S. cerevisiae* is used as a model system in the understanding of gene expression and genetic networks, because it combines considerable variation in key cell characteristics such as protein levels and expression, cell size, shape and age. It also has short

generation time and immobility [Don13b]. Moreover, it can be manipulated and genetically engineered.

Genetic and hereditary diseases are still incurable because we lack the understanding behaviours of many proteins. This fact drives research groups to use the *S. cerevisiae* yeast cell to understand the behaviour of proteins responsible for many processes in the cells. Such as the 14-3-3 family of proteins which is considered vital to cell life. In order to prepare such cells for the experiments, the gene expressing a protein under study is attached (tagged) to a report gene that expresses fluorescent protein such as the popular green fluorescent protein (*GFP*) or any of its variants. How we use this model organism to recognize patterns is the topic of next section.

1.3 Pattern Recognition

Pattern recognition aims to classify data (patterns) based on either a priori knowledge or on statistical information extracted from the patterns. The patterns to be classified are usually groups of measurements or observations, defining points in an appropriate multidimensional space. A complete pattern recognition system consists of: (i) a sensor that gathers the observations to be classified or described, (ii) a feature extraction mechanism that computes numeric or symbolic information from the observations and (iii) a classification or description scheme that does the actual job of classifying or describing observations relying on the extracted features [dic16].

In our research, the sensor used is the fluorescence microscope that gathers the observations, i.e. the yeast cell images. Such image observations are acquired by *CLSM* microscopy and are discussed in the first sub-section. From such observations image analysis techniques are applied to extract the features. Such techniques involve image processing, image segmentation, and object measurement techniques; which will be the topic of the second sub-section. The last part of the pattern recognition discussion is dedicated to the data analysis and classification of the measured features obtained by image analysis.

1.3.1 Image Acquisition

All the images created in this work were analyzed by confocal laser scanning microscopy (CLSM) to view fluorescent tagged proteins expressed within the cells. Images are acquired as two or three channel images. One or two channels of the reporter construct (*GFP, YFP, CFP*). The *GFP* is excited at 488nm and *YFP* at 514nm, and emission is at 500-550 for *GFP* or 530-600nm for *YFP* [Zah12]. In addition, a bright-field channel is acquired; the bright-field image depicts the structure of the cells.

The Bright-field image channel is acquired through a bright-field technique embedded within the confocal microscope. Since it depicts the yeast structure, this channel of the microscope images of yeast *S. cerevisiae* cells is used primarily, in our research, to detect the cell contours. For optimal detection, optimal microscope settings have to be set and hence the parameters to be used have to be determined. To determine these parameters, an experiment was done where several images were generated under different microscope settings.

Table 1.1 lists all the different settings used, where each setting is labelled with a different letter (A...E). This label represents a different category of images. To choose the optimal image category that works best with segmentation algorithms. We applied a segmentation algorithm to try the detection of cell objects in these images. A score of true positive cell detections was computed for each category.

Image Label	Resolution	Master Gain	Digital Offset	Laser
A	512x512	320	-1.20	18%
B	1024x1024	320	-1.20	18%
C	1024x1024	340	-1.90	20%
D	1024x1024	261	-0.05	18%
E	1024x1024	301	-0.50	18%

Table 1.1: *microscope settings*

The result in Table 1.2 shows different labelled images with their acquired scores. Each image label corresponds to images acquired with the microscope settings listed in Table 1.1. The score of true positive rate indicates the number of cells correctly detected. The highest score corresponds to category labelled as B, i.e. the microscope settings of 1024x1024 resolution, 320 master gain, -1.20 digital offset and 18% laser power.

Image Label	True Positive
A	67%
B	70%
C	52.6%
D	48.4%
E	47.6%

Table 1.2: *Percentage of True Positives*

The availability of *GFP* and its derivatives has thoroughly redefined fluorescence microscopy and the way it is used in cell biology and other biological disciplines [Yus05]. Such Fluorescent (photon-emitting) molecules are introduced into yeast cells because they have a helpful property of fluorescing when in the presence of non-fluorescent molecules or structures under study. When the *S. cerevisiae* specimen is illuminated using laser techniques in confocal microscopy, the fluorophores (fluorescent molecules) absorbs the light photons raising them to an excited state with a wavelength (energy) specific to the fluorophore itself. The fluorophore then returns to its ground state and may emit a photon with lower energy (longer wavelength). This photon might then strike the detector with the proportion of light entering the objective lens of the microscope. The charges of electrons produced by photons striking the detector are quantified and from these quantifications, pixel values are determined. These pixel values correspond to the number of detected photons (cf. Fig. 1.2). From the number of photons detected at each pixel, interpretations can be made about the presence or absence of some feature, the size and shape of a structure, or about the relative concentration of a molecule [Ban13]. The excitation and emission wavelength used to detect the fluorescent molecules varies with different types. For example the *GFP* is imaged with excitation at 488 nm and emission at 505-530 nm.

Now that the image acquisition is performed, we shift our discussion to the image analysis phase.

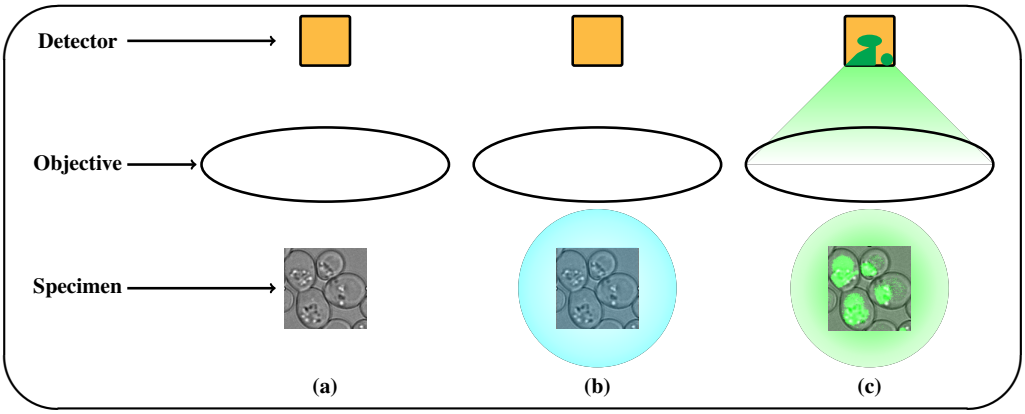


Figure 1.2: *Schematic Image Formation of Fluorescence Microscopy. (a) - basic setup. (b) - the specimen is illuminated in higher energy light, fluorophores become excited. (c) - Lower energy Light is emitted; some enters the objective lens and is detected.*

1.3.2 Image Analysis

When digital cameras where introduced to microscopy, digital image analysis has developed as an established complement, allowing routine quantification of microscope observations [Tho96]. Currently, flow cytometry is one of the methods used to measure levels of fluorescent proteins; this can, however, not provide the quantification that image analysis can. Hence, the use of image analysis was probed to accomplish further progress. Digital image analysis, widely known as image analysis, is when a machine automatically studies an image to obtain useful information from it. The applications of digital image analysis are continuously expanding through all areas of science and industry. It has been successfully applied in a wide variety of fields ranging from astronomical observations to cell analysis. To name few other fields: nuclear medicine, medical diagnostics, industry, lithography, microscopy, lasers, biological imaging, remote sensing, law enforcement, radar images, geological exploration [Gon08], sports management [Fer99] and chemical imaging [Sch95]. There are many different techniques used in automatically analyzing images. Each technique may be useful for a small range of tasks. However, there still are not any known methods of image analysis that are generic enough for wide ranges of tasks, compared to the abilities of a human's image analyzing capabilities [Sol11]. In our domain, we considered few of these techniques including image processing, image segmentation and feature extraction discussed hereafter.

Image Processing

Image processing is usually used to refer to digital image processing. It is a process that accepts an image as an input, and the output may be either image or a set of characteristics or parameters related to the image. It is the use of computer algorithms to perform image processing on digital images. Since images are defined over two dimensions (perhaps more),

image processing may be modelled in the form of multidimensional systems. Image processing is the only practical technology for classification, feature extraction, multi-scale image analysis and pattern recognition [Gon08].

Image Segmentation

Crucial to image analysis is image segmentation, which is defined as subdividing an image into its constituent regions or objects [Gon08]. The result of image segmentation is objects representing connected regions of similar intensity. From these regions, measurements can be conducted. Some practical applications of image segmentation are image processing, computer vision, face recognition, medical imaging, digital libraries, image and video retrieval, and cell image analysis to measure gene expression [Tle13, van07].

Segmentation of individual cells relies on the ability to detect cell boundaries and classifying all pixels in a given image as foreground or background. The differentiation between foreground and background pixels can be accomplished by a threshold function determined by a simple intensity based method, or by more complex functions such as graphical models, pattern recognition, deformable templates, cell contours or the watershed algorithm [Don13b].

There are a number of refining segmentation algorithms, tracking algorithms, morphology characterization, and protein localization. However, we lack a robust approach for the segmentation and tracking of budding yeast [Don13b]. The result of such a robust segmentation algorithm enables us to extract binary masks and contours of cells. These obtained masks and contours allow us to measure various features of those objects, i.e. cells. More details on object measurement follow hereafter.

Measurement and Features

Herein, we will define the concepts of measurement, features, textures and feature extraction.

Measurement

In its classical definition throughout physical sciences, measurement is the determination or estimation of ratios of quantities [Mic99]. However, information theory recognises that all data are inexact and statistical in nature. Thus the definition of measurement in information theory is "a set of observations that reduce uncertainty where the result is expressed as a quantity" [Hub07]. In general, we can state that measurement is the assignment of a number to a characteristic of an object or event, which can be compared with other objects or events [Ped91]. The science of measurement is pursued in the field of metrology, which includes all theoretical and practical aspects of measurement [BIP08].

Measurement is a cornerstone in Bio-Imaging and pattern recognition as well as in most science fields. It is an important step in the discovery process. In Biological image analysis, measurement is strongly related to features and textures extracted from images. Hereafter, we will define what a feature is? what a texture is? and we will highlight on the well-known feature extraction techniques followed within bio-imaging.

Features

In computer vision and image processing, a feature is a piece of information which is relevant for solving the computational task related to a certain application. This is the same sense as feature in machine learning and pattern recognition generally, though image processing has a very sophisticated collection of features. Features may be specific structures in the image such as points, edges or objects. Features may also be the result of a general neighborhood operation or feature detection applied to the image. In general definition, an image feature is a representation or an attribute of an image describing certain special characteristics of the pattern of interest.

In our application it is not sufficient to extract only one type of feature to obtain the relevant information from the image data. Instead multiple different features are extracted, resulting in multiple feature descriptors at each image point. The information provided by all these descriptors is organized as the elements of one single vector, referred to as a feature vector. The set of all possible feature vectors constitutes a feature space. A common example of feature vectors appears when each image point is to be classified as belonging to a specific class. Assuming that each image point has a corresponding feature vector based on a suitable set of features, meaning that each class is well separated in the corresponding feature space, the classification of each image point can be done using standard classification methods [wik15]. Chapter 4 applies such classification on our feature space.

Textures

An image texture is a set of metrics calculated in image processing designed to quantify the perceived texture of an image. Image texture gives us information about the spatial arrangement of color or intensities in an image or selected region of an image [Sha01]. In other words, it is defined as the visual effect which is produced by spatial distribution of total variations over relatively small areas [Bar95]. Image textures are believed to be a rich source of visual information. They are complex visual patterns composed of entities, or sub-patterns, that have characteristic brightness, colour, slope, size, etc... Thus a texture can be regarded as a similarity grouping in an image. Image textures can be artificially created or found in natural scenes captured in an image. Image textures are one way that can be used to help in segmentation or classification of images. To analyze an image texture in computer graphics, there are two ways to approach the issue: Structured Approach and Statistical Approach. A structured approach sees an image texture as a set of primitive texels in some regular or repeated pattern. This works well when analyzing artificial textures. However, since natural textures are made of patterns of irregular sub-elements as is the case in our application, statistical approach is used. In general, statistical approach is easier to compute and is more widely used. It sees as image texture as a quantitative measure of the arrangement of intensities in a region.

Feature extraction

Feature extraction is defined as locating those pixels in an image that have some distinctive characteristics [Gub09]. The most known feature extraction techniques in image analysis are classified into first-order histogram based features, invariant moment features, co-occurrence matrix based features, and multi-scale features [Mat98]. More details about these techniques will be discussed in Chapter 4.

1.3.3 Data Analysis

After performing the measurement, data analysis is necessary to make the measurement meaningful for users. Data analysis can be defined as the process for obtaining raw data and converting it into information useful for decision making and suggesting conclusions [Jud11]. The workflow depicted in Fig. 1.3 illustrates the general process followed within bioinformatics. Initially images are acquired from the imaging device during the image acquisition phase; then these images undergo segmentation to locate the individual objects. Following the segmentation is the measurement phase where objects are measured for various features. The obtained data must be processed or organized for analysis [Sch13], but the data could be incomplete, contain duplicates, or contain errors. Such issues are presented and corrected through data cleaning [Ara15]. Once the data is processed and cleaned, it can be analyzed. Analysts apply a variety of techniques referred to as exploratory data analysis to begin understanding the messages contained in the data [Few04]. Descriptive statistics are generated to help understand the data. These data are examined in graphical format using data visualization techniques to communicate key messages contained within the data through these graphical means and charts [Fri08]. Machine learning models and algorithms can be applied to identify relationships among the variables or to classify the instances of the data based on these variables. Data mining and machine learning plays a major role here. They focus on modelling and knowledge discovery for predictive rather than purely descriptive purposes. The output of such models is data products fed back to the user such as conclusions or identified subtle patterns residing within the data.

1.4 Thesis Structure

In the remaining chapters we explain the research we have set out to perform.

Chapter Two: Image Analysis Platform. This chapter presents a complete framework for biological experiments. It demonstrates how an automated platform based on a complete image analysis pipeline assist biologists in their experiments? Moreover, this chapter discusses the individual modules that form the complete framework, including the segmentation, measurement, data analysis and the GUI that combines these modules together.

Chapter Three: Hough Transform Based Contour Extraction and Optimization. In this chapter, we show how a new approach based on Hough transform and minimal path algorithms can improve the segmentation of ovoid objects, i.e. yeast cells. We start by defining Hough transform and minimal path algorithms. Subsequently we present our general approach to detect ovoid objects in microscope images by detecting circular arcs using a variety of the Hough transform. In addition, we discuss the application of minimal path algorithms to extract the exact contour of detected objects from a polar representation of the image surrounding the object. Furthermore, this chapter presents an additional novel algorithm to expand the extracted contours of ovoid objects. Such expansion is necessary for some settings due to the inherently fuzzy nature of edges and delicate microscope settings. This chapter explains how the polar

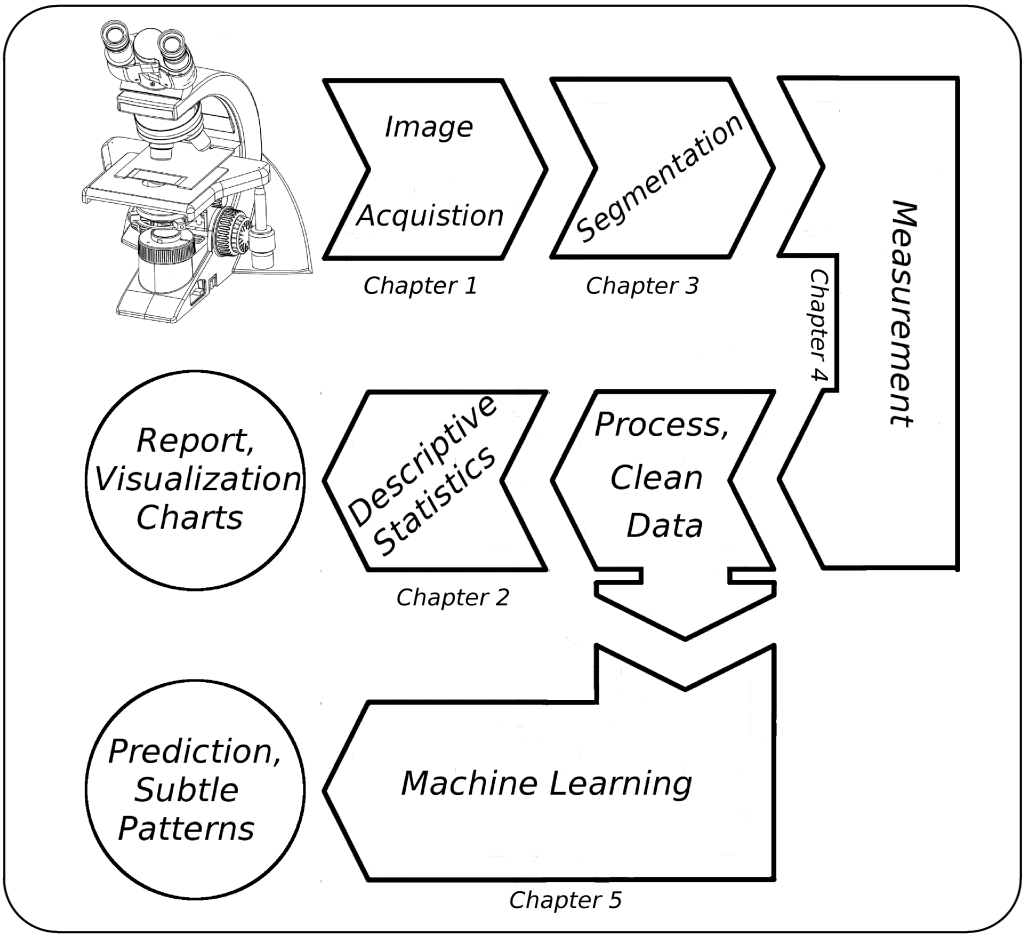


Figure 1.3: *Flow Chart of the data analysis process*

representation of images is used to expand the initially detected contours by applying circular shortest paths. In addition, it explains the three introduced parameters to control the expansion process. These parameters are *resistance, limit* and *convergence*. The expansion of a sample contour is demonstrated as well in this chapter. Finally, results and comparison with other methods are evaluated using a dataset of *S. cerevisiae* yeast cells.

Chapter Four: Machine Learning to Improve Object Recognition and Discrimination of Cell Groups Using Sophisticated Features. This chapter specifically address machine learning where we introduce features to be used in a machine learning approach to automatically identify cell groups cultivated in two different media. We use the same approach to classify cell objects from artefacts. First we discuss the feature extraction techniques including first-order histogram features, texture measurement, moment invariants, co-occurrence matrix based features and multi-scale wavelet-based texture measurement. Subsequently, various

classification methods are evaluated to build a model imported into the yeast analysis platform to be trained for the automatic discrimination of cell groups. This discrimination helps showing that there are different gene expression patterns between cells cultivated under different stress levels. Moreover, the same classification methods are evaluated to build another model for the identification of cell objects. This model is used to discriminate the segmented objects in images into intact cell objects or artefacts such as debris and dead cells.

Chapter Five: The Effect of NaCl on 14-3-3 Proteins and Nha1 antiporter. In this chapter, the designed image analysis platform is used in a case study to determine the effect of sodium chloride on 14-3-3 genes including Bmh1 and Bmh2 in addition to the *NHA1* encoding an antiporter. The study also includes a mutant of *BMH1 (Δbmh1)* to study the expression of Nha1 under different osmotic stress levels. The result obtained from using the yeast analysis software is also validated with that obtained from flow cytometry.

Chapter Six: Discussion. In this chapter, we draw out conclusions learned from this dissertation and give scope for further research and applications.

2

Yeast Analysis Platform

“This chapter describes the yeast analysis framework we have developed to support research in yeast biology. We start by presenting the workflow of the system and elaborate on its various modules including segmentation, measurement, analysis and visualization, and finally the Graphical User Interface (GUI) that connects all these modules. A validation small scale experiment is performed as well.”

This chapter is based on the following publications:

- Mohamed Tleis, Ginny Anemaet, Paul van Heusden, and Fons J. Verbeek. Image analysis platform for yeast biologists. In The 2nd International Conference on Advances in Biomedical Engineering 2013 (ICABME'13), pages 57–60, Tripoli, Lebanon, September 2013.
- Mohamed Tleis and Fons J. Verbeek. Yeast-Cell features extraction plugin. In ImageJ Conference 2012, Mondorf, Luxembourg, October 2012.

2.1 Background

THE goal of this study is to address which components and processes can form an objective and comprehensive image analysis pipeline to analyze image-based gene expression in single cells. The developed objective image analysis platform should be applicable for live cell imaging, thus without additional staining and should perform in an automatic way the entire image analysis pipeline:

- Fluorescence (confocal) microscopy images.
- Detection of individual cells on the images (segmentation).
- Quantification of the fluorescent signal and other characteristics of each individual cell (measurement).
- Statistical analysis of the measurements.
- Visualization of the measurements in several graphical ways.

To quantify expression levels of GFP-tagged reporter genes in fluorescent microscope images, several software tools have been developed. The first step is to detect the individual cells in the image through segmentation. Detection of cells can be performed both in the bright-field channel as well as in the fluorescence channel(s). Until recent, a software tool has been used for screening the yeast GFP-fusion library to investigate global cellular protein reorganization on exposure to the alkylating agent methyl-methane-sulfonate [Maz13]. In that image analysis software, developed in Matlab, cellular boundaries are detected after staining the cell-wall with Alexa 647 conjugated Concanavalin A. A disadvantage of such an approach is that additional staining is required. Another recent research implements a pipeline including high-throughput microscopy, automated image analysis, and pattern classification through machine learning [Cho15]. The approach followed in that research also requires staining. It uses yeast synthetic genetic array (SGA) technology to introduce a cytosolic red fluorescent protein (RFP) to mark cell boundaries. Moreover, a number of software tools that processes bright-field images have been described, such as Pombex [Pen13], CellStat [Kva08] and CellSerpent [Bre11]. Pombex segments *Schizosaccharomyces pombe* cells in bright-field images. For *S. cerevisiae*, however, Pombex is not optimal as they have different shape characteristics. CellStat is able to segment bright-field images of *S. cerevisiae*, but it has a constraint on the cells to be detected, as they must not be encapsulated by other cells [Kva08]. CellSerpent utilizes an active contour segmentation algorithm for cell detection, where only few features are measured [Bre11]. Moreover, both CellStat and CellSerpent requires the Matlab software to be installed. In another study, images obtained by confocal microscopy are analyzed to investigate the localization of the plasma membrane protein *Mrh1p-GFP* [Bir11]. Image analysis is carried out using modules derived from the Acapella software that is supplied with the Opera microscope that the software is developed for, and hence the modules are not freely available.

As expression levels and subcellular localization of tagged genes are highly variable, we will use bright-field for the detection of cells. In addition, the information from the fluorescent datasets is given. Nevertheless, the software offers segmentation methods that works well with fluorescence images when needed. In the next section, we describe the workflow of our developed platform. The platform is named *YeastAnalysis*.

2.2 YeastAnalysis Workflow

Image analysis was probed to accomplish further progress to complement flow cytometric analysis for the *S. cerevisiae* yeast cells. This should be part of a comprehensive image analysis platform. Such platform must have the following components: (1) Segmentation, (2) Measurement, (3) Data analysis and visualization and (4) a graphical user interface (*GUI*).

The proposed workflow is depicted in Fig. 2.1. The three major components offered by the system are shown as separate blocks connected through the *GUI*. In the next sections we elaborate on each of these modules including segmentation, measurement, data analysis and visualization and the *GUI*. Subsequently, a small scale validation experiment is presented in Section 2.7. Section 2.8 discusses the hardware and software tools we used during the development of the platform. We complete this chapter with a conclusion in Section 2.9.

2.3 Segmentation Module

For different image modalities, different segmentation methods give different results. In our research we identify four different type of images. In Fig. 2.1 we label these images as Fluorescent Type A, Fluorescent Type B, Bright-field Type C and Bright-field Type D. Four different segmentation processes are developed for each image type. As a general approach we always follow the same segmentation path used with Type D; i.e. applying *Hough Transform* and *Minimal Path* algorithms as the core methods. However, when results are not satisfactory, one can always choose a different method according to the image modality. Type A fluorescent images are those whose cells show evenly distributed fluorescent signal throughout the cell. Example of this type are images showing the expression of *BMH* gene tagged with *GFP* as a reporter gene. Type B fluorescent images are those whose cells show strong signal throughout the cell or nucleus and possesses speckle noise. For example, images showing the nucleus stained with *DAPI* signal, and the cytoplasm of the cell shows speckle noise throughout the cytoplasm possibly corresponding to mitochondria. Type C images uses brightfield channel where the cell structures are well separated and their contours are well highlighted. Example of this type, is when the fluorescent channel is weak and its segmentation is not possible with standard methods; then we use this brightfield channel to detect the cell objects. Type D images represent the most sophisticated case, where the fluorescent signal is weak, and the brightfield channel has cells clumped within other cells, or attached to budding cells, leading to disconnected cell contours.

In this section, we discuss two types of segmentation algorithms used by the system. The first is filter-based and the second uses *Hough Transform* and *Minimal Path* algorithms.

2.3.1 Filter-Based Methods

In these type of segmentation methods, we use one of three different filters as its core processing followed by triangle auto-threshold, fill holes and morphological watershed algorithms. The three different filters are the Sigma, Despeckle and Sobel. These filters will be discussed first. Subsequently we discuss the triangle auto-threshold, fill holes and morphological watersheds.

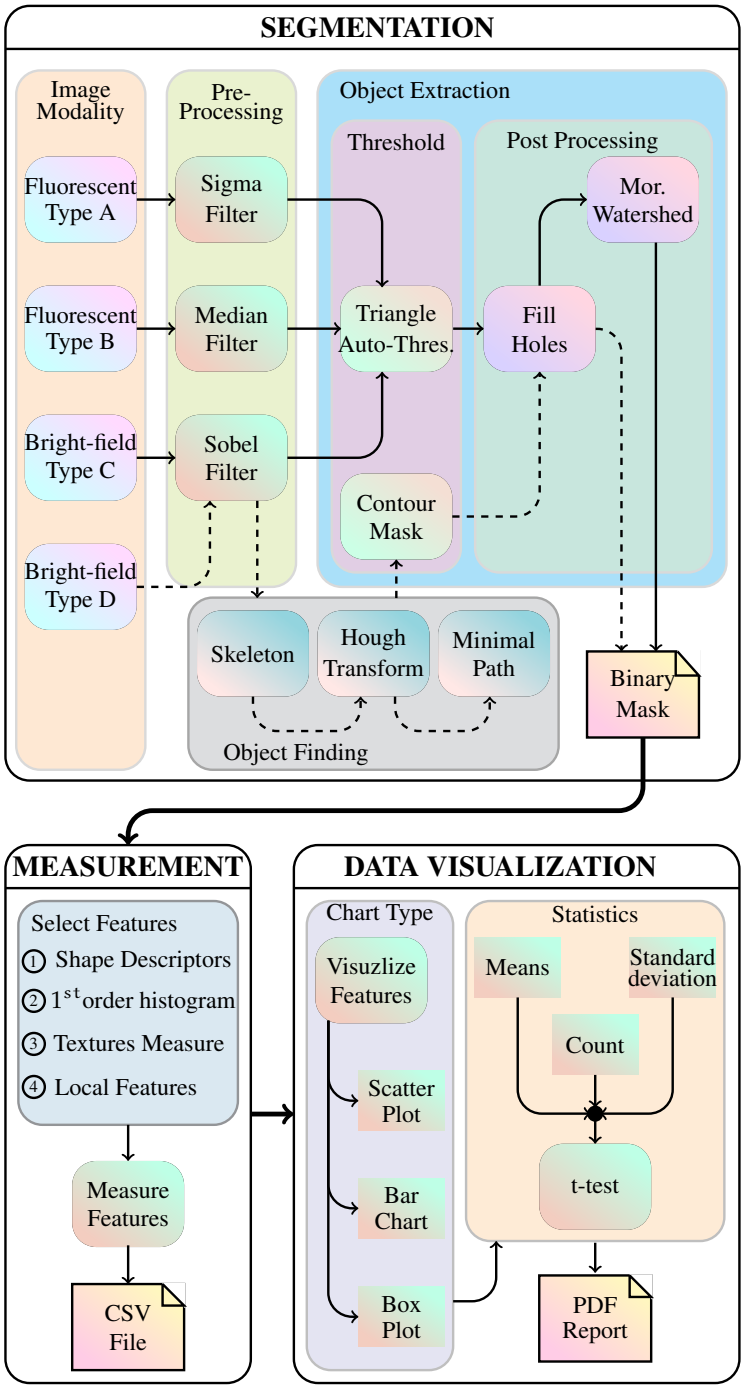


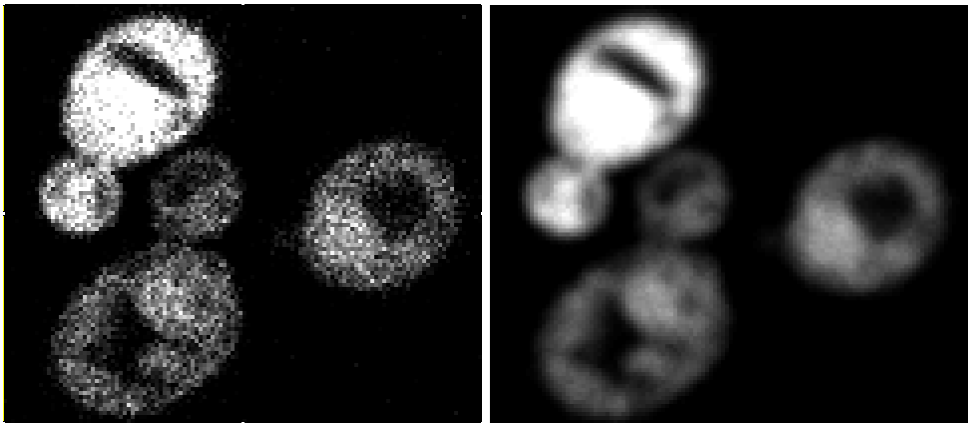
Figure 2.1: *Workflow of the YeastAnalysis Platform.*

Sigma Filter

We use Sigma filter [Lee83, Sch07] in one of the simple segmentation methods on fluorescent images. This filter is motivated by the sigma probability of the Gaussian distribution, and it smooths the image noise by averaging only those neighbourhood pixels that have the intensities within a fixed sigma range of the center pixel. Smoothing and blurring the images through Sigma filter makes it possible to acquire binary masks that better represent the shape of the objects. It was selected for a number of advantages including:

- Noise near edge areas will be smoothed without blurring the edge because only pixels on one side of the edge are included in the average;
- Preservation of subtle details and linear features;
- Not sensitive to shape distortion;
- Retention of step edges and sharpening of ramp edges;
- Removal of high-contrast spot noise;
- Computationally efficient.

Figure 2.2 illustrate a sample application of sigma filter on *S. cerevisiae* cells.

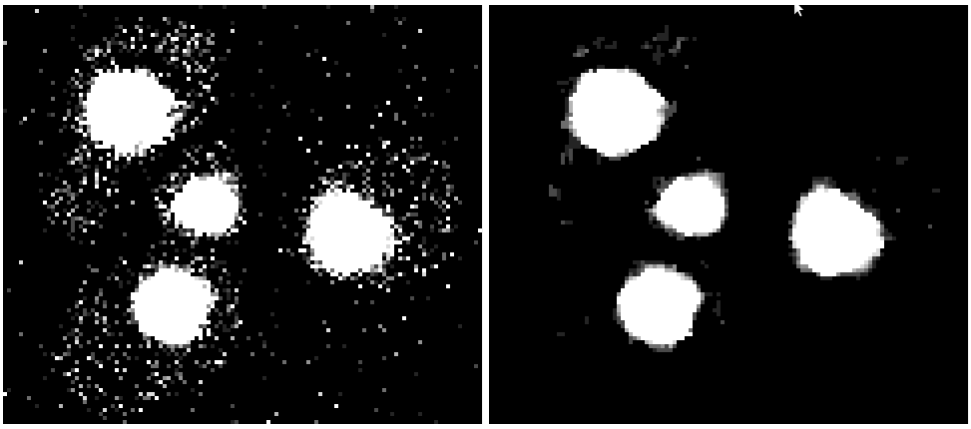


(a) Fluorescence Image (b) Applying Sigma Filter

Figure 2.2: *Applying Sigma Filter on a Fluorescence Image.*

Median Filter

In some fluorescent images, there exists speckle noise, also known as salt-and-pepper kind of noise. Since Median filters are well known as a good approach to remove such kind of noise [Gon08], we apply a Despeckle algorithm [Ras16], which is a median filter that replaces each pixel with the median value in its 3 x 3 neighbourhood. Figure 2.3 shows a threshold image of yeast cell nuclei before and after the application of the Despeckle algorithm.



(a) impulsive noise (b) Application of a Median Filter

Figure 2.3: *Median Filter.*

Sobel Filter

Sobel filter is a discrete differentiation operator, computing an approximation first estimate of the gradient of the image intensity function. We use the Sobel filter to highlight and detect the edges of the cells. At each point in the image, the result of the Sobel operator is either the corresponding gradient vector or the norm of this vector. This vector has the important geometrical property that it points to the direction of the greatest rate of change at a certain location (x, y) in the image. The Sobel operator is based on convolving the image with a 3x3 filter masks. These masks are separable and integer valued in the horizontal and vertical directions and is therefore relatively inexpensive in terms of computations [Gon08, Ras16].

Triangle auto threshold

In the filter-based segmentation methods, the Triangle auto-threshold [Zac77, Ras16] is used to obtain a binary image as shown in Fig. 2.5(a). The threshold level in the Triangle auto threshold method is determined on the basis of the histogram of pixel intensities as illustrated in Fig. 2.4.

We evaluate the distance from the histogram at every level to the hypotenuse of the triangle having the histogram height and dynamic range as sides. The histogram level having the maximum distance corresponds to the final threshold used by this method. In Fig. 2.4, point T corresponds to such threshold.

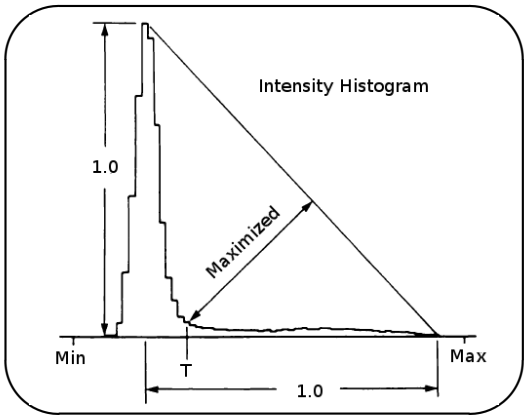


Figure 2.4: *Triangle auto threshold.*

The microscope we used throughout our research produce images whose histogram has a weak bimodal distribution and a high kurtosis as it has a distinct peak near the mean, which belongs to the low extreme of the histogram. Triangle's method has proved its superiority for these kind of images as it assumes a peak near one of the extremes and searches the threshold value toward the other end [Car12]. Additionally, Triangle's method is applied once on the histogram and it is computationally in-expensive unlike some other threshold methods such as the Otsu and Isodata.

The final result of the threshold operation is a binary image. Sometimes some inner parts of the objects are not recognized as foreground pixels. Hence, we apply *Fill Holes* algorithm.

Fill Holes

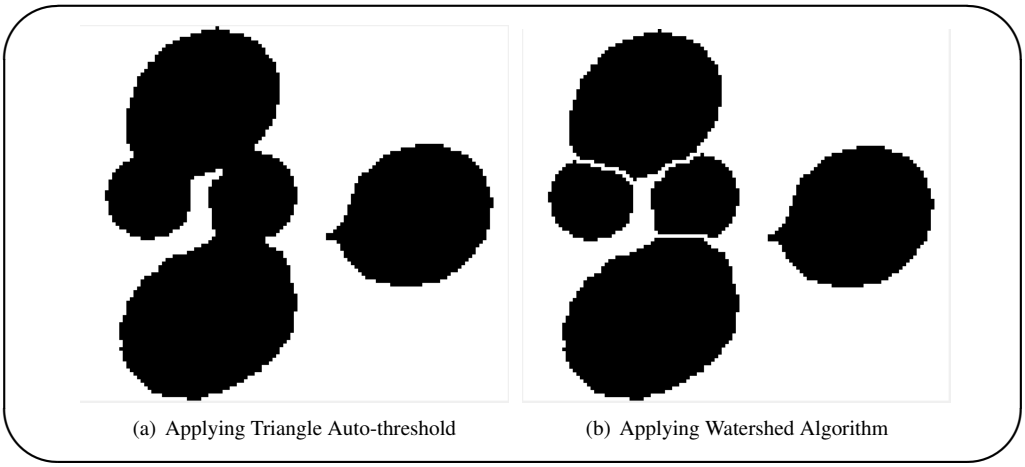
After obtaining a binary image through threshold, the *Fill Holes* algorithm simply walkthrough the pixels starting from the boundaries of the image and filling all the pixels with a background label and stops when facing a foreground pixel. Labelling all the background pixels leaves all the closed contours and their inner regions to be considered as foreground pixels. This operation produces a binary mask of the cells. To separate the cells that are connected together, we apply the morphological watersheds discussed hereafter.

Morphological watersheds

As Fig. 2.5(a) shows, the obtained binary image might have cells clumped together. For separation of such clumped cells, *morphological watersheds* [Gon08, Ras16] is used to obtain the final binary mask of the cells. Figure 2.5(b) shows the same cells from Fig. 2.5(a) processed with *watersheds*. The concept of *watersheds* is based on visualizing an image in three dimensions: two spatial coordinates versus intensity. In such a "topographic" interpretation, we consider three types of points: (a) points belonging to a regional minimum; (b) points at which a drop of water, if placed at the location of any of those points, would fall with certainty to a single minimum; and (c) points at which water would be equally likely to fall to more than one such minimum. For a particular regional minimum, the set of points satisfying condition (b) is called the *catchment basin* or *watershed* of that minimum. The points satisfying condition (c) form crest lines on the topographic surface and are termed *divide lines* or *watershed lines* [Gon08]. These *watershed lines* are located by the algorithm to separate the clumped *S. cerevisiae* cells.

2.3.2 Hough Transform and Minimal Path

The second type of segmentation methods is developed to be used as a general segmentation algorithm on bright-field images and it uses our novel *Hough Transform* and *Minimal Path* algorithms in its core. The idea of the new method is to use *Hough Transform* to locate the geometrical circles that contains part of the cell contours in a skeleton of the gradient image where the foreground shapes are reduced to a skeleton of one pixel width. Subsequently, a polar transformation is applied to resample the pixels surrounding the cells, and apply *Minimal*



(a) Applying Triangle Auto-threshold (b) Applying Watershed Algorithm

Figure 2.5: *Threshold Image and Separating Cells.*

Path algorithm to find the path (full contour) of every cell. Chapter 3 explains more about this method. In addition, Chapter 3 describes an extension for the *Hough Transform* and *Minimal Path* algorithm by adding a contour expansion process. This expansion offers more optimized contours in some image modalities. The extracted contours are then used to obtain a binary mask of every yeast cell, enabling its measurement in the overlaid fluorescence channel.

After the segmentation process, the extracted contours or the binary masks are used to measure the individual *S. cerevisiae* yeast cells. The measurement is performed within the Measurement module in the image analysis pipeline. The Measurement module is discussed hereafter.

2.4 Measurement Module

This part of the workflow measures and describes the *S. cerevisiae* yeast cells for various features and textures. Using the labelled objects from the binary masks generated during the segmentation process, measurement of individual cells is made possible. This system provides an option to choose from a list of features to be measured; moreover, it provides an option to exclude outliers based on the values of the circularity feature and size of the measured cells.

In this platform, basic feature extraction techniques in image analysis are considered, while more sophisticated techniques are studied later in Chapter 4. We categorize the basic techniques used in this platform into two classes. The first is based on first-order statistics and the second is based on texture measurement. In the following sub-section we start discussing the first-order statistic features; subsequently, we discuss the basic texture measurements.

2.4.1 First order statistics based features

By first-order statistic features, we mean all those features computed based on single pixel values including first-order histogram based features. This is in addition to basic shape and intensity descriptors.

Assuming that our microscope image is a function $f(x,y)$ of two space variables x and y, $x = 0,1,\ldots.N-1$ and $y = 0,1...M-1$. The function $f(x,y)$ can take discrete values $i = 0,1...L-1$, where L is the total number of intensity levels in the image. The intensity-level histogram is a function showing the number of pixels for each intensity level in the whole image. This function is depicted in Eq. 2.1.

$$h(i) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \delta(f(x,y),i), \tag{2.1}$$

where $\delta(j,i)$ is the Kronecker delta function, depicted in Eq. 2.2. The Kronecker delta function simply increments the intensity level histogram by the value of one at every pixel whose intensity value $j = f(x,y)$ is equal to that histogram intensity level i.

$$\delta(j,i) = \begin{cases} 1, j = i \\ 0, j \neq i \end{cases} \tag{2.2}$$

The histogram of intensity levels is obviously a concise and simple summary of the statistical information contained in the image. Calculation of the grey-level histogram involves all single pixels. The histogram contains the first-order statistical information about the image. The histogram can also be computed for a sub-image, i.e. the region of interest (RoI) of cell objects. Different useful image features are worked out from the histogram to quantitatively describe the first-order statistical properties of the objects. In this research, we considered many basic shape and intensity descriptors based on the first order statistical information in the images. A list of those features is depicted in Table 2.1.

2.4.2 Texture Measurement

An image or object texture is an important metric as it gives us information about the spatial arrangement of intensities in that image or selected region of an image, i.e. the object. A set of basic texture features [Gon08] is available to be measured in the measurement module. A list of these texture features is depicted in Table 2.2.

All these measurements of features and simple feature textures are saved automatically into a CSV (comma-separated values) file that is used at the following step to generate a report,

Table 2.1: Features based on first order statistics.

Features	Description
Size	The size of a yeast cell (object) is simply the number of pixels occupied by that cell in the image, and for SI unit, it is multiplied by the area of one pixel in square μ-meters (width x height of pixel).
Total Intensity	The total intensity of a yeast cell is the total gray-level values of the pixels occupied by this cell, which ranges between 0 and 255 for the gray images used.
Intensity Standard Deviation	The standard deviation from the mean of the intensity values in each cell.
Perimeter	The perimeter representation method used to estimate the perimeter of the cell is that of Vossepoel and Smeulders [Vos82]. This representation is based on chaincodes: $$L_{vs} = (0.980).N_e + (1.406).N_o - (0.091).N_c \quad (2.3)$$ where N_e, N_o and N_c represents the number of even codes, odd codes and corners in the chaincode of cell boundaries [Gon08].
Density	The density of a particle object is its area multiplied by the average mean of gray-level values: $$d = A * \mu \quad (2.4)$$ where *d* represents density, *A* represents area and μ represents mean of intensity of a measured cell.
Circularity	The circularity of detected shapes [Ras16]. $$\text{Circularity} = \frac{4\pi x \text{Size}}{\text{perimeter}^2}. \quad (2.5)$$
Vacuole Size	If the fluorescent protein is expressed by genes known to have expression in the cytoplasm and nucleus without its vacuoles, the size of the central vacuole can be estimated. This is computed by using a vacuole filter algorithm that looks in the fluorescent images for a region, inside the RoI (region of Interest) representing every cell, that forms the largest connected region with the lowest intensity values.
Membrane Features	Different features can be measured in the region close to the cell borders where membrane proteins are expressed. Such features include size, total Intensity and Intensity standard deviation.
Nucleus Features	For those images that contain a nuclear stain such as DAPI, the nucleus can be measured for all the features an individual cell could be measured for.

Table 2.2: Basic texture measurement.

Features	Description
Variance	The variance (μ_2 or σ^2) is a measure of intensity contrast and can be computed from the second statistical moment. $$\mu_2(z) = \sum_{i=0}^{L-1} (z_i - m)^2 . P(z_i) \quad (2.6)$$ where z_i is the intensity value of the histogram at location i, m is the mean intensity value, L is the total number of intensity levels (histogram range), and $P(z_i)$ is the corresponding histogram with i between 0 and $L - 1$ [Gon08]).
Smoothness	The variance (σ^2) is used to establish the descriptor of relative smoothness (S): $$S(z) = 1 - \frac{1}{1 + \sigma^2(z)} \quad (2.7)$$ This measure is zero for areas of constant intensities where the variance is zero there, and it approaches 1 for large values of the variance [Gon08].
Skewness	The skewness (μ_3) of the intensity histogram which is the third statistical moment. $$\mu_3(z) = \sum_{i=0}^{L-1} (z_i - m)^3 . P(z_i) \quad (2.8)$$ A negative skewness means that most of the pixel values are high and thus concentrated at the right side of the histogram. A positive skewness means that most of the pixel values are low and thus concentrated at the left side of the histogram [Gon08].
Uniformity	The uniformity (U) has a maximum value for a cell image in which all intensity levels are equal [Gon08]. $$U(z) = \sum_{i=0}^{L-1} P^2(z_i) \quad (2.9)$$
Entropy	The Entropy (e), which is a measure of variability, is zero for constant images [Gon08]. $$e(z) = -\sum_{i=0}^{L-1} P(z_i) . \log_2 P(z_i) \quad (2.10)$$

or to be inspected manually. More advanced sophisticated feature extraction techniques are considered in our research to improve the recognition of objects after the segmentation process and to discriminate yeast cells into different categories with the purpose of identifying subtle patterns. These sophisticated features are explained in Chapter 4.

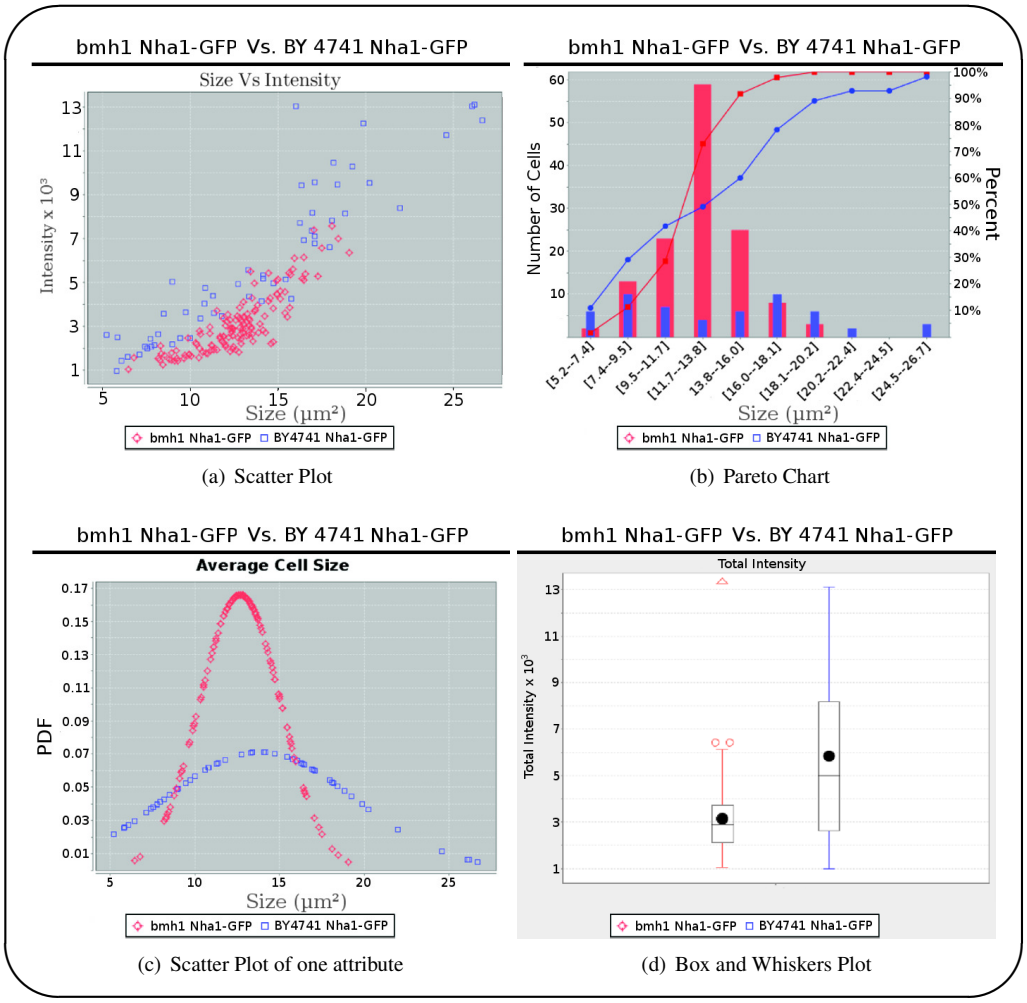
2.5 Data Analysis and Visualization Module

After measurement, usually molecular biologists would like to compare two classes of cells to study any significant changes in cell size or fluorescence intensity in cells having a certain gene mutation or cultured in a different medium (ex. 2M NaCl, 50mM KCl, etc...). The changes in features of cells in different conditions are compared usually to wild-type cells. To assist in achieving this goal of comparison, our system generates automatically a report in pdf format including basic statistics about the experiment and graph charts to visualize the results.

The statistical information created into the report includes counts of the cells belonging to different cultures, the mean value of their surface area, and fluorescence intensity along with its standard deviations. The unpaired student t-test is performed to report the t-value and p-value to assist in recognizing the significant of the difference between two different cell samples.

The module in the yeast analysis pipeline performs statistical descriptive data analysis. More advanced analysis of the measurement data is considered in Chapter 4. Such advanced analysis makes use of machine learning techniques. This machine learning approach is used to improve the segmentation process. It also makes use of a similar approach to discriminate various cell conditions aiming at identifying subtle patterns.

The process of data analysis would not provide significant details for users without some visualization tools. This platform offers some data visualization techniques to assist biologists in extracting meaningful information from their experiments. A very frequently used graph chart in yeast cell biology are those scatter plots that visualize the different values of cell surface area on one axis against fluorescence intensity on the other, with a different label for each cell culture. Figure 2.6(a) depicts a sample of such plot. A scatter plot is typically used to compare between two observations represented by two variables to study their correlation, and determine if the two variables exhibit the same or opposite direction. Other visualization options available are charts visualizing one attribute of the cells in one chart, as a Pareto chart (cf. Fig. 2.6(b)), a scatter plot visualizing the Gaussian density distribution of the data for the two cell cultures (cf. Fig. 2.6(c)), or a box and whiskers plot to facilitate the analysis of the range of data (cf. Fig. 2.6(d)).



(a) Scatter Plot

(b) Pareto Chart

(c) Scatter Plot of one attribute

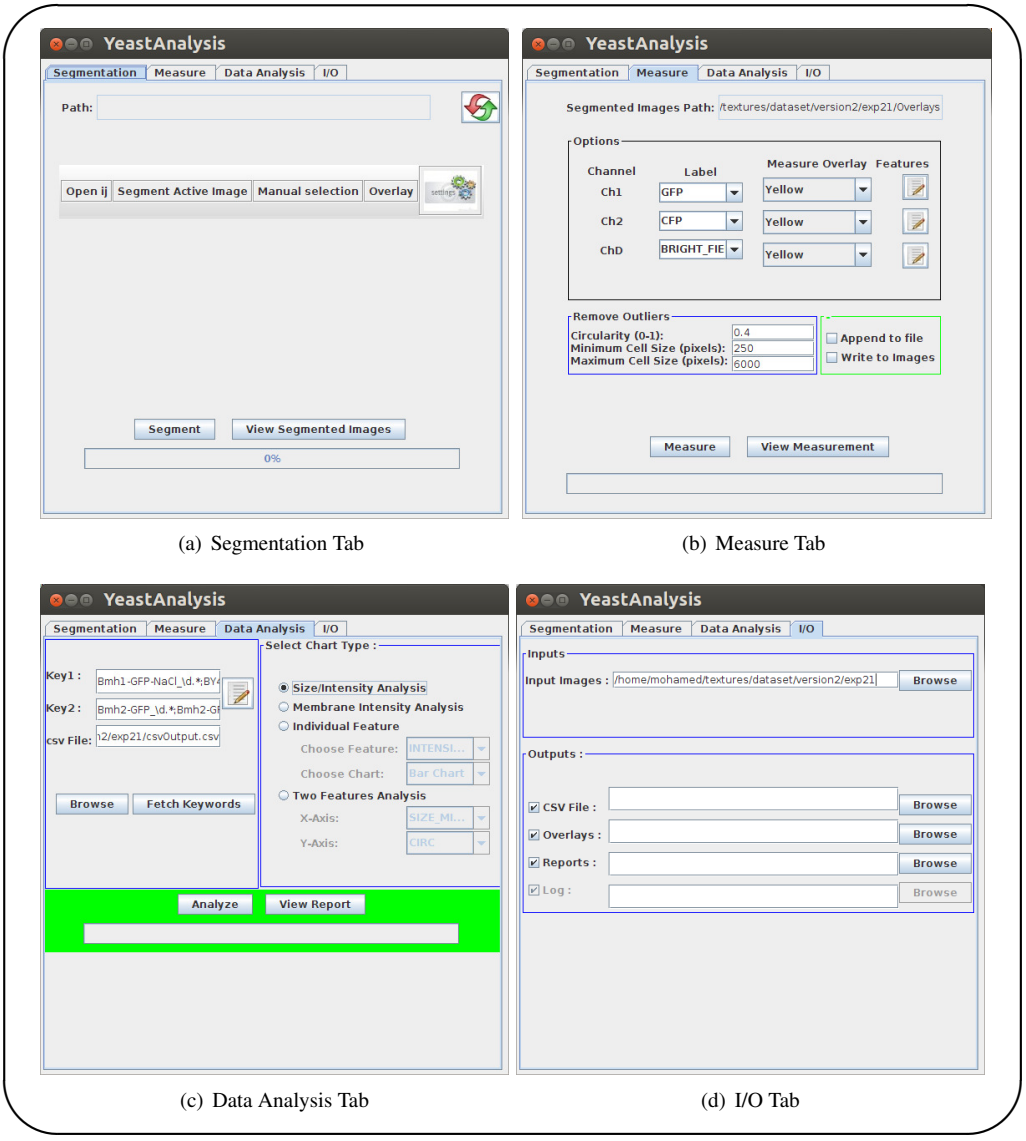
(d) Box and Whiskers Plot

Figure 2.6: *Graph Charts to visualize measurement.*

2.6 System GUI

The graphical user interface is a necessary tool that connects the workflow components together. In its basic structure, the *GUI* is composed of the different modules mentioned in the workflow in Fig. 2.1. These modules are provided as separate tabs in a user friendly interface. Figure 2.7 depicts the interfaces of the *GUI*. Figure 2.7(a) shows the interface where a user can set the segmentation method and parameters to perform the segmentation process. Figure 2.7(b) depicts the interface in the measure tab, where users can specify the channels and features to be processed. Figure 2.7(c) shows the data analysis part, where one can set multiple pairs of keywords corresponding to different cell strains or cultures. The last tab in the *GUI* is

an interface to set the directory of the input images as well as the output directories of the segmentation, measurement and analysis results. Besides being implemented as a stand-alone desktop application, this *GUI* was also implemented as a plugin for the imageJ software [Tle12].



(a) Segmentation Tab

(b) Measure Tab

(c) Data Analysis Tab

(d) I/O Tab

Figure 2.7: *GUI of the Yeast Analysis platform.*

2.7 Workflow validation

To validate *YeastAnalysis* platform, a small scale experiment is performed to compare two cell cultures. The first sample belonging to a strain of cells having a mutation in the *BMH1* gene and expressing *Nha1* protein attached to a green fluorescent protein (*GFP*). This cell strain is labelled *Δbmh1 Nha1-GFP*. The second sample is from a wildtype cell strain expressing *Nha1* protein attached to *GFP*. This strain is labelled *BY4741 Nha1-GFP*. A set of 18 images are acquired with a Zeiss LSM5 Exciter confocal microscope. From the segmentation, 188 cells are detected using our novel segmentation method on the bright-field image channels; i.e. *Hough Transform* followed by Circular Shortest Path (*HCSP*). More details about this method are explained in Chapter 3. Ensuing segmentation, the cells are measured for various features and all the measurements are written to a CSV file. The measurements are then analyzed automatically to generate a report including graph charts and statistics. Such charts are shown in Fig. 2.6. Example of the statistical information automatically generated into the report is shown in Fig. 2.8.

Having a look at the chart and the statistics, and specifically the p-values of the cell size and intensity, reveals meaningful information. It is clearly visible that mutant cells belonging to the *Δbmh1 Nha1-GFP* strain are smaller and have less *GFP* fluorescence than the wildtype cells in *BY4741 Nha1-GFP* strain. The results obtained using the image analysis platform we developed in our research are in good correspondence with the flow cytometry test. Figure 2.9 reveals these results obtained by flow cytometry.

Un-paired Student's t-test:
Size t-Value : -1.46
Size P-Value (Assuming null hypothesis): 0.15
Intensity t-Value : -5.54
Intensity P-Value (Assuming null hypothesis): <0.001

bmh1Nha1-GFP : 133 Cells.
Size Mean : 12.67
Size Stdev : 2.4
Intensity Mean : 3.14×10^3
Intensity Stdev : 1.39×10^3

BY4741Nha1-GFP : 55 Cells.
Size Mean : 13.82
Size Stdev : 5.61
Intensity Mean : 5.84×10^3
Intensity Stdev : 3.5×10^3

Figure 2.8: *Sample report from the experiment analysis.*

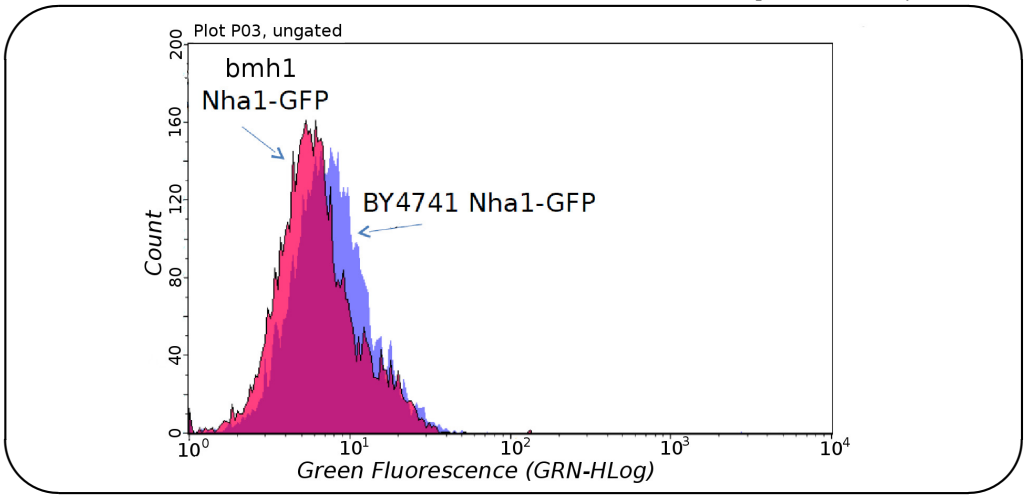


Figure 2.9: *Results of the Flow Cytometry test.*

2.8 Development of *YeastAnalysis*

The hardware used in the development of *YeastAnalysis* is a desktop computer equipped with an Intel® Core™ i7-2600 CPU @ 3.40GHz × 8 processor, and 16 GB of memory installed with the Ubuntu 12.04 LTS operating system. The software is written in Java using JDK-1.7 including a variety of open-source packages. An ImageJ library package [Ras16] is utilized to access some existing functions such as the Watershed Algorithm, Fill Holes Algorithm and Overlays feature. ImageJ plugins are also imported, such as the Sigma-Filter-Plus which is an implementation of the sigma filter [Sch07]. The output of measurement was integrated with spreadsheet applications through Java Excel API [jex16]. The t-test and statistical analysis of these measurements are performed with the support of the Apache commons Mathematics library [apa16]. Visualization of statistical results was made possible with the JfreeChart API [jfr16]. Creation of the Pdf reports was done with iText programmable pdf software [ite16]. Moreover, for drawing of the ground-truth images we used the TDR software package [Ver04]. *YeastAnalysis* and a manual are available for download from [git16].

2.9 Conclusion

In this chapter, we addressed the components and processes that builds up an automatic image analysis pipeline for the objective analysis of gene expression in single cells, i.e. yeast cells. We designed a workflow for this pipeline. This workflow is composed of a segmentation module, a measurement module, a data analysis and visualization module and a *GUI* connecting the aforementioned modules.

The segmentation module has various segmentation and image processing algorithms adopted in this system including the sigma filter based method, median filter based method, Sobel filter based method and *Hough Transform* and *Minimal Path* algorithms. Including as well the post-processing methods such as Triangle, fill holes and watershed. After the segmentation phase, there is the measurement phase. The measurement module has various features that could be selected and measured. These features are categorized into first-order statistical features and basic texture measurements based on the histogram intensities. After the measurement phase, comes the data analysis and visualization phase. In this part, the statistical descriptive data analysis that is performed include the computation of basic statistical metrics and Student's t-test for statistical analysis. The *GUI* of the developed image analysis platform connects the various modules as separate tabs. There is also a room to add new components. Finally, the platform is validated in a small scale experiment; the analysis of a study experiment revealed decreased expression of the *Nha1* protein in a yeast strain having a mutation in the *BMH1* gene compared to the wildtype strain.

The results demonstrate that this platform can potentially contribute to the improvement of the objective analysis and diagnosis of gene expression studies in yeast. It reveals an advantageous comprehensive image analysis platform that can be used in the laboratory assisting in analyzing experiments.

3

Hough-based Contour Extraction and Optimization

“In this chapter, we present our novel algorithm for the segmentation of ovoid objects. Our method implements a variation of the Hough Transform and Minimal Path algorithms. We propose equations and parameters to control the detection of objects through Hough Transform. The contours of these objects are extracted through minimal path algorithms implemented on a polar resampled representation of the object images. In addition, we present a contour expansion algorithm using the same polar representation of object images. Such expansion is necessary under some microscope settings.”

This chapter is based on the following publications:

- Mohamed Tleis and Fons J. Verbeek. "Extracting contours of oval-shaped objects by Hough transform and minimal path algorithms." In Sixth International Conference on Digital Image Processing, pp. 915903-5. International Society for Optics and Photonics, 2014. *(Excellent Paper Award)*.
- Mohamed Tleis and Fons J. Verbeek. "Contour Expansion preceded by the Application of Hough transform and Minimal Path Algorithm". In Fifth International Conference on Image Processing Theory, Tools and Applications, pp. 149-154. IEEE, 2015.

3.1 Background

In Chapter Two, the *Hough* transform based segmentation methods and contour optimization were introduced. This chapter explains in details the underlying algorithms of such methods and their implementation. The following section introduces *Hough* transform, generation of the cube-like accumulator, and how this accumulator is threshold in order to get information about the object locations in the image. In Section 3.3 polar transformations are described and how these are applied to ovoid objects to generate a rectangular array that is used to obtain the minimal path in the object image. In Section 3.4, the implemented minimal path algorithms are explained. The polar representation of object images is also used for the purpose of contour optimization; our novel optimization method to expand object contours is explained in Section 3.5. Section 3.6 validates the introduced methods by comparing them to other state-of-the-art solutions and assess their performance under various noise levels.

3.2 Hough Transform

In the daily practice of processing microscope images, sets of pixels yielded by edge detection methods seldom characterize the edge because of noise, breaks in the edges due to non-uniform illumination and the effects that introduce spurious discontinuities in intensity values. This is often observed with the *S. cerevisiae* cells in bright-field images.

A well-known global approach to edge linking is *Hough* transform [Gon08]. *Hough* transform is applicable to any function of the form $g(v, c)$ where v is a vector of coordinates and c is a vector of coefficients. Since many objects in systems and micro-biology such as *S. cerevisiae* yeast cells are characterized as having nearly ovoid shapes [Fel10], detecting circular arcs in yeast images would help in detecting the yeast cells. A circle is represented as in Eq. 3.1, which can be rewritten in its normalized form as depicted in Eq. 3.2.

$$(x - c_1)^2 + (y - c_2)^2 = c_3^2 \tag{3.1}$$

$$\begin{aligned} x &= r.\cos\theta \\ y &= r.\sin\theta \end{aligned} \tag{3.2}$$

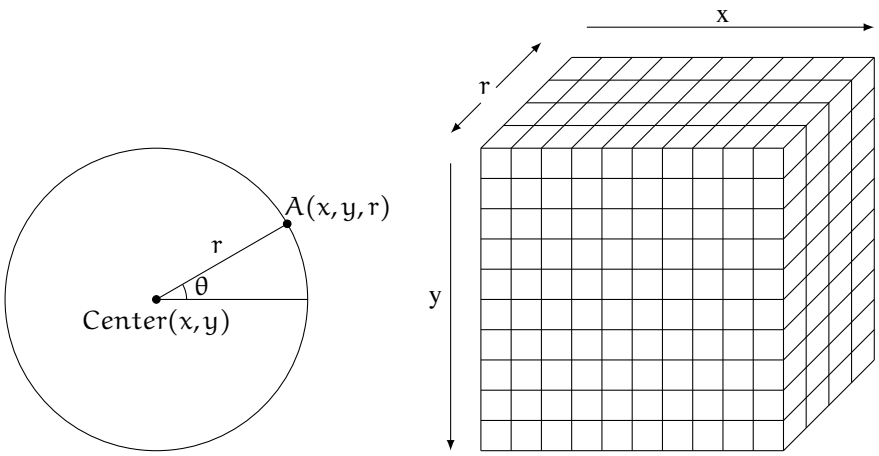
Our implementation of the *Hough* transform includes two steps. The first step is to fill the accumulator and the second is to threshold this accumulator in order to extract the circles corresponding to the estimated object locations. These steps are discussed hereafter.

3.2.1 Filling the Accumulator

Figure 3.1(a) illustrates the geometrical interpretation of the parameters x, y, θ and r. The x, y, r parameter space is sub-divided into accumulator cells forming a 3-D cube-like cells and accumulator of the form $A[x, y, r]$. The x and y dimensions of this accumulator are related with the width and height of image I and the r dimension is defined by the constraint of the possible object radius values. This accumulator is illustrated in Fig. 3.1(b). The procedure is to increment x and y and solve the equations in Eq. 3.2 for θ at a pre-specified value of the radius r and update the accumulator cell associated with the triplet (x, y, r), i.e. increment accumulator cell by 1 if a foreground edge pixel is found at angle θ for the circle of center (x, y) and radius r. The flowchart in Fig. 3.2 illustrates the filling process in more detail. At radius r specified from a range of radii, all the pixels in the image are checked and if a certain pixel is an edge pixel, i.e. foreground in the binary image, the accumulator cell is incremented by the value of one. After filling the *Hough* accumulator array, we need to decide on which pixels in the binary image are to be considered as center of circles. This required specifying a rule to find a threshold for the accumulator. The following sub-section discusses our approach to find this accumulator threshold.

3.2.2 Accumulator Threshold

Hough values in the accumulator array are related to the number of pixels located in the circumference of a certain circle with a specified center c and radius r where a maximum of $2\pi r$ pixels can belong to that circumference. A threshold value is set to allow detection of structures that lie a pre-specified percentage of the circumference. This threshold starts at $2\pi r$ and decrements for smaller values of the radius. The threshold value is calculated according to



(a) Geometrical interpretation of parameters x, y ,θ and r.

(b) Cube Accumulator $A[x, y, r]$.

Figure 3.1: *The Hough Accumulator.*

the equations depicted in Eq. 3.3 and 3.4, where r is the radius of the circle, r_{max} and r_{min} are the maximum and minimum specified radius values, r_{index} is an index to track the current radius value. α and β are control values. Parameter p in Eq. 3.4 ensures that the threshold value is relatively high for smaller circles. This allows detecting some deformed shapes at a smaller radius $(r-1)$ if not detected at radius r.

$$T = 2\pi r - \{2\pi r \times \alpha + p\} \tag{3.3}$$

$$p = \beta \times (r_{max} - r_{min}) - r_{index} \tag{3.4}$$

The flowchart in Fig. 3.3 walks-through the process of using the threshold value to get the geometrical arcs that form part of circles in the binary image. Starting from the largest radius and decrementing until the minimal radius, we loop through *Hough* values in every cell of the cube accumulator starting from the maximum *Hough* value until the specified threshold given in Eq. 3.3. As long as there are pixels associated with that *Hough* value, we check if the circles occupying that space overlap with the previously detected circles. If not, then the space occupied by the new circle is reserved, preventing subsequent circles from occupying this space. The parameters α and β control the threshold equation depicted in Eq. 3.3 and 3.4.

The parameter α is a weighting factor in the range $[0,1]$ and is used to determine the percentage of pixels that must exist as edge pixels in order to consider them as being part of a closed circumference of a circle; e.g. at $\alpha = 0.5$, at least 50% of the pixels located on the circumference must be edge pixels. The parameter β acts similar to α except that it increases in value at the examination of circles with lower radius values, allowing more shape deformation at lower radii.

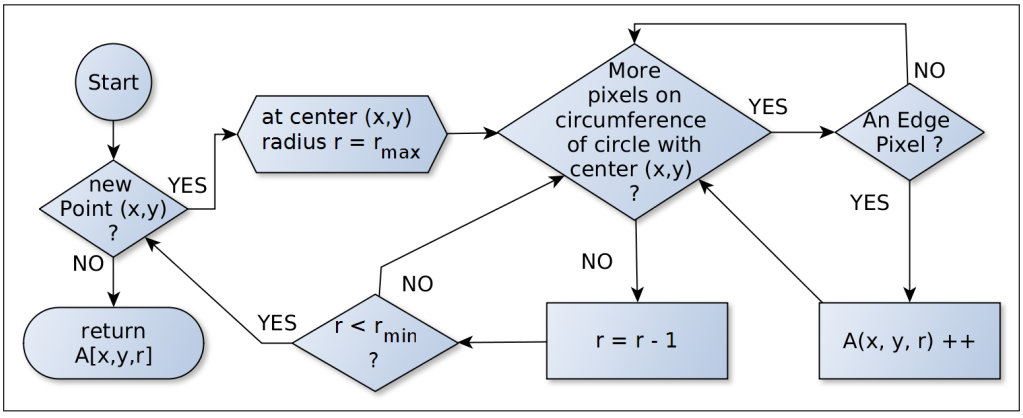


Figure 3.2: *Filling the Hough Accumulator.*

3.3 Polar Transformation

After estimating the initial position of the ovoid objects such as *S. cerevisiae* yeast cells in bright-field microscope images through *Hough* transform, our objective is to extract the exact contour of that object. For this, we resample each object into a polar representation for each object where a polar transformation is applied on the image part as a *RoI* for that object. On this polar image, we extract the minimal path corresponding to the actual contour of the object. In this section, we explain the Polar Coordinate system? How the transformation is achieved from cartesian to polar system? And how is that applied on actual objects such that *S. cerevisiae* cells? The minimal path algorithm is discussed later in Section 3.4.

3.3.1 Polar Coordinate system

In 2D space, the polar coordinate system is a two-dimensional coordinate system in which each point on a plane is determined by a distance from a reference point and an angle from a reference direction. The reference point is analogous to the origin of a cartesian system. It is known as the "pole", and the ray from the pole in the reference direction is known as the "polar axis". The distance from the pole is called the radial coordinate or radius and we will refer to it as r, and the angle is the angular coordinate, polar angle, or azimuth and we will refer to it as θ [Bro97].

3.3.2 Cartesian to Polar System

The cartesian coordinates x and y can be converted to polar coordinates r and θ with $r \geq 0$ and θ in the interval $(-\pi, \pi]$ using Eq. 3.5 and 3.6. The $\mathtt{atan2}$ notation in Eq. 3.6 is a common variation of the $\mathtt{arctangent}$ function as defined in Eq. 3.7. The geometrical interpretation of the relationship between the polar and cartesian coordinates is further illustrated in Fig. 3.4.

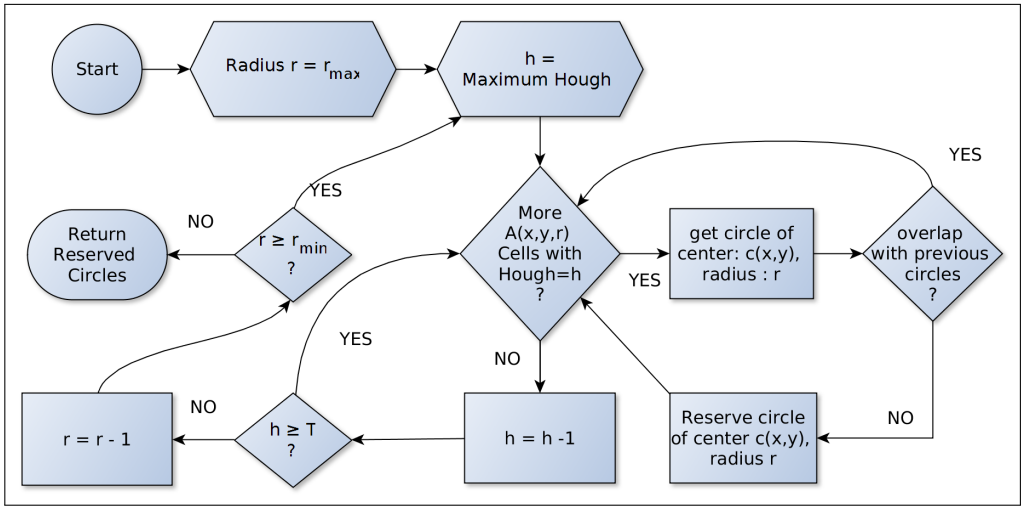


Figure 3.3: *Extracting the Circles from Image.*

$$r = \sqrt{(x^2 + y^2)} \tag{3.5}$$

$$\theta = \operatorname{atan2}(y, x). \tag{3.6}$$

$$\operatorname{atan2}(y,x) = \begin{cases} \arctan(\frac{y}{x}), & \text{if } x > 0 \\ \arctan(\frac{y}{x}) + \pi, & \text{if } x < 0 \text{ and } y \geq 0 \\ \arctan(\frac{y}{x}) - \pi, & \text{if } x < 0 \text{ and } y < 0 \\ \frac{\pi}{2}, & \text{if } x = 0 \text{ and } y > 0 \\ -\frac{\pi}{2}, & \text{if } x = 0 \text{ and } y < 0 \\ \text{undefined}, & \text{if } x = 0 \text{ and } y = 0 \end{cases} \tag{3.7}$$

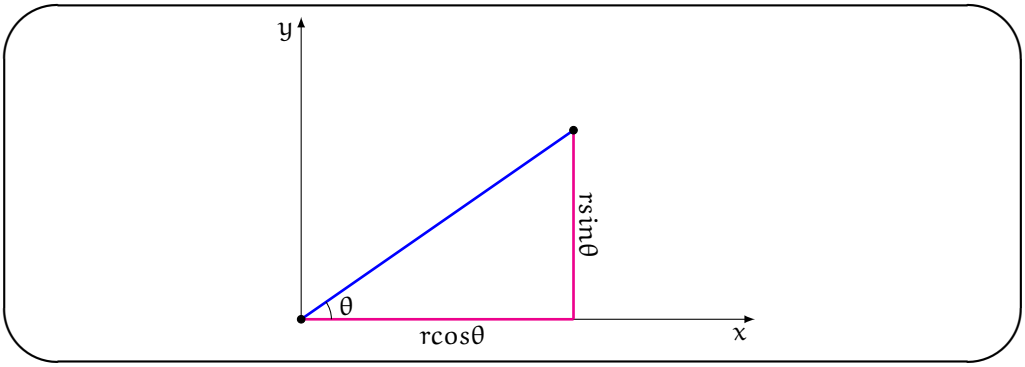


Figure 3.4: *Geometrical interpretation of relationship between polar and cartesian coordinates.*

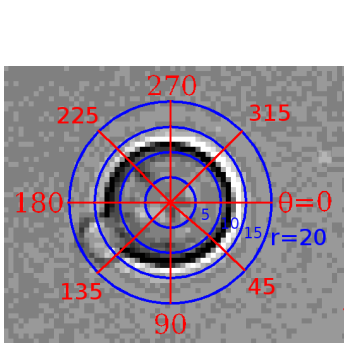
3.3.3 Polar Image of Yeast Cells

Figure 3.5(a) shows an example of a yeast cell after locating its center point by *Hough* transform. The *red* circles and *blue* lines are labels to illustrate the resampling of the polar image of the pixels surrounding the center point of the cell as shown in Fig. 3.5(b). The radius at an angle θ in the original image plane is transformed into a column in the polar image. The circles surrounding the center point, i.e. *red* circles of radius r in the original image plane in Fig. 3.5(a), are transformed into rows of the polar image, i.e. red horizontal lines in Fig. 3.5(b).

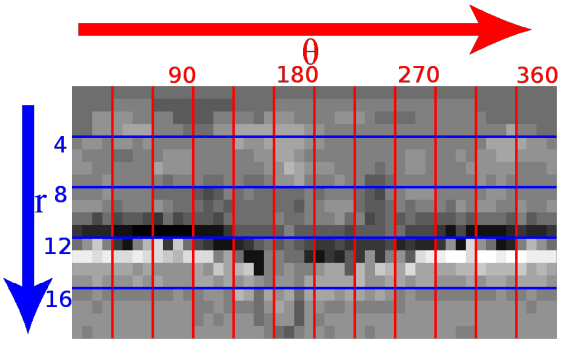
As a rule, the height of the polar image I is determined based on the scale information of the images, and a priori generic knowledge on the objects, i.e. scale × maximum radius. The width of I is dependent of the length of the contour. In order to have a sufficient region to evaluate, we use a resampling length equal to twice the height of I. The minimal path algorithm then finds the circular shortest path from the first to the last angular coordinate in image I.

3.4 Minimal Path Algorithms

Once all the center of objects, i.e. cells, are estimated as circle centers using the *Hough* transform, a polar image I relative to each circle center is resampled from the original intensity image [Kva08]. In our work, we adopted dynamic programming to extract the object contours by finding the minmal path in the polar representation of these objects. In this section, we discuss our implementation of a minimal path algorithm that uses grey weighted distance transform followed by our novel approach to extract a circular shortest path for the polar image.



(a) An example of a cell after detection of the center point by Hough transform.



(b) A polar image is generated for the yeast cell in Fig. 3.5(a). The columns correspond to the pixels along the radius at an angle θ of the largest possible circle (red lines). The rows correspond to the circles surrounding the center point (blue lines, blue circles in Fig. 3.5(a)).

Figure 3.5: *Polar transform of a yeast cell image.*

3.4.1 Grey-weighted Distance Transform

Grey-weighted distance transform algorithm was developed originally [Vin88] to find the path in grey-scale images. Let I be a polar image, which is formally a matrix of weights $C_{(r,\theta)}$; then let A denotes the first column of pixels in image I and B denotes the last column of pixels. The objective is to find the minimal path from A to B. Considering I as a neighbouring graph, let E be the edge between two neighbouring pixels p and q. E is assigned a value $V_I(E) = V_I(p,q) = I(p) + I(q)$. Let P be a path in this graph, and let $C_I(P)$ be a cost associated with path P. This cost equals to the sum of all the edge values. This provides a new metric in image I for which the distance d_I between two pixels p and q is given by: $d_I(p,q) = \min C_I(P)$, P: path between p and q. A pixel p belongs to the minimal path if and only if $d_I(p,A) + d_I(p,B) = d_I(A,B)$, therefore, the grey-weighted distance transform to A for each pixel is created by computing $d_I(p,A)$ for every pixel p, and similarly creating the grey-weighted distance transform to B. The two distance functions are added. Subsequently, a column-wise threshold is applied to the resulting image taking the minimum pixel value of the column as a threshold in order to keep the pixels p belonging to the minimal cost between A and B. This result represents the actual contour of the object. We ensure a closed contour by assigning the coordinates of the extracted pixels $p \in I$ as vertices of a polygon region of Interest (RoI), which is filled to create a binary mask representing a closed object.

Figure 3.6(a) shows a sample polar image, where the first and last columns are labelled as A and B as shown in Fig. 3.6(b). Figure 3.6(c) shows the grey-weighted distance transform to A for each pixel created by computing $d_I(p,A)$. Similarly, Fig. 3.6(d) shows the grey-weighted distance transform to B. The addition of the two distance images is shown as Fig. 3.6(e). The result of the application of column-wise threshold to obtain the minimal cost between A and B is shown in Fig. 3.6(f). This minimal path is the acutal contour of the cell. Figure 3.7 shows another example for a cell whose contour is extracted from within a group of clumped cells.

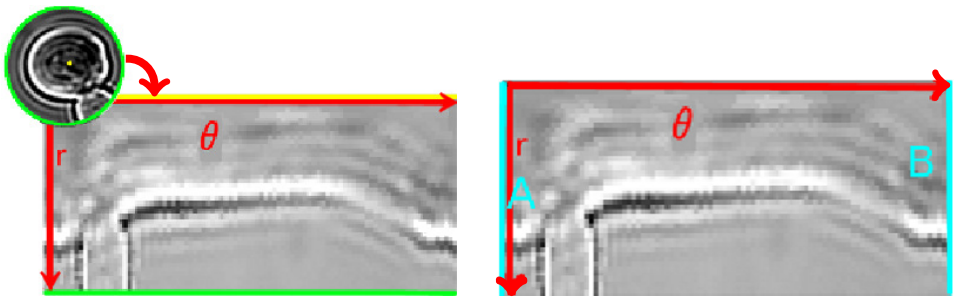
3.4.2 Circular Shortest Path

Object contours are extracted from the computation of the minimal path through distance transform as discussed in the previous subsection. However, this extracted contour is not circular as the pixels p_A and p_B of column A and B in the polar image I are actually the same pixels (at $\theta = 0$ and $\theta = 360°$). Hence, we developed an algorithm to extract the circular shortest path from column A to column B in the polar image with the constraint that a path that starts at row r_i must also end at r_i.

While the standard shortest path problem aims at finding a shortest path between two known nodes in a graph or an image grid, the shortest circular path problem is to find a closed path with minimum cost. This problem is more difficult than the standard shortest path problem since no explicit start or end node is known. Our circular shortest path algorithm uses the concept of ordinary shortest path obtained with dynamic programming [Buc97] but with the constraint that the first and last pixels p_A and p_B of the detected path P are at the same radial coordinate in order to ensure a circular path (cf. Fig. 3.8).

We apply the shortest path algorithm just once starting at the first angular coordinate $\theta = 0$ of each row r_i in image I and only evaluating the pixels p_h that would have a high probability of belonging to the path. The global minimum $\min\{C_I(P)\}$ of these shortest paths $C_I(P)$ is guaranteed to be the global circular shortest path *CSP* [App03]. Our algorithm does not require evaluating all the pixels while checking for shortest paths. It actually visits the pixels $(\frac{w^2 l}{4})$ times instead of $w^2 l$ times stated in the standard circular shortest path algorithm [Sun03], where w and l are the width and height of image I respectively.

In **Box 3.1** the pseudo-code for the circular shortest path algorithm is depicted. For each radial coordinate r in the polar image I (cf. line 2), we create a matrix $C_{(r,\theta)}$ having the same dimensions as I, i.e. $w \times l$ (cf. line 3). Ultimately all contour points need to get the status *final*.



(a) Polar representation of *S. cerevisiae* yeast cell.

(b) Labeling First and Last Column as A and B.

(c) I_A = Distance Transform to Column A.

(d) I_B = Distance Transform to Column B.

(e) Add grey-weighted distance transform to A (I_A) and that to B (I_B), i.e. $(I_A + I_B)$.

(f) Threshold $(I_A + I_B)$ in Fig. 3.6(e).

Figure 3.6: *Hough Transform and Minimal Path Algorithm to Extract Cell Contours.*

Input : A polar image I of size $w \times l$.
Integer variables i, j, i_2, k and $Cost_{min}$.
Output : Integer array CSP of length w.

Initialization: $Cost_{min} \leftarrow$ Maximum Value; $i_2 \leftarrow w-1$;
for *each radial coordinate* r **do**
  Create cost matrix $C_{(r,\theta)}$ of size $w \times l$;
  /* Initialize first column of $C_{(r,\theta)}$ to the value 1 */
  $C_{(r,\theta)}[0...l-1,0] \leftarrow 1$;
  /* For each column in the left half of the image */
  for i *from* 1 *to* $(\frac{w}{2}-1)$ **do**
    /* And for each row within the possible range R */
    for j *from* $(r - i\%\frac{w}{2})$ *to* $(r + i\%\frac{w}{2})$ **do**
      $C_{(r,\theta)}[i,j] \leftarrow I[i,j] + \min(C_{(r,\theta)}[i-1,j+k]) \quad \forall k \in [-1,1]$;
      $K[i,j] \leftarrow \underset{k}{\operatorname{argmin}}(C_{(r,\theta)}[i-1,j+k]) \quad \forall k \in [-1,1]$;
    endfor
  endfor
  /* For each column in the right half of the image */
  for i *from* $(\frac{w}{2})$ *to* $(w-1)$ **do**
    /* And for each row within the possible range R */
    for j *from* $(r - i_2\%\frac{w}{2})$ *to* $(r + i_2\%\frac{w}{2})$ **do**
      $C_{(r,\theta)}[i,j] \leftarrow I[i,j] + \min(C_{(r,\theta)}[i-1,j+k]) \quad \forall k \in [-1,1]$;
      $K[i,j] \leftarrow \underset{k}{\operatorname{argmin}}(C_{(r,\theta)}[i-1,j+k]) \quad \forall k \in [-1,1]$;
      $i_2 \leftarrow i_2 - 1$;
    endfor
  endfor
  /* Keep track of the global minimal Cost */
  if $C_{(r,\theta)}[w-1,r] < Cost_{min}$ **then**
    $Index_{min} \leftarrow r$;
    $Cost_{min} \leftarrow C_{(r,\theta)}[w-1,r]$;
  end
endfor
/* BackTrace path at row r */
$CSP[w-1] \leftarrow Index_{min}$;
for j *from* $(w-2)$ *to* 0 **do**
  $CSP[j] = CSP[j+1] + k[CSP[j+1], j+1]$;
endfor
return CSP.

Box 3.1: *Find Circular Shortest Path.*

For each angular coordinate less than $\frac{w}{2}$, we check all pixels that have a radial coordinate in the possible range $R \in [(r - i\% \frac{w}{2}), (r + i\% \frac{w}{2})]$, where i is the current angular coordinate. When a pixel meets the aforementioned conditions, its value is added to the minimum cost at the previous angular coordinate and a radial coordinate in the range of $r \pm 1$ (cf. line 7), where r is the radial coordinate of the current pixel. The distance from the current radial coordinate to the minimum cost at the previous angular coordinate is stored in an index matrix K (cf. line 8). For the right part of the image I, the same procedure is performed with the constraint that the range of R is dependent on the integer variable i_2 instead of the current angular coordinate i. i_2 decrements from $w - 1$ as i increments from $\frac{w}{2}$ to $w-1$ (cf. lines 11- 17). After computing the cost matrix $C_{(r,\theta)}$, we update the minimal circular shortest path cost $\mathrm{Cost}_{\mathrm{min}}$ by first checking if the minimal cost at the last angular coordinate is less than the global variable $\mathrm{Cost}_{\mathrm{min}}$ (cf. line 18). If this is the case, the current radial coordinate is stored as an index for the global minimal path $\mathrm{Index}_{\mathrm{min}}$ (cf. line 19). The minimum cost value is also stored by updating the global variable $\mathrm{Cost}_{\mathrm{min}}$ (cf. line 20). Now that all the radial levels are examined in the polar image I, we created a rule to extract the circular minimal path by back-tracing the pixels of this path starting from the index of the global minima $\mathrm{Index}_{\mathrm{min}}$ that will be the radial index for the last contour point (cf. line 23). As an additional rule at each current angular coordinate, the radial index of the circular shortest path is stored by checking the radial index at the next angular coordinate and adds to it the distance stored in the index matrix K (cf. line 25).

A sample application of the circular shotest path algorithm is applied on the *S. cerevisiae* yeast cell shown previously in Fig. 3.5(a). The result of this application is illustrated in Fig. 3.9(a), where the final extracted contour is displayed as *cyan* points on the resampled image. The backprojection of the contour on the original image is shown in Fig. 3.9(b).

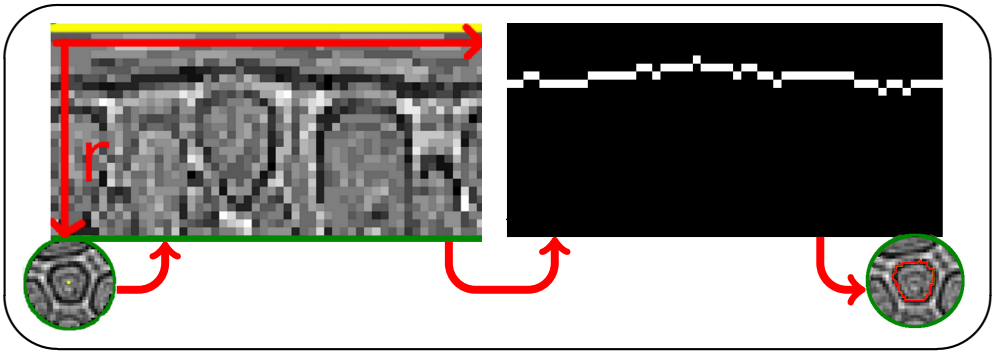
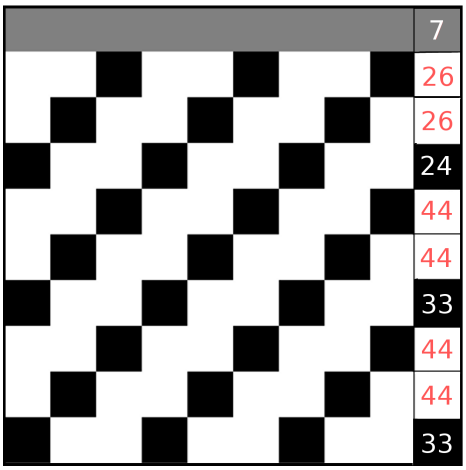


Figure 3.7: *Hough transform and grey-weighted distance transform to extract cell contours.*

3.5 Contour Optimization

Our objective is to find reliable methods for the extraction of an accurate contour delineating the object. We have shown to be successful in achieving this objective through dynamic programming (DP) [Ger86], where we used *Hough* transform and set a cost matrix where we applied a minimal path algorithm to estimate the contour location [Tle14]. We described two minimal path methods, consequently we have two segmentation algorithms. The first is the



(a) Pathological cost function from [App03]. Last column shows CSP cost for each row.

(b) Shortest path at first row with a total weight=7.

(c) Shortest path at 3rd row with a total weight=24.

(d) Shortest path at last row with a total weight=33.

Figure 3.8: *Finding the Circular Shortest Path (CSP). Black, grey and white squares represent pixel intensity values of 0, 1 and 11 respectively (similar to [App03]). Pixels marked with X are not evaluated. The path with the minimum weight is the global CSP.*

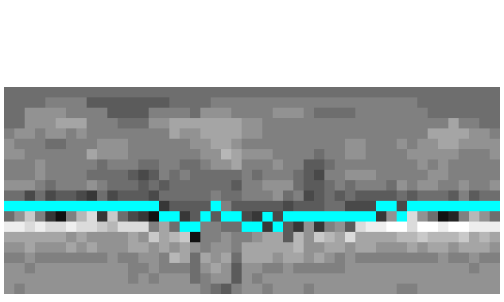
Hough transfrom followed by grey-weighted distance transform method; i.e. *HDT*, and the second is the *Hough* transform followed by circular shortest path method; i.e. *HCSP*. However, since microscope imaging can be very delicate and edges have an inherently fuzzy nature, a refinement of the initial estimate is sometimes required. For example, the objects to be segmented have contours that are described by more than one pixel thick contour line. This is typically true for the yeast image dataset that we show as a case study in this section. This problem is observed when high resolution images using high numerical aperture (NA) lenses are used with image sizes larger than 512x512 pixels. Further analysis of the results demonstrates that the circular minimal path method has a bias toward the inner part of the object as a result of the fact that some inner pixel values are lower than the edge pixel values. We state that, in these cases, the minimal path is no longer the ultimate valid representation of the contour. In this section, we use a yeast dataset for which this is typically the case.

As a possible solution to this problem, we developed an expansion algorithm that directly evaluates the polar image I used to extract the minimal path. We refer to I as a polar image, which is a polar representation of each object O_i in the original image. With our additional heuristic, we are able to achieve the necessary refinement of contours. This contributes to the accomplishment of precise measurements of the objects so that machine learning techniques can better recognize the subtle patterns within the data.

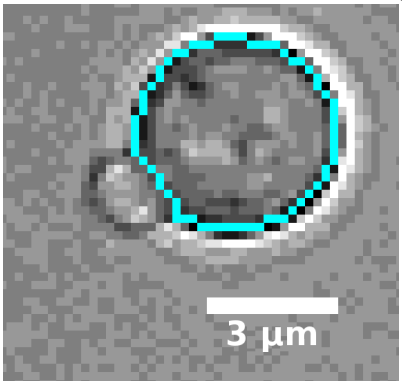
In the following subsections, we provide a background and subsequently the control parameters are described. In the last subsection we discuss the contour expansion (*CE*) algorithm.

3.5.1 Background

Our goal is to obtain the exact contours of ovoid objects; here applied to yeast cells. The initial contour detected by *HCSP* is expanded out toward the background surrounding the object until it meets certain criteria. The expansion is realized using dynamic programming in the polar image I. Starting from the initial contour detected by *HCSP* [Tle14], the contour is expanded in such a way that $r' \geq r$, where r' is the radial coordinate of the new contour pixel in image I and r is the initial radial coordinate. Every contour pixel considers a number of factors that decides



(a) Extracted minimal path from the resampled image.



(b) Contour backprojected on the original image.

Figure 3.9: *Sample application of circular shortest path on S. cerevisiae cell.*

whether it should be expanded to the next level or if it is a *final* contour point. To decide if the current contour point is a *final* contour point, we have identified three parameters; i.e. the *resistance*, *limit* and *convergence*; these parameters are discussed hereafter.

3.5.2 Control parameters

The *resistance* parameter res is a float variable in the range $res \in [0,1]$. It is the key player in this algorithm. It decides if the current contour point is a *final* point by evaluating the value of the point at the next radial coordinate. When the value of res is 0 the expansion is blocked, while at the value of 1 the pixels in I would not block the expansion. At a value of 0.5 we threshold the image I at the mean of its histogram and consequently the background pixels above that threshold value will block the expansion of the contour. At res values below and above 0.5, we would threshold I according to Eq. 3.8 and Eq. 3.9 respectively.

$$t = h_{min} + 2(h_{\mu} - h_{min}) \times res \quad \forall\ res\ \in\ (0, 0.5] \tag{3.8}$$

$$t = h_{\mu} + 2(h_{max} - h_{\mu}) \times (res - 0.5) \quad \forall\ res\ \in\ (0.5, 1] \tag{3.9}$$

where h_{μ} is the mean of the histogram of I, h_{min} is the minimum value of I histogram, and h_{max} is its maximum histogram value, res is the value of *resistance*, and t is the returned threshold value. The pseudocode of the threshold process is presented in **Box 3.2**.

The *limit* control parameter is an integer variable specifying a constraint of the maximally allowed radial coordinate for a new contour point. Specifically, it is the number of radial positions after the largest radial coordinate in the initial contour. After that level, the contour expansion is blocked.

Input : A polar image I of size $w \times l$
An integer $resistance \in [0,1]$
Output : A binary image B_i of size $w \times l$

```
if resistance ≤ 0.5 then
    t ← h_min + 2(h_μ − h_min) × resistance
end
else
    t ← h_μ + 2(h_max − h_μ) × (resistance − 0.5)
end
B_i ← threshold I at t.
return B_i
```

Box 3.2: *Polar image threshold based on* $resistance$.

The *convergence* parameter is an integer variable specifying a constraint of the maximum number of iterations before the expansion process is terminated. Subsequently, the contour at the last iteration is considered as a final contour. In general, the contour is saturated and reaches its final status after a few iterations. This will be shown in an example in the following sub-section.

3.5.3 The Expansion Algorithm

The expansion algorithm is applied after the initial contour processing. The flowchart in Fig. 3.10 explains how the expansion algorithm works and how the pixels are processed in the polar image I. The pseudocode of this algorithm is referred to in **Box 3.3**. The algorithm iterates until the *convergence* criterion is met, or until, in an iteration, no point is expanded anymore (cf. line 2). In each iteration, the *expanded* boolean variable is set to false (cf. line 3), and all the contour points are processed (cf. line 4). If any point is expanded at that iteration, then *expanded* variable is set to true, and the algorithm keeps running as long as the *convergence* criteria is not reached.

The processing procedure of the contour points at each iteration is shown in a separate block in the flowchart (*Process Contour Pixels*), and its pseudocode is presented separately as **Box 3.4**. This procedure evaluates each point in the contour and considers whether to expand it to the next radial coordinate or not. As long as the current contour point is not marked as a *final* contour point (cf. line 2), it is marked as *final* if the *limit* parameter is reached (cf. line 4); otherwise, the previous and the next neighbour contour points are checked if any of them is at a radial position lower than that of the current point (cf. line 7), in which case the point is skipped in this iteration. If none of these neighbour contour points is at a previous radial coordinate, these neighbour points are checked whether they are both marked as *final* points, forcing the current point to be *final* contour point as well (cf. line 8). If both neighbours are not marked as *final* points, the current point is marked as *final* if the point at the next radial position is blocked by the *resistance* parameter (cf. line 11). When the *resistance* control parameter does not block the expansion, the point is expanded to the next radial position unless one of the neighbours is marked as *final* and the current point has higher radial coordinate than that neighbour (cf. line 13), then the current point is marked as a *final* contour point (cf. line 14). Other than that, the point is expanded to the next radial position (cf. line 16), and normally this expanded contour point will be marked as a *final* contour point in the next iteration or when the *convergence* criteria is met before the next iteration.

Figure 3.11 illustrates an example showing the extracted initial estimate by *HCSP* algorithm and the final expanded contour extracted by *CE* method in a pathological cost image function. The image function shown in Fig. 3.11(a) is a sample of a polar image I having ten angular and ten radial coordinates. The first and last angular coordinates θ_A and θ_B are actually for the same point since as the path from θ_A to θ_B is a circular path. The *black* squares in I represent pixels with an intensity value of 0, the *grey* squares represent pixels with an intensity value of 1, and the *white* squares represent pixels with an intensity value of 11. In Fig. 3.11(b), the initial contour points extracted by applying *HCSP* are displayed by *green* spots.

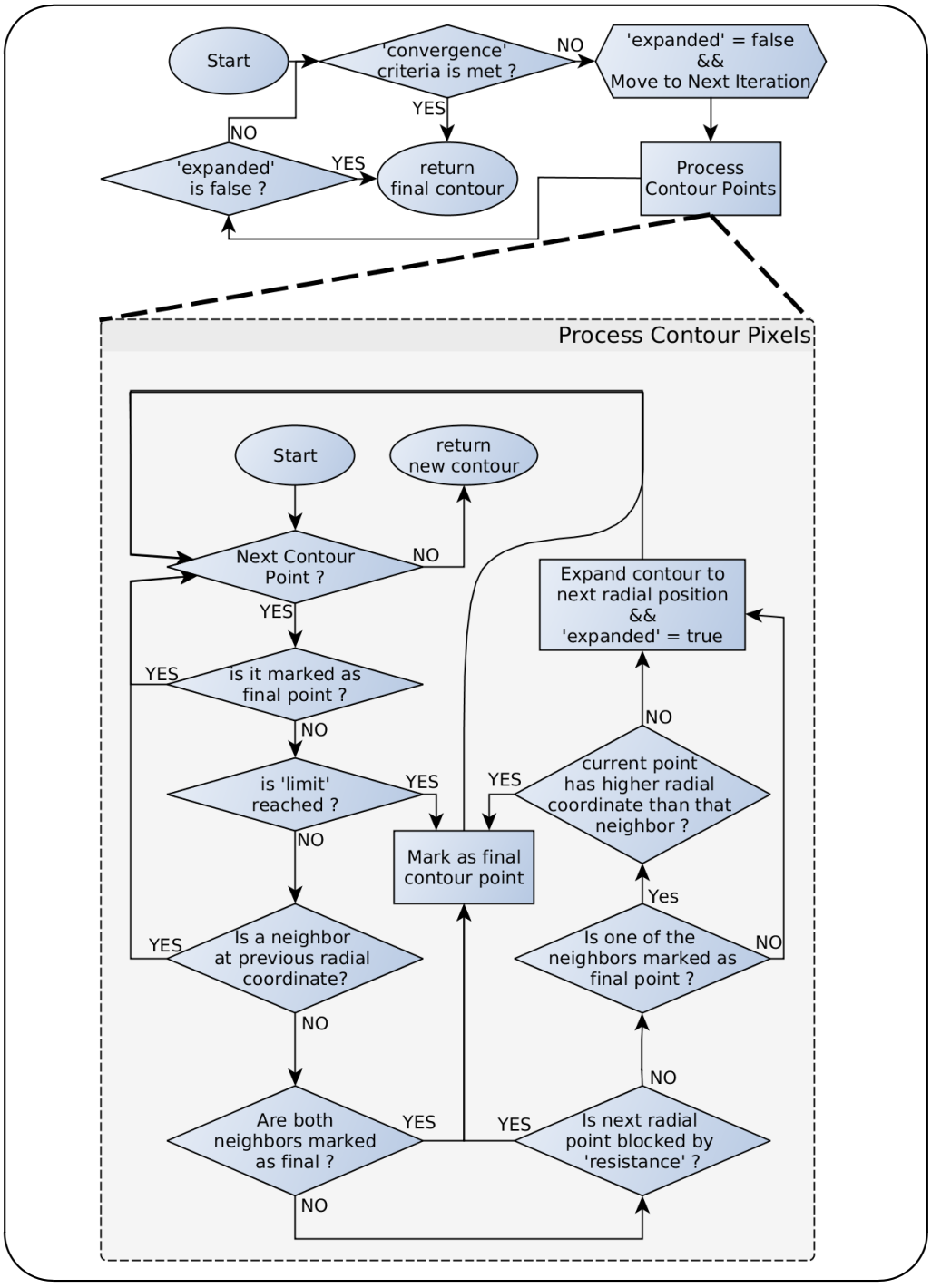


Figure 3.10: *Flow chart explaining the expansion algorithm.*

To illustrate how the *CE* algorithm works, we apply contour expansion on image I with the parameters set as follows: *resistance = 0.5*, *limit = 1*, and *convergence = 5*. After the first iteration, the contour evolves to take the position illustrated in Fig. 3.11(c). The points that were not considered for expansion in this iteration are left as *green* spots. Those expanded are displayed by *blue* spots and the *red* spot is the contour point marked as final. The point at the eighth angular coordinate is marked as *final* because the point at the next radial position was blocked by the *resistance* parameter. Figure 3.11(d) shows the contour points after the second iteration. Note here that the point at the seventh angular coordinate is marked as *final* because both its connected neighbours are marked *final* as well. Figure 3.11(e) shows the path after the third iteration. In the fourth iteration, the first four points and consequently the last contour point are marked as *final* points by the *limit* parameter (cf. Fig. 3.11(f)). In addition, the points at the fifth and ninth angular coordinates were marked *final* because their connected neighbours are marked *final* contour points as well. Since no contour point was expanded in this iteration, the contour expansion process is terminated at this point, even before the *convergence* criteria is met.

To demonstrate the application of this algorithm on a real sample, a *RoI* containing two yeast cells is depicted in Fig. 3.12. The initial estimate is shown as well as the expanded contour for two yeast cells from a sample image acquired by a Zeiss LSM5 confocal microscope [Inc12]. The *yellow* contours represent the initial estimate extracted after the application of *Hough* transform and circular minimal path algorithm, while the *red* contours represent the final expanded version of the contours acquired ensuing the application of our new contour

Input : A binary image B_i of size $w \times l$.
Integer array CsP of initial contour, of size w.
Boolean array isFinal of size w.
Boolean variable expanded.
Integer limit $\in [0, l]$.
Integer convergence ≥ 0.
Output : Integer array finalContour of size w.

```
1  initialize isFinal and expanded to false, i ← 0.
2  while (expanded ≠ false && i < convergence) do
3  |  i ← i + 1
4  |  expanded ← false
5  |  CsP ← ProcessContour(CsP) (Procedure in Box 3.4)
6  end
   /* Current contour is final                                    */
7  for each element p in CsP do
8  |  finalContour[p] ← CsP[p].
9  endfor
10 return finalContour
```

Box 3.3: *Expand Contour Algorithm.*

expansion algorithm. Figure 3.12(a) shows the contours on the bright-field channel used for initial estimation and expansion of the contours, while Fig. 3.12(b) shows the same contours superimposed on the fluorescent counterpart of the cells acquired as a second image channel by the same microscope. The latter channel is used to obtain the meaningful measurements and pattern recognition regarding protein levels and gene expression in the yeast cells. It is clear from this image, how much of the green signal would not be included in the cell measurement if contour expansion would not have been applied. This is even more quintessential when measuring proteins that are bound to the membrane of the cell.

3.6 Validation

In order to validate the new segmentation methods described in this chapter, i.e. the *Hough* transform followed by minimal path algorithms, we create an artificial dataset of 1000 images representing the yeast cells in an approach similar to that implemented in [Yan12]. Such dataset

```
  /* Process contour pixels                                          */
 1 for each point c in CsP with radial coordinate r do
 2     if isFinal[c] ← false then
 3         if r ← limit then
 4             isFinal[c] ← true
 5         else
 6             if (CsP[c − 1] ≠ (r − 1)) and (CsP[c + 1] ≠ (r − 1)) then
 7                 if CsP[c − 1] ← true and CsP[c + 1] ← true then
 8                     isFinal[c] ← true
 9                 else
10                     if (B_i[c, r + 1] ← Background pixel then
11                         isFinal[c] ← true
12                     else
13                         if (isFinal[c − 1] ← true and c has higher r than c − 1) or
                            isFinal[c + 1] ← true and c has higher r than c + 1 then
14                             isFinal[c] ← true
15                         else
16                             Expand c to next radial position (r + 1).
17                             expanded ← true
18                         end
19                     end
20                 end
21             end
22         end
23     end
24 end
```

Box 3.4: *Process Contour Points Procedure.*

has a random number of elliptical shapes ranging from $2\,\mathrm{to}\,100$ per image with a random Gaussian values for the background. The Gaussian values for the background is created to resemble that of the real bright-field microscope images. The eccentricity of the ellipses ranges randomly with $\frac{\mathrm{semi-minor}}{\mathrm{semi-major}}$ factor between $0.75\,\mathrm{and}\,1$ to resemble the actual cell shape. As a case study, an experiment was performed and 8 images were acquired. These images depict 110 cells in total. Our methods were applied on this dataset.

To evaluate our segmentation methods, we considered well-known metrics; i.e. the Pratt Figure of Merit (*FoM*) and *F1-measure*. Moreover, the tuning of parameters for the different segmentation methods and inspection of the segmented results were performed using *Paramorama* [Pre11], a prototype developed to optimize parameters based on parameter sampling and interactive visual exploration. *Paramorama* is implemented as a plug-in for the *CellProfiler* biomedical image analysis framework [Car06].

The following subsections provide definitions for the Pratt *FoM* and that of *F1-Measure*. The last subsection shows the result comparing *HCSP* and *HDT* with state of the art segmentation package, i.e. *CellStat*. In addition, it shows the result comparing the extension algorithm *HCSP-CE* with *CellStat*, *CellSerpent* and the initial *HCSP*.

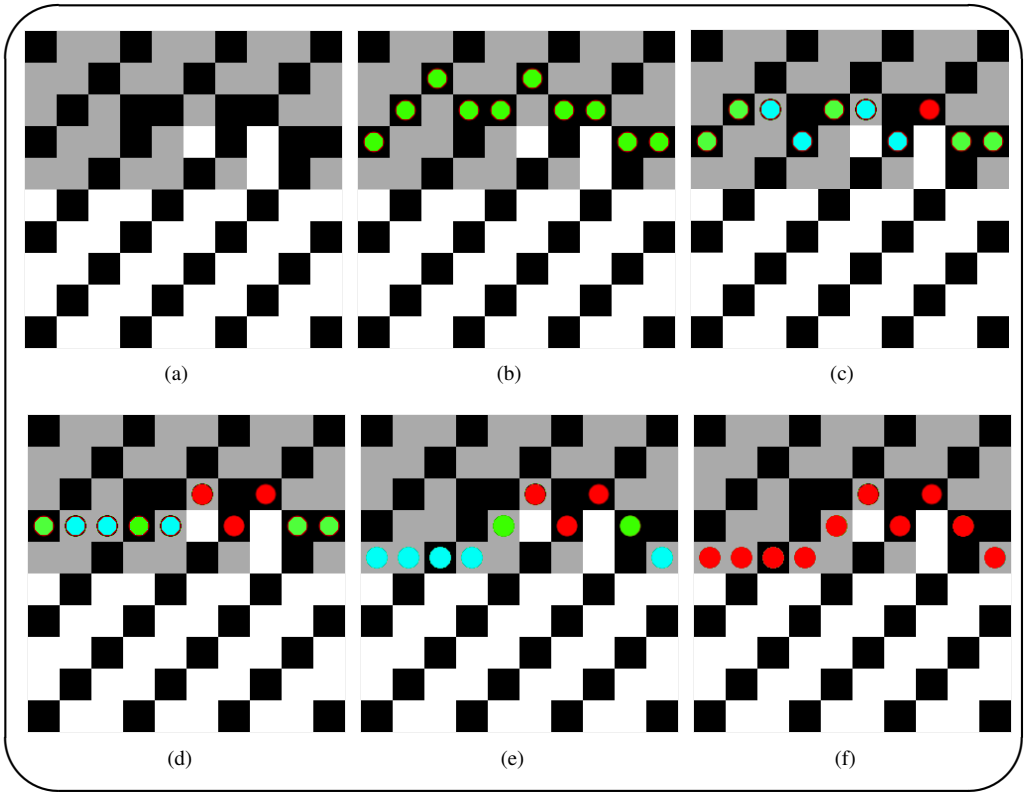
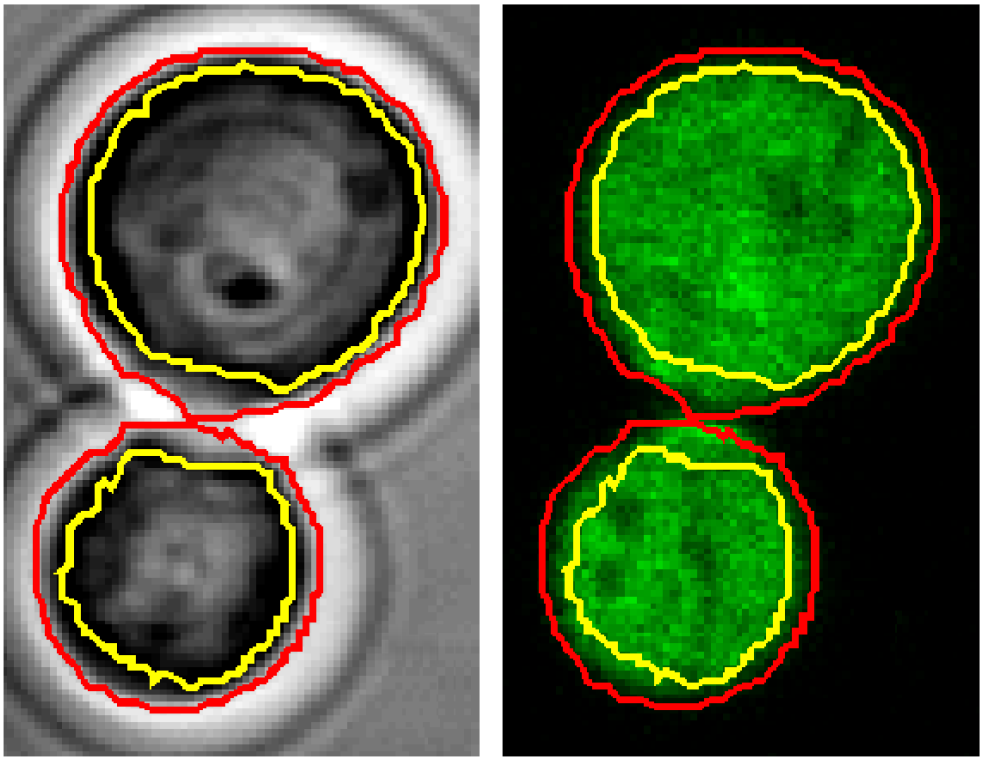


Figure 3.11: *Expanding the Initial estimate of the contour. Black, grey and white squares represent pixels with intensity values of* 0, 1 *and* 11 *respectively.*

3.6.1 Pratt Score

The Pratt Figure of Merit (*FoM*) is considered because it has been reported as a very good criterion to detect the differences in segmentation results [Cha06]. Pratt's *FoM* provides a quantitative comparison of the results of the different contour detection methods by measuring the deviation of the output contours from a ground-truth. The ground-truth dataset were separately delineated using our dedicated structural annotation software (*TDR*) with a digitizer tablet (*WACOM, Cintiq LCD-tablet*) [Ver02]. The Pratt measure is computed according to Eq. 3.10.

$$P = \frac{1}{\max(I_A, I_I)} \sum_{i=1}^{I_A} \frac{1}{1 + \alpha.d^2(i)} \tag{3.10}$$



(a) Bright-Field view of yeast cells.

(b) Contours superimposed on a fluorescent channel of the same yeast cells.

Figure 3.12: *The expanded contour is depicted in red, while the yellow contour is the initial estimate extracted by the Hough transform and Circular Shortest Path algorithm (HCSP).*

where I_A being the number of detected edge points, I_I the edge points in the ideal ground-truth image, α a scaling factor or an empirical calibration constant set to the optimal value established by Pratt, $\alpha = 1 \div 9$ [Abd79], $d(i)$ is the distance between the detected edge point and the nearest edge point in the ideal ground-truth [The10]. Pratt's *FoM* is an indicator of the quality of edge and reflects the overall behaviour of the distances between the edges. Pratt's *FoM* is a relative measure, which varies in the range [0,1], where 1 represents the optimal value, i.e., the detected edges coincide with the ground truth [Boa09].

3.6.2 F1-Measure

The *F1-measure*, also known as *F1-score*, is a measure of accuracy used in many domains including the comparison of segmentation algorithms. It considers the precision and recall in its computation [Hou13]. The precision is defined as the number of correctly detected objects divided by the number of detected objects by the segmentation method. The recall is defined as the number of correctly detected objects divided by the number of actual objects. We considered a yeast cell object correctly detected if the centroid of its detected binary mask also exists as a foreground pixel in the ground-truth binary mask. The F1-score can also be used to measure the accuracy of the extracted contour by considering the binary mask of the individual detected object. In this case, the precision is defined as the correctly detected pixels in the binary mask divided by the number of detected pixels. The recall is then defined as the number of correctly detected pixels divided by number of actual pixels in the ground-truth mask. When this later definition is used we will refer to the F1-score as $F1_p$-score to indicate its computation at the pixel level. The F1-score is computed according to Eq. 3.11.

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{3.11}$$

3.6.3 Results

In this subsection, we present the validation results of the methods we prooposed earlier including the *HDT* and *HCSP* methods proposed in Section 3.2 and Section 3.4; in addition to the optimization method applied after *HCSP* as discussed in Section 3.5, i.e. the *HCSP-CE* method.

HDT and *HCSP* validation

The *HDT* method uses *Hough* transform followed by the grey-weighted distance transform as the minimal path algorithm. The *HCSP* method uses *Hough* transform followed by our proposed circular shortest path algorithm. To evaluate these methods, we compare them with state of the art yeast segmentation software, i.e. *CellStat*.

CellStat also uses dynamic programming and cost functions and it is implemented in *Matlab*. It is designed to segment bright-field images of *S. cerevisiae*, but it has a constraint on the cells to be detected, as they can not be clumped within other cells [Kva08].

	Detection Rate	F1-score
HCSP	97.23%	0.972
HDT	97.23%	0.972
CellStat	94.97%	0.950

Table 3.1: *Detection Rates and F1-measure on artificial dataset.*

In Table 3.1 the detection rates and *F1-scores* [Hou13] are shown after applying the three methods on the artificial dataset. In the last step of the contour extraction process, the shape is checked for its circularity value. If this value is less than an empirically determined value, i.e. 0.614, the object is rejected. Checking the detection rate on the whole artificial dataset using *CellStat* is not feasible. Therefore, we manually checked on a sample drawn from the complete set. To verify whether this sample is representative for the whole dataset, an unpaired student t-test was performed on the results obtained from the *HCSP* method. The p-value of this test is 0.95.

As a case study, an experiment with *S. cerevisiae* yeast cells was performed with 8 representative images. In total, these images contain 110 cells. Our methods were applied on this dataset (both *HDT* and *HCSP*). The results are compared to that obtained from *CellStat* [Kva08]. The detection rates and F1 – measures are shown in Table 3.2.

	Detected	Correct	Precision	F1
HCSP	115	106	0.964	0.942
HDT	120	104	0.945	0.904
CellStat	112	101	0.918	0.910

Table 3.2: *Detection Rates and F1-measure on yeast images.*

For the comparison of the accuracy of the detected contours, Pratt's *FoM* is selected. Table 3.3 shows the results of the comparison between the proposed methods and that of the *CellStat* software on our *S. cerevisiae* dataset. The detection rate outperforms that of *CellStat* and the number of detected contours that gains better Pratt score (referred to as Wins in Table 3.3) is also higher than that of *CellStat* making it a suitable segmentation method to be used in analysis of yeast cells. Hence, this method is successfully used in our yeast analysis platform proposed in Chapter 2.

	HCSP	*CellStat*	*HDT*	*CellStat*
Detected	106	101	104	101
Precision	96.4%	91.8%	94.5%	91.8%
Wins	60	50	57	51

Table 3.3: *Comparing Detected Contours of yeast images.*

HCSP-CE validation

In order to validate the new *HCSP-CE* method, its performance is compared to the previous *HCSP*, *CellSerpent* and *CellStat*.

Figure 3.13 shows the number of detected cells that gained a higher Pratt score using *HCSP* method followed by contour expansion (*CE*) algorithm (*HSCP-CE*). These cells were obtained from a dataset of 312 *Saccharomyces cerevisiae* yeast cells distributed over 14 bright-field

images acquired by a *Zeiss LSM 5 Exciter-AxioImager M1* confocal microscope. Compared to *CellSerpent*, the *HCSP-CE* method has 163 cells with better Pratt scores, referred to as wins, while *CellSerpent* has only 138 wins. In addition, comparison with *CellStat* leads to 284 wins of cells with higher Pratt scores in the *HCSP-CE* method, with only 16 wins for *CellStat*. This result is expected as the method implemented in *CellStat* looks for the minimal path as the final contour of yeast cells in a similar way to that of *HCSP*, which scored 18 wins. On the other hand, *CellSerpent* performs better than *CellStat* because its method is based on Active Contour models for cell detection. However, Active Contour models couldn't outperform our algorithm.

In Table 3.4 a comparison between the detection rates of the four used segmentation methods on the dataset of 312 cells is listed. In Table 3.5, the accuracy of the detected contours is compared between the four methods using average Pratt scores and $F1_p$-scores for the individual objects.

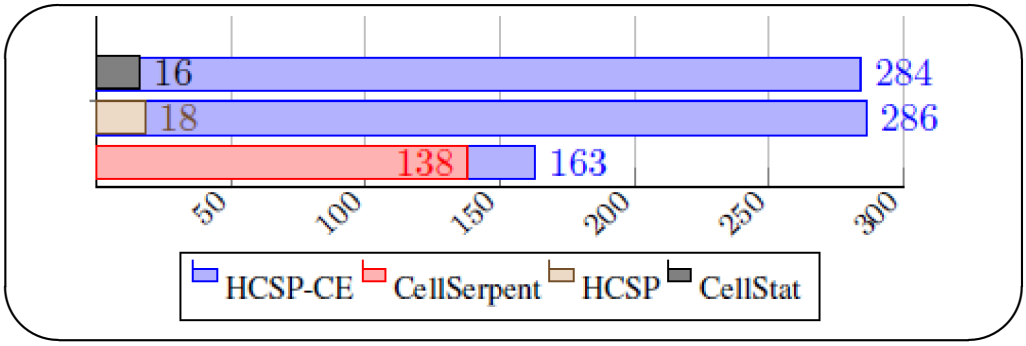


Figure 3.13: *Number of cells with higher Pratt score (wins).*

.

	HCSP-CE	CellSerpent	HCSP	CellStat
Detected	**322**	359	**322**	231
Correct	296	**302**	296	181
Precision	**0.92**	0.84	**0.92**	0.78
Recall	0.95	**0.98**	0.95	0.58
F1	**0.93**	0.90	**0.93**	0.67

Table 3.4: *F1-measure and average Pratt Score of different segmentation methods on a yeast dataset.*

.

	HCSP-CE	CellSerpent	HCSP	CellStat
$F1_p$	**0.92**	0.89	0.76	0.47
Pratt	**0.58**	0.55	0.19	0.15

Table 3.5: *Accuracy of detected contours.*

The *HCSP-CE* segmentation method has an overall highest scores with an F1-score of 0.93, $F1_p$ of 0.92 and an average Pratt score of 0.58. The following method is *CellSerpent* with an F1-measure of 0.90, $F1_p$ of 0.89 and average Pratt of 0.55. *CellStat* follows with an F1-measure of 0.67, $F1_p$ of 0.47 and average Pratt of 0.15. *HCSP* scores similar F1-measure to that of *HCSP-CE* as they both use the same underlying method for detection, however the Pratt score of *HCSP* is 0.19 and the $F1_p$ is 0.76, which is noticeably worse than *HCSP-CE* and close to that of *CellStat* as they both look for minimal paths for contour extraction.

3.6.4 Robustness under Noise

The *HCSP-CE* method is further validated for its robustness under noise. We evaluate how the method performs under various noise levels. First we selected a random sample and prepared a groundtruth. The ground-truth is separately delineated using our dedicated structural annotation software (*TDR*) with a digitizer tablet (*WACOM, Cintiq LCD-tablet*) [Ver02]. Subsequently, we determined the optimal parameter values to perform the segmentation. This step involves an initial heuristic determination of parameter values, then quantitatively evaluate the segmentation accuracy after iterating each of the parameters individually while fixing values of all the other parameters. This procedure is repeated until parameter values are saturated. Figure 3.14 shows the optimal saturated parameter set values after three repetitions.

In order to evaluate the segmentation accuracy, we used F1-scores for the detected cells per image in addition to F1-scores of the extracted mask per cell as described in Section 3.6.2. We used the average of the two F1 scores to determine the optimal values.

In order to generate noisy images, we first determined three main types of noise. Gaussian noise, Poisson noise and Speckle noise. The principal sources of Gaussian noise in digital images arise during acquisition e.g. sensor noise caused by poor illumination and/or high temperature, and/or transmission [Cat13]. Poisson noise is known also as Shot noise, it occurs in photon counting in optical devices. Poisson noise describes the fluctuations of the number of photons detected (or simply counted in the abstract) due to their occurrence independent of each other [Bla00]. Speckle noise, also known as salt and pepper kind of noise, is caused usually by the scattering of light from a highly coherent source, such as a laser, and generates a random-intensity distribution of light that gives the image a granular appearance [fre16].

We vary the signal to noise ratio *SNR* for each type of noise, and perform the segmentation on the sample using the optimal parameters mentioned in Fig 3.14. Image noise can often be described by an additive noise model, where the resulted image $f(i,j)$ is the sum of the true image $s(i,j)$ and the noise image $n(i,j)$ as depicted in Eq. 3.12.

$$f(i,j) = s(i,j) + n(i,j) \tag{3.12}$$

We considered the noise $n(i,j)$ as zero-mean and we describe it by its variance σ_n^2. The impact of the noise on the image is described by the signal to noise ratio (SNR), which is given by Eq. 3.13, where σ_s^2 and σ_n^2 are the variances of the true image and the noise image respectively.

$$SNR = \frac{\sigma_s}{\sigma_n} = \sqrt{\frac{\sigma_s^2}{\sigma_n^2} - 1} \tag{3.13}$$

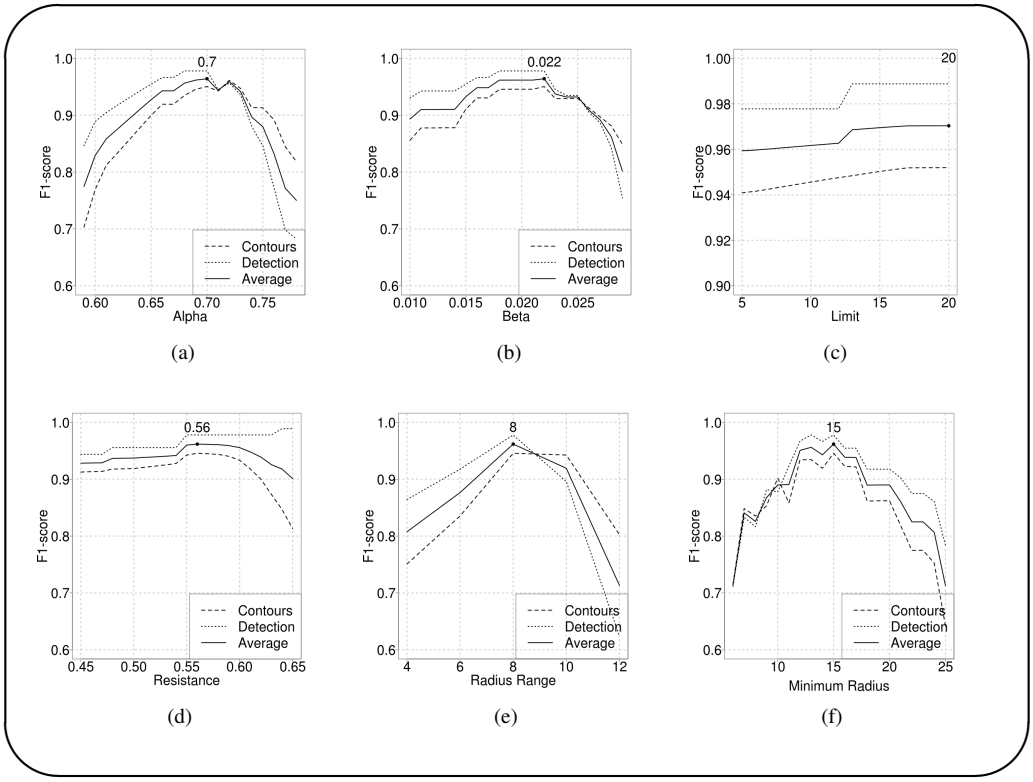


Figure 3.14: *Optimal Segmentation Parameters. (a) - The optimal value of the alpha parameter for the sample dataset is* 0.7*. This parameter is part of the thresholding equation described in 3.2.2. (b) - Beta parameter associated with alpha has an optimal value at* 0.022*. (c) - The limit parameter used in the optimization of contours was described in 3.5.2. It has an optimal value at 20. (d) - The resistance is also a contour optimization parameter described in 3.5.2. Its optimal value is 0.56. (e) - The radius range used in filling the Hough accumulator as described in 3.2.1. The optimal value is 8. (f) - The minimum radius to consider during the acuumualtor filling has an optimum at 15.*

Figure 3.15 shows the segmentation performance under the noisy conditions with an *SNR* ranging from 1 to 35. For Gaussian noise, the final average F1-score is 0.926 and the standard deviation is 0.009. For Poisson noise the final average F1-score is 0.934 and the standard deviation is 0.012. For Speckle noise the final average F1-score is 0.928 and the standard deviation is 0.010. These numbers along with the visual inspection of the plots show that the *HCSP-CE* segmentation method is very robust under various noise conditions.

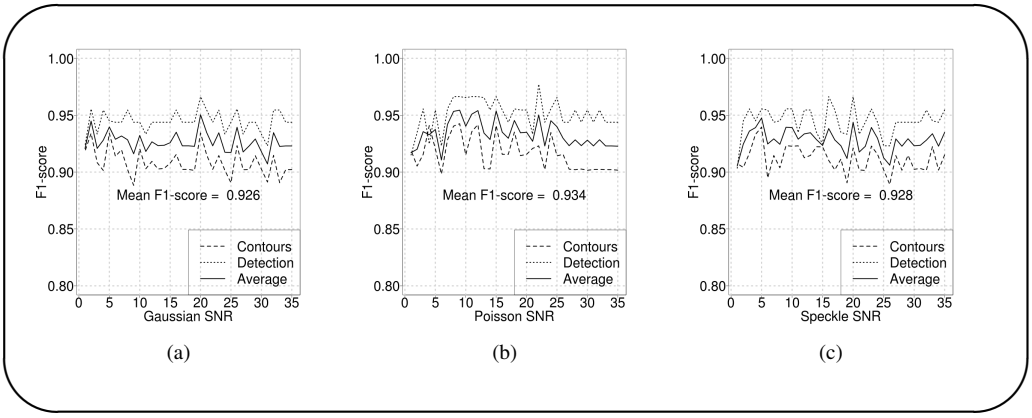


Figure 3.15: *F1-score at various noise levels. (a) - Assessing segmentation performance under various levels of Gaussian noise. (b) - Assessing segmentation performance under various levels of Poisson noise. (c) - Assessing segmentation performance under various levels of Speckle noise.*

3.7 Conclusion

In this chapter, we proposed new methods to detect oval-shaped objects, such as yeast cells in bright-field microscopy images. We started by defining *Hough* transform and introducing our proposed equations to threshold the *Hough* accumulator array. Subsequently, we introduced the definition of polar transformation, the concept of polar coordinate system and how transformation from cartesian to polar system is achieved. We also demonstrated the polar representation of a sample yeast image. The polar representation of an object image is required to extract the minimal path corresponding to the contour of the object. The minimal path concept is also discussed in this chapter and two algorithms to extract this minimal path are introduced. These algorithms are the *Grey-weighted distance transform* and *Circular shortest path*. Hence, we proposed two segmentation methods that uses *Hough* transform and minimal path algorithms; i.e. *HDT* and *HCSP* methods. Subsequently, we introduced our additional expansion algorithm that operates on the polar representation of object images; i.e. *HCSP-CE*. To validate the *HDT* and *HCSP* new methods, they have been applied on two datasets along with state of the art software packages common to yeast biology. The first dataset is a large artificial dataset of ovoid shapes, and the second is a actual small dataset of *S. cerevisiae* yeast cells. For evaluation, we considered two metrics namely the Pratt Figure of Merit (*FoM*) and *F1-Measure*. The results show higher Pratt score and *F1-Measure* compared to the method from the *CellStat* software. On the other hand, the *HSCP-CE* is applied to a different dataset of 312 *S. cerevisiae* yeast cells distributed over 14 bright-field images. *HCSP-CE* was compared to methods from *CellStat* and *CellSerpent*.

4

Machine Learning to Identify Subtle Patterns and Improve Object Recognition

In this chapter, we choose some sophisticated features and use them in a machine learning approach to improve the recognition of objects and to identify subtle patterns between different objects. As a case study, we applied this machine learning approach on S. cerevisiae yeast cells. In the first experiment, we identify different characteristics between two different cell groups cultivated under different stress levels. In the second experiment, we improve the recognition of S. cerevisiae yeast cells by classifying the intact cells from artefacts, i.e. debris and dead cells existing in the culture medium. Since the dataset generated in both experiments are imbalanced, we were careful in the evaluation of classifiers built by the machine learning process. We considered sampling of data, scaling, feature selection, cross-validation and evaluation metrics.

This chapter is based on the following publications:

- Mohamed S. Tleis and Fons J. Verbeek. "Machine Learning approach to discriminate *Saccharomyces cerevisiae* yeast cells using sophisticated image features." Journal of Integrative Bioinformatics, 12(3):276, 2015.
- Mohamed S. Tleis and Fons J. Verbeek. "Machine Learning approach to segment *Saccharomyces cerevisiae* yeast cells." In third International Conference on Advances in Biomedical Engineering (ICABME). pp. 278-281. IEEE, 2015.

4.1 Introduction

SUBTLE Patterns such as different characteristics that are hard to notice from the basic measurment and statistical analysis of data, are not easily extracted from these measurement or basic statistical analysis. These subtle patterns are especially intrinsic in high throughput screening (HTS) where thousands of images are analysed. Moreover, when biologists study an organism in two different conditions, it can be not possible to know if different groups of that organism have different characteristics. In addition, recognition of objects in images prior to their analysis is quintessential in order to generate meaningful conclusions. Thus, the need of an automatic system to extract those hidden features and to recognize objects.

As an application, we take the *S. cerevisiae* as a case study. The segmentation module of the automated analysis platform, i.e. *YeastAnalysis*, discussed in Chapter 2, provides a segmentation of the cells obtained from the microscope images, while the measurement module measures various features of the segmented cells. The data analysis part analyzes the measurement and report relevant statistics about the different cell groups. When biologists construct and study different yeast cell strains cultivated in different media, it can be not possible to know if the different cell groups have different characteristics for the same expressed proteins. Thus the need for an automatic system to identify any charactersitic difference.

Our novel segmentation algorithm in *YeastAnalysis* segments intact cells as well as artefacts. These artefacts can be dead cells or debris existing in the culture medium. Artefacts have negative effects on the analysis result of the experiments. These artefacts can be interactively excluded after the segmentation using *YeastAnalysis* platform; however, in high throughput screening (HTS) where thousands of images are analysed, manual checking is not feasible anymore. Thus, the need of an automatic system to exclude such artefacts prior to analysis.

The research question in this work is what machine learning approach can be used to improve object recognition and identify subtle patterns in a dataset of sophisticated features? and the sub-question is whether the combination of various feature sets can improve the performance of this machine learning approach? The features (attributes) for training the machine learning system were selected in a way to offer a more sophisticated description of the intensity and morphology characteristics of the objects. The machine learning workflow is depicted in Fig. 4.1.

Applying the extraction techniques that we will use is mentioned in the first section. We created two different dataset on the basis of sophisticated features. Since our dataset is imbalanced with a different ratio for different cell groups, we were careful in our evaluation to choose the classifier system that can classify well the majority as well as the minority classes. Therefore, we considered sampling and normalization techniques prior to feature selection algorithms. A number of linear and non-linear classifiers were evaluated to select the best model that can classify the instances in the first dataspace into cells belonging to a different group, i.e. different strain, or cultured in a different medium, and the best model that can classify the instances in the second dataset into intact cells or artefacts.

The result shows that many classifiers have an excellent performance after data preprocessing. Consequently, two classification models are built. As a result, the first model allows to identify the different characteristics of objects, while the second model has raised the performance level of the measurement and consequently the pattern recognition system.

In the next section, we will discuss the sophisticated feature sets we presented then we discuss the classification process for our dataset.

4.2 Sophisticated Features

In Chapter 2 we presented our developed platform to perform image analysis on *S. cerevisiae* yeast cells. This platform can segment the cells and measure a range of features and textures for each individual cell obtained from two channel images acquired by a laser scanning confocal microscope (CLSM). Figure 4.2 shows a sample two-channels image of *S. cerevisiae* yeast cells. The first channel in Fig. 4.2(a) is a bright-field channel depicting yeast cell structures. The second overlaid channel in Fig. 4.2(b) is a fluorescent channel of the *BMH1* gene expressed Bmh1 protein binding with GFP protein cultivated in low NaCl medium. The yellow contours surrounding the cells in Fig. 4.2 are the results of our segmentation algorithm [Tle14]. Such segmentation of individual cells enables us to introduce sophisticated features and texture measurement to describe the characteristics of cell morphology and intensity distribution of individual cells. These features and texture measurement (cf. Section 1.3.2) facilitate the analysis and discrimination of different yeast cells.

Next, we define the concept of extraction techniques. Subsequently, we select and discuss well-known extraction techniques that are useful in analyzing biological images. The benefits of combining these extraction techniques will be revealed in the results. Before that, we discuss building a machine learning system using the feature set extracted using these techniques explained in this section.

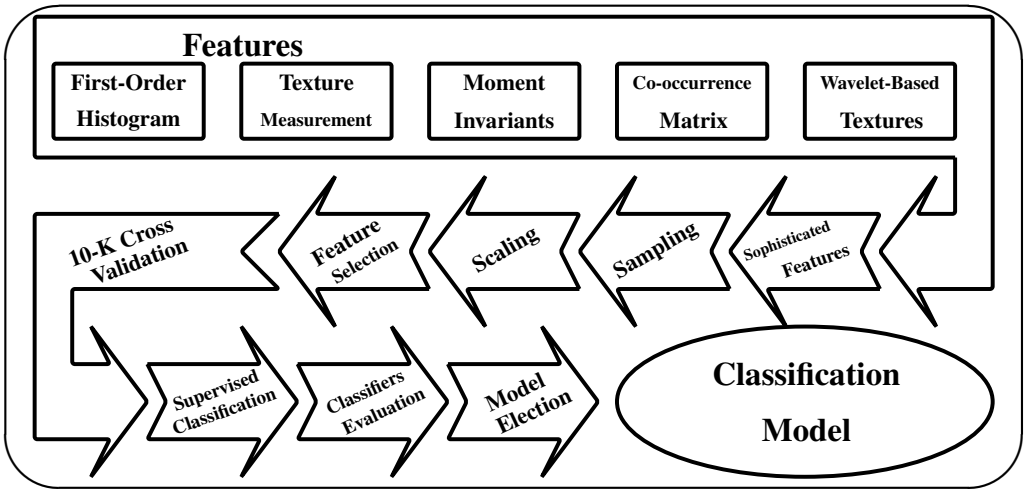


Figure 4.1: *Workflow for building the classification models.*

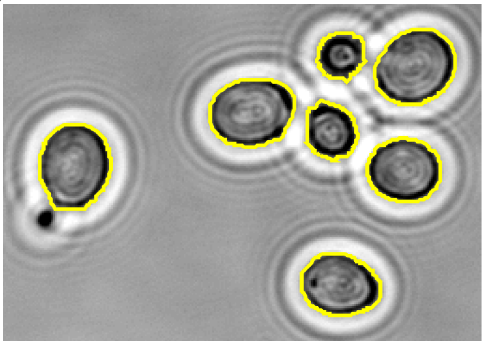
4.2.1 Feature Extraction Techniques

Feature extraction is one very important area of application in image processing, in which algorithms are used to detect and isolate various desired portions or shapes (features) of a digitized image. Feature extraction is defined as locating those pixels in an image that have some distinctive characteristics [Gub09]. In our research, we considered the most known feature extraction techniques in image analysis. These techniques are classified into histogram based features and the moment invariants derived from them, co-occurrence matrix based features, and multi-scale features [Mat98]. Hereafter, we start discussing the histogram based features then the moment invariants derived from them. Subsequently, we explain the additional features derived from the co-occurrence matrix. We complete this sub-section by highlighting on the multi-scale features and specifically wavelet-based texture features.

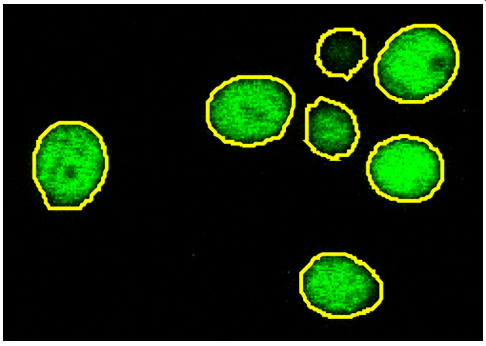
First order statistics based features

The histogram of intensity levels is a concise and simple summary of the statistical information contained in the image. Calculation of the grey-level histogram involves single pixel. Thus the histogram contains the first-order statistical information about the image, i.e. the region of interest (RoI) of cell objects. Different useful image features are worked out from the histogram to quantitatively describe the first-order statistical properties of the cells. In this study, we considered many basic shape descriptors based on the first order statistics, and the most relevant texture features among those originally proposed by Haralick et al. [Har79, Gon08]. These features were already discussed in Section 2.4 and a list of those features are listed in Table 2.1 and Table 2.2.

Another way to characterize the texture is by deriving moment invariants from the first order statistical information in the histogram [Pap65]. The following sub-section discusses these moment invariant features.



(a) Bright-field channel depicting structure of cells.



(b) Fluorescent channel depicting *Bmh1-GFP* gene expressed protein.

Figure 4.2: *S. cerevisiae Yeast Cell Images.*

Moment invariant features

Recognition of visual patterns independent of position, size, and orientation in the visual field has been a goal of much recent research. To achieve maximum utility and flexibility, it would be useful if the extraction technique is insensitive to variations in shape and provide improved performance with repeated trials. This property is known in moment invariant techniques. An image moment is a certain particular weighted average, i.e. moment, of the pixel intensities in an image or a function of such moments chosen to have some attractive property or interpretation. Traditionally, moment invariants are computed based on the information provided by both the shape boundary and its interior region. Image moments are useful after segmentation to describe cell characteristics that uniquely describe the shape of that cell. Low order moments are used to derive simple properties including area, total intensity, centroid, skewness, kurtosis and information about the cell's orientation. Moment invariant values are invariant to translation, scale and rotation of the cell [Key01]. For example, the first rotation invariant moment is known to be analogous to the moment of inertia around the image's centroid, where the pixel intensities are analogous to physical density. The seventh one, is skew invariant, which enables it to distinguish mirror images of otherwise identical images [Hu62]. Although wavelet transform (discussed in section 4.2.1) is scale invariant, it is not in all cases translation or rotation invariant [Jaf05], this is an additional advantage of moment invariants in our measurements. Moment Invariants have been frequently used as features for image processing, remote sensing, shape recognition and classification. They showed to be fairly reliable at distinguishing certain classes of topographic objects [Key01] as well as in many other applications [Mat98]. In yeast studies, the first and second moment invariants were the top predictors to classify virulent from non virulent cells [van07].

Hu [Hu62] defines seven shape descriptor values derived from central moments through order three that are independent to object translation, scale and orientation. Translation invariance is achieved by computing moments that are normalized with respect to the center of gravity so that the center of mass of the distribution is at the origin (central moments). Size invariant moments are derived from algebraic invariants but these can be shown to be the result of a simple size normalization. From the second and third order values of the normalized central moments a set of seven invariant moments can be computed which are independent of rotation. The set of seven moment invariants proposed by Hu are widely known [Gon08], and hence we adopted them in our study. We define Hu's set as in Eq. 4.1, where Φ_1 and Φ_2 are invariants based on second order moments, while Φ_3 ... Φ_7 are invariants based on third order moment.

$$hu = \{\Phi_1, \Phi_2, \Phi_3, \Phi_4, \Phi_5, \Phi_6, \Phi_7\} \tag{4.1}$$

The effectiveness of moment invariants will increase when fused with the results of other techniques [Key01]. In this research we fuse them with wavelet-based texture measurements to get the best from these approaches in the classification step required to discriminate between various cell conditions. The co-occurrence matrix based features and the wavelet-based texture features are discussed in the following sub-sections.

Co-occurrence matrix based features

The simplicity of texture attributes can not completely characterize texture of the cells. Studies state that similar textures agree in their second-order statistics [Mat98] and hence textures can be discriminated if they differ in their second-order statistics. Therefore one of the major statistical methods used in texture analysis is the one based on the definition of the joint probability distribution of pairs of pixels. Methods based on second-order statistics, i.e. statistics given by pairs of pixels, have been shown to achieve good discrimination rates in texture classification [Wes76], and considered to be important in automated image analysis [Nie81]. The second-order statistical features for texture analysis are derived from the co-occurrence matrix [Har79]. They were demonstrated to feature a potential for effective texture discrimination in biomedical images as well [Ler93].

The second-order histogram is defined as the co-occurrence matrix $C_{d\theta}(i,j)$. When divided by the total number of neighbouring pixels $R(d,\theta)$ in the image, this matrix becomes the estimate of the joint probability $P_{d\theta}(i,j)$ of two pixels, a distance d apart along a given direction θ having particular "co-occurring" values i and j [Mat98]. Formally, for image $f(x,y)$ with a set of L discrete intensity levels, the matrix $C_{d\theta}(i,j)$ is defined such that its $(i,j)^{th}$ entry is equal to the number of times that: $f(x1,y1) = i$ and $f(x2,y2) = j$, where $(x2,y2) = (x1,y1) + (d\cos\theta, d\sin\theta)$. This yields a square matrix of dimension equal to the number of intensity levels in the image, for each distance d and orientation θ. Most relevant co-occurrence matrix derived features used for the purpose of texture discrimination are the angular second moment, correlation, inertia, absolute value, entropy and maximum probability [Har79, Lah09]. Table 4.1 lists descriptions for these features. In this research we calculated the measures at distance $d = 1$ for horizontal, vertical and diagonal orientations at $\theta = 0°, 90°, 45°$ and $135°$ to achieve a degree of rotational invariance.

Multi-scale features and Wavelet-based texture measurement

Various methods adopted for calculating multi-scale features. The most commonly used are the Wigner distributions, Gabor functions and wavelet transforms. These transform methods of texture analysis represent an image in a space whose coordinate system has an interpretation that is closely related to the characteristics of a texture, i.e. frequency or size. Wigner distribution are found to possess interference terms between different components of a signal. These interference terms lead to wrong signal interpretation. Gabor filters are criticized for their non-orthogonality that result in redundant features at different scales or channel. On the other hand, the wavelet transform, being a linear operation, does not produce interference terms nor redundant features. For this reason, our interest is in the application of the wavelet transform to texture analysis. Discrete wavelet transform (DWT) derived features appear to be a suitable tool to be used for digital image texture analysis, because they allow analysis of images at various levels of resolution. The DWT provides powerful insight into an image's spatial and frequency characteristics [Koc01]. Moreover, it has shown to be an efficient descriptor for phenotyping [Cao14]. In general, wavelet analysis is highly capable of revealing aspects of data such as trends, breakdown points, discontinuities in higher derivatives and self similarity [Shr13].

Table 4.1: Co-occurrence-matrix based features

Features	Description
Angular second moment (ASM)	Also known as uniformity and it is a measure of cell homogeneity. The maximum value is achieved when all the elements in the co-occurrence matrix are equal. $$ASM = \sum_{i=0}^{L-1}\sum_{j=0}^{L-1}[p(i,j)]^2 \quad (4.2)$$
Correlation	The correlation measures the dependencies between the yeast cell image pixels. μ_x, μ_y and σ_x, σ_y denote the mean and standard deviations of the row and column sums of the co-occurrence matrix respectively. $$Correlation = \sum_{i=0}^{L-1}\sum_{j=0}^{L-1}\frac{ijp(i,j) - \mu_x\mu_y}{\sigma_x\sigma_y} \quad (4.3)$$
Inertia	Inertia is also known as contrast. It is calculated by squaring the subtraction of the examined pixel values. Thus, the minimum value is when the pixels have the same grey-level value, and the maximum is achieved when squaring the subtraction of L and 1. $$Inertia = \sum_{i=0}^{L-1}\sum_{j=0}^{L-1}(i-j)^2p(i,j) \quad (4.4)$$
Absolute value	It calculates the absolute value of the subtraction of the examined pixel values. Hence it ranges between 0 and L. $$Absolute\ value = \sum_{i=0}^{L-1}\sum_{j=0}^{L-1}|i-j|p(i,j) \quad (4.5)$$
Inverse difference	It has relatively high value when the high value in the co-occurrence matrix are near the main diagonal, where the difference $(i-j)$ is smaller there. $$Inverse\ difference = \sum_{i=0}^{L-1}\sum_{j=0}^{L-1}\frac{p(i,j)}{1+(i-j)^2}. \quad (4.6)$$
Entropy	The entropy measures the complexity of the texture. It is a measure of randomness, achieving its highest value when the elements in the matrix are maximally random. $$entropy = -\sum_{i=0}^{L-1}\sum_{j=0}^{L-1}p(i,j)log_2[p(i,j)]. \quad (4.7)$$
Maximum probability	Gives an indication of the strongest response in the co-occurrence matrix. $$Maximum\ probability = max_{i,j}p(i,j) \quad (4.8)$$

Approximations and details are the most important terms in wavelet analysis. The approximations are the high-scale, low-frequency components of the image signal, while the decomposition process in the wavelet transform generates the coefficient matrices for the level-one approximation and horizontal, vertical and diagonal details. In this study, we include a bi-orthogonal wavelet in which texture details are derived from the three different directions on the same scale as the original image [Cao14]. The derived wavelet-based textures are the same as those derived for the original cells and listed in Table 2.2.

In this section we highlight on the powerful set of extraction techniques that we haven chosen in our research to measure individual yeast cells. The extracted sophisticated features and textures are quintessential for pattern recognition and identification of subtle patterns residing within our measured data. In the following section, we will discuss the application of machine learning using these sophisticated features. Then we show the results that reveal the advantages of these features.

4.3 Constructing the Classification Model

In this section, we will address the creation of our datasets and discuss the sampling techniques applied on these datasets to study whether such techniques have any impact on the classification results. Similarly we discuss the various normalization schemes used. Subsequently we highlight the used feature selection algorithms. Thereafter we provide detail about the evaluation metrics. The last part is about the classifiers considered in the comparison. The results are discussed later in the next section.

4.3.1 Imbalanced Dataset and Sampling Techniques

In order to generate our dataset for the purpose of object pattern discrimination and object recognition, we extracted features to describe the object characteristics and morophology in a more sophisticated way.

In our two experiments performed on *S. cerevisiae* yeast cells, we generate two datasets S_1 and S_2. S_1 consisting of 1440 yeast cell instances belonging to two major classes, one representing cells showing genes tagged with (GFP) reporter and expressing 14-3-3 proteins. The first class of cells is cultivated under low osmotic stress level, i.e. *non-NaCl* medium, and the other representing same cell strains under a high stress level, i.e. *0.5M-NaCl* medium. The second dataset, i.e. S_2, consists of 1380 segmented objects belonging to two major classes, one representing intact yeast cells and the other representing artefacts in the microscope images. After segmentation, all cells are measured for the features mentioned previously in Section 4.2. Each instance I in these two datasets is mapped to one element of the set $(\mathtt{p}, \mathtt{n})$ of positive and negative class labels representing the two different cell classes of low vs. high stress levels in S_1 and intact vs. artefacts in S_2 respectively. We need to build a classification model to map from instances to predicted classes. Given a classifier and a test set of instances, a two-by-two confusion matrix, also known as contingency table is constructed to represent the dispositions of the set of instances. This matrix forms the basis for many metrics we used to evaluate the classifiers [Faw06].

Our dataset S_1 and S_2 are considered imbalanced since they exhibit an unequal distribution between their positive class, i.e. yeast cells under low osmotic stress in S_1 or intact cells in S_2, and their negative class, i.e. yeast cells under high stress in S_1 or artefacts in S_2 classes. The ratio of positive to negative classes being around $2.7 : 1$ in S_1 and $8 : 1$ in S_2. Hence, in this domain, we require a classifier that will provide high accuracy for the negative minority class without severely jeopardizing the accuracy of the positive majority class. In conventional evaluation practice, singular assessment criteria, e.g. the overall accuracy or error rate are used. These criteria do not provide adequate information in the case of imbalanced learning because most standard algorithms assume or expect balanced class distributions or equal misclassification costs.

The problem of learning from imbalanced data is a relatively new challenge that has attracted growing attention and has become apparent in all kinds of datasets.

The induction rules that describe the minority concepts are often fewer and weaker than those of majority concepts, since the minority class is often both outnumbered and under-represented. Successive partitioning of the dataspace results in fewer and fewer observations of minority class examples resulting in fewer rules describing minority concepts and successively weaker confidence estimates. In addition, concepts that have dependencies on different feature space conjunctions can go unlearned by the sparseness introduced through partitioning. The application of sampling techniques has shown to improve classifier accuracy. Therefore, we considered three well-known sampling techniques, namely under-sampling, over-sampling and Synthetic Minority Oversampling technique (*SMOTE*).

We define subsets $S_{min} \subset S_d$ and $S_{maj} \subset S_d$ where S_d refers to either dataset S_1 or S_2, S_{min} is the set of minority class instances in S_d, and S_{maj} is the set of majority class instances in S_d, so that $S_{min} \cap S_{maj} = \{\phi\}$ and $S_{min} \cup S_{maj} = \{S_d\}$.

Random under-sampling removes data from the original data set. In particular, we randomly select a set of majority class instances in S_{maj} and remove these instances from S_d so that $|S_d| = |S_{min}| + |S_{maj}| - |E|$, where E represents the set removed by the sampling procedure. Under-sampling readily gives us a simple method for adjusting the balance of the original data set S_d; however, removing instances from the majority class may cause the classifier to miss important concepts pertaining to the majority class.

In oversampling, multiple instances of certain examples become "tied" since it simply appends replicated data to the original dataset, leading to overfitting. In particular, overfitting in oversampling occurs when classifiers produce multiple clauses in a rule for multiple copies of the sample example which causes the rule to become too specific; although the training accuracy will be high in this scenario, the classification performance on the unseen testing data is generally far worse.

The synthetic minority oversampling technique (*SMOTE*), on the other hand, is a powerful method that has shown a great deal of success in various applications. It creates artificial data based on the feature space similarities between existing minority instances. Specifically, for subset S_{min}, consider the K-nearest neighbours for each instance $x_i \in S_{min}$ for some specified integer K; the K-nearest neighbours are defined as the K elements of S_{min} whose euclidean distance between itself and x_i under consideration exhibits the smallest magnitude along the

n-dimensions of feature space *X*. To create a synthetic sample, we randomly select one of the K-nearest neighbours, then multiply the corresponding feature vector difference with a random number $\in [0,1]$, and finally add this vector to the minority instance $x_i \in S_{min}$ as depicted in Eq. 4.9, where $\hat{x_i} \in S_{min}$ is one of the K-nearest neighbours for x_i, and $\delta \in [0,1]$ is a random number. Therefore, the resulting synthetic instance according to Eq. 4.9 is a point along the line segment joining x_i under consideration and the randomly selected K-nearest neighbour $\hat{x_i}$ [He09]. Applying *SMOTE* to balance both of our datasets increased the accuracy of the classification as we will see in the next section. Moreover, the classification accuracy obtained after applying *SMOTE* significantly outperformed both the under-sampling and over-sampling methods.

$$x_{new} = x_i + (\hat{x_i} - x_i) \times \delta, \quad \text{where } \delta \in [0,1] \tag{4.9}$$

4.3.2 Feature Scaling or Normalization

Feature scaling is a method used to standardize the range of independent variables or features of data. In data processing, it is also known as data normalization and is generally performed during the data preprocessing step [Ver14]. Since the range of values of the raw data in our dataset varies, some machine learning algorithms will not work properly without normalization. For example in distance classifiers, the range of all features should be normalized so that each feature contributes approximately proportionately to the final distance. In our work we considered two well-known normalization techniques which are unit-length normalization (UL) and zero-mean and unit-variance normalization (MV) [Štr09]. Feature normalization techniques represent a vital part in building the classification model. They aim at normalizing the individual components of the extracted feature vectors in such a way that the resulting vectors are better suited for classification.

Unit length normalization (UL) scales all of the components x_i (i = 1, 2,. . . .,d) of vector x of all instances in our dataset S_d in accordance with the expression in Eq. 4.10 to produce the normalized feature vector x^*, where ||.|| denotes the norm operator, and $x_i{}^*$ stands for the i^{th} component of the normalized vector x^*.

$$x_i{}^* = \frac{x_i}{||x||}, i = 1,2,...d, \tag{4.10}$$

The zero-mean and unit-variance normalization is defined in Eq. 4.11, where μ denotes the mean value of the feature vector x and σ represents its standard deviation. The MV technique transforms the feature vector x to a random variable with a mean value of zero and variance of one. It is assumed the individual components of the feature vector are normally distributed.

$$x_i^* = \frac{(x_i - \mu)}{\sigma}, i = 1, 2, \ldots, d, \tag{4.11}$$

Applying both normalization schemes to scale our dataset improved the accuracy of the classification. More details are illustrated in the next section. The zero-mean and unit-variance normalization outperforms the unit length normalization in both experiment dataset, hence, we adopted the zero-mean unit variance as a normalization scheme for our datasets.

4.3.3 Feature Selection

In machine learning and statistics, dimensionality reduction or dimension reduction is the process of reducing the number of random variables under consideration, and can be divided into feature selection and feature extraction. It is used to assist in the data analysis process.

Feature selection also known as variable or attribute selection, can be defined as a process that chooses a minimum subset of M features from the original set of N features, so that the feature space is optimally reduced according to a certain evaluation criterion. The reduced feature space are the most important parameters which help in predicting the outcome. Finding the best feature subset is usually intractable and many problems related to feature selection have been shown to NP-hard. The objective of feature or variable selection is three-fold:

- improving the prediction performance of the classifiers on the testing dataset.
- providing faster and more cost effective classifiers.
- providing a better understanding of the underlying process that generated the data.

Selecting the most relevant features is suboptimal for building our predictor, particularly if the features are redundant or irrelevant. Redundant features are those that provide no more information than the currently selected features, and irrelevant features provide no useful information in any context. For our data, we considered two well-known feature selection algorithms which are the Information Gain (IG) method and the Correlation Feature Selection (CFS).

Next to feature selection is feature extraction, which transforms the data in the high-dimensional space to a space of fewer dimensions. We considered the main linear technique for feature extraction, i.e. the principal component analysis (PCA), which performs a linear

mapping of the data to a lower-dimensional space in such a way that the variance of the data in the low-dimensional representation is maximized.

4.3.4 Building a Classifier

The prediction problem for our case study is to predict whether the cells are cultivated under high osmotic stress vs. those under low osmotic stress in dataset S_1 or those intact cells vs. artefacts in dataset S_2. The classification models are given a training dataset of known ground-truth data, and a testing dataset of unknown first-seen data against which the models are tested. In order to limit problems like over-fitting and give an insight on how the model will generalize to an independent dataset, the widely used 10-fold cross validation is considered [Kim11]. Over-fitting occurs when the classification model does not fit this validation data as well as it fits the training data. Cross validation is important in protecting against testing hypotheses suggested by the data, also known as Type III errors [Don13a]. Its advantage is that all observations are used for both training and validation, and each observation is used for validation exactly once.

4.3.5 Evaluation metrics, *ROC* and *AUC*

A receiver operating characteristic (*ROC*) curve is a two dimensional graphical plot that illustrates the performance of a binary classifier system, which predicts a two-class problem in which the outcomes are labelled either as positive (p) or negative (n) [Lak11]. The curve is created by plotting the true positive rate (TPR) on the Y axis against the false positive rate (FPR) on the X-axis at various threshold settings. TPR is also known as sensitivity in biomedical informatics, or recall in machine learning [Li14]. TPR defines how many correct positive results occur among all positive samples available during the test. On the other hand, FPR also known in biomedical informatics as (1-Specificity) defines how many incorrectly labeled positive results occur among all negative samples available during the test [Sen13]. Each prediction result or instance of a confusion matrix represents one point in the *ROC* space [Ras13].

The *ROC* graphs are useful for evaluating our classifiers and visualizing their performance. *ROC* graphs have been successfully used in medical decision making, and in recent years, they are popular in machine learning and data mining research, due to the realization that scalar measures such as simple classification accuracy, error rate or error cost are often poor metrics for measuring performance. *ROC* graphs have properties that make them especially useful for domains with skewed class distribution and unequal classification error costs. These characteristics have become increasingly important as research continues into the areas of cost-sensitive learning and learning in the presence of unbalanced classes [Faw06]. *ROC* analysis provides tools to select possibly optimal models [Bes10]. Classifiers appearing on the left-hand side of an *ROC* graph, near the X axis, may be thought of as “conservative”: they make positive classifications only with strong evidence so they make few false positive errors, but they often have low true positive rates as well. Classifiers on the upper right-hand side of an *ROC* graph may be thought of as “liberal”: they make positive classification with weak evidence so they classify nearly all positives correctly, but they often have high false positive rates. Many

real world domains are dominated by large numbers of negative instances, so performance in the far left-hand side of the *ROC* graph comes more interesting. We have used the area under the *ROC* curve (AUC), also known as c-statistic [Her11], which is usually interpreted according to the following ratings: $\chi = 1$, perfect; $1 > \chi \geq 0.9$, excellent; $0.9 > \chi \geq 0.8$, good; $0.8 > \chi \geq 0.7$, fair; $0.7 > \chi \geq 0.6$, poor; $0.6 > \chi \geq 0.5$, fail (random guessing for *AUC*); $\chi < 0.5$, unacceptable [Tsa12]. *AUC* is a common statistic most often used for model comparison in the machine learning community [Chi13]. Despite its popularity, some machine learning researches show that the *AUC* is quite noisy as a classification measure [Han10], and has some other significant problems in model comparison [Lob08, Han09]. Therefore, we also considered the standard accuracy metric, which is widely used to get an additional insight into the results. More importantly, we considered the *AUC* for the minority class ($\mathrm{A}_{\mathrm{min}}$) as well. $\mathrm{A}_{\mathrm{min}}$ reveals the dangers of blindly looking into the *AUC* alone in the raw dataset classifiers evaluation. However, the *AUC* becomes a safe metric when the data is preprocessed with a sampling technique as the result will show in the following section.

4.3.6 Classifiers Evaluated

In this work, 23 different linear and non-linear classification systems were evaluated on our dataset. These systems including the popular predictors such as decision trees, naive Bayes, least-square linear predictors, and support vector mahcines. A complete list of these models along with short descriptions are shown in Table 4.2. We applied a supervised classification on our two datasets S_1 and S_2, with a 10 fold cross validation.

4.4 Results

Using our dataset with the presented sophisticated features, we evaluated the classifiers by considering the area under *ROC* curve (*AUC*) as our main metric in addition to the minority class $\mathrm{A}_{\mathrm{min}}$ and *ACC* (accuracy) of each classification system. For dataset S_1, we listed in Table 4.3 the *AUC*, $\mathrm{A}_{\mathrm{min}}$ and *ACC* scores for each classifier under the best sampling, normalization and feature selection algorithms. For dataset S_2, we only listed the top ten classifiers in Table 4.4. Figures 4.3 and 4.4 shows the noticeable difference between *AUC* and $\mathrm{A}_{\mathrm{min}}$. This is the rational behind evaluating the *AUC* for the minority class as well as that of the complete dataset.

For our case study, most classifiers have optimal results with the *SMOTE* sampling technique and the *MV* normalization scheme. The Feature Selection with the best results was the *IG* method. It is clear from the results that a number of classifiers were able to excellently discriminate between the two cell groups in both datasets S_1 and S_2 with an *AUC*, $\mathrm{A}_{\mathrm{min}}$ and *ACC* metrics above 0.9. The following subsections reveal the power of sampling, the effect of normalization and feature selection, in addition to the power of the discriminators based on our feature space.

Table 4.2: Classification Algorithms evaluated in this study

Classifier	Description
C4.5	A decision tree classifier. At each node of the tree, C4.5 chooses the attribute that most effectively splits its set of samples into subsets enriched in one class or the other [Qui93].
Adaptive Boosting (AdaB)	A machine learning meta-algorithm used in conjunction with a weak learner (Decision Tree) to improve its performance [Fre96].
PART	Builds a partial C4.5 decision tree in each iteration and makes the "best" leaf into a rule [Fra98].
Decision Table Majority (DTM)	A decision table with a default rule mapping to the majority class. It has a set of features (schema) and a set of labelled instances (body) [Koh95].
Decision Stump (DSmp)	A model consisting of a one-level decision tree. i.e, it is a decision tree with one internal node (the root) which is immediately connected to the terminal nodes (its leaves) [Iba92].
One Rule (OneR)	OneR generates one rule for each predictor in the data, then selects the rule with the smallest total error as its "one rule" [Hol93].
JRip	JRip implements a propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction (RIPPER) [Coh95].
Bayes Network (BNet)	Bayes Network learning represents the dataset variables via a directed acyclic graph (DAG) based on probability theory [Fri97].
K-Nearest Neighbour (IBK)	Predicts the class of the single nearest training instance for each test instance [Aha91].
Locally Weighted Learning (LWL)	Uses an instance-based algorithm (Decision Stump) to assign instance weights [Atk96].
LogitBoost (ALR)	Performs additive logistic regression on the base learner (Decision Stump) [Fri98].
Random Committee (RCom)	Builds an ensemble of randomization base classifiers (Random Tree). The final prediction is a straight average of the predictions generated by the individual base classifiers..
Random Subspace (RSub)	Constructs a decision tree based classifier (REPTree) with multiple trees constructed in randomly chosen subspaces [Ho98].
Hoeffding Tree (VFDT)	An incremental decision tree induction algorithm capable of learning from massive data. It assumes that the distribution of variables does not change over time [Hul01].
Logistic Model Tree (LMT)	A logistic model tree basically consists of a standard decision tree structure with logistic regression functions at the leaves [Lan05].
REPTree	Fast decision tree learner. Builds a decision/regression tree using information gain/variance and prunes it using reduced-error pruning.
Random Forest (RFor)	Constructs a forest of random decision trees at training time and outputting the mode class (classification) or mean prediction (regression) of the individual trees [Bre01].
Random Tree (RTre)	Constructs a tree with randomly chosen attributes at each node.
Logistic (Log)	Building and using a multinomial logistic regression model with a ridge estimator [Ces92].
Stochastic Gradient Descent (SGD)	Implements stochastic gradient descent for learning various linear models (binary SVM, binary logistic regression, squared loss, Huber loss and epsilon-insensitive loss).
Sequential Minimal Optimization (SMO)	Sequential minimal optimization algorithm for training a support vector classifier [Pla98].
SimpleLogistic (SLog)	Classifier for building linear logistic regression models. LogitBoost with simple regression functions as base learners is used for fitting the logistic models [Lan05].
Voted Perceptron (VPer)	Based on a linear predictor function combining a set of weights with the feature vector, and a transformation of online learning, in that it processes elements one at a time [Fre98].

4.4.1 Power of Sampling

The first obvious fact from the result in Table 4.3 and 4.4 is the power of data sampling on the classifiers performance. *SMOTE* has not failed to improve the overall classification within all the algorithms tested, unlike the under-sampling and over-sampling methods. This makes *SMOTE* an excellent choice for our data. Moreover, *SMOTE* addresses the information loss of the under-sampling and the over-representation issue of the over-sampling methods.

After applying *SMOTE* on dataset S_1, the *AUC* was improved in average by .025, and the average accuracy increased slightly by 0.007. However, the strongly significant different is shown when considering the *AUC* for the minority class (A_{min}) where the average improvement was increased by .139 per classifier from an average of .708 to .847. This demonstrates the reason why accuracy is not a sufficient measure in our imbalanced dataset. Generally, power of sampling is more conspicuous in *PART, C4.5, JRIP, SMO* and *VFDT* classifiers (cf. Fig. 4.5).

On dataset S_2, the power of sampling is more obvious since S_2 exhibits a higher ratio between majority and minority class instances. After applying *SMOTE* on dataset S_2, the *AUC* was improved in average by .127, but the average accuracy decreased slightly by 1.1%.

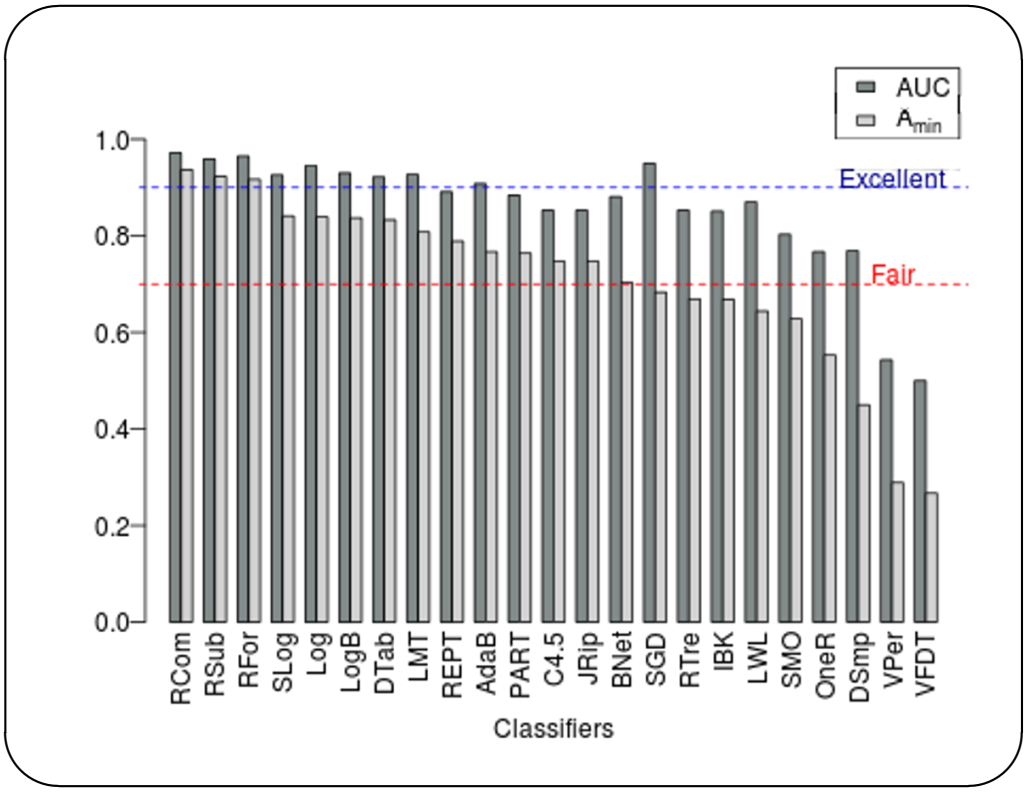


Figure 4.3: *AUC and* A_{min} *of classifiers for raw dataset* S_1.

However, the strongly significant different is shown when considering the *AUC* for the minority class ($A_{\min}$) where the average improvement was increased by .369 per classifier from an average of .565 (poor classification) to .935 (excellent classification) (cf. Fig. 4.6). This also reveals the reason why accuracy is not a sufficient measure in our imbalanced dataset.

4.4.2 The effect of Normalization and Feature Selection

On dataset S_1, Normalization showed an extra average improvement of .015 and .013 for the *AUC* and $A_{\min}$ respectively, and improved accuracy by 1.2% over the original dataset. As can be noticed from Fig. 4.7 and 4.8, the most significant difference of normalization was shown in *VPer* and *VFDT* classifiers. Similarly for dataset S_2, Normalization showed an extra average improvement of .007 and .009 for the *AUC* and $A_{\min}$ respectively, and improved accuracy by 5.8% over the sampled dataset.

The best feature selection algorithm, i.e. *IG*, showed no significant change in the averages of *AUC* and $A_{\min}$ for both dataset. On dataset S_2, the *AUC* was .949 and $A_{\min}$ .944. The accuracy metric has a slight extra increase of .03% up to a final average accuracy of 92.2%.

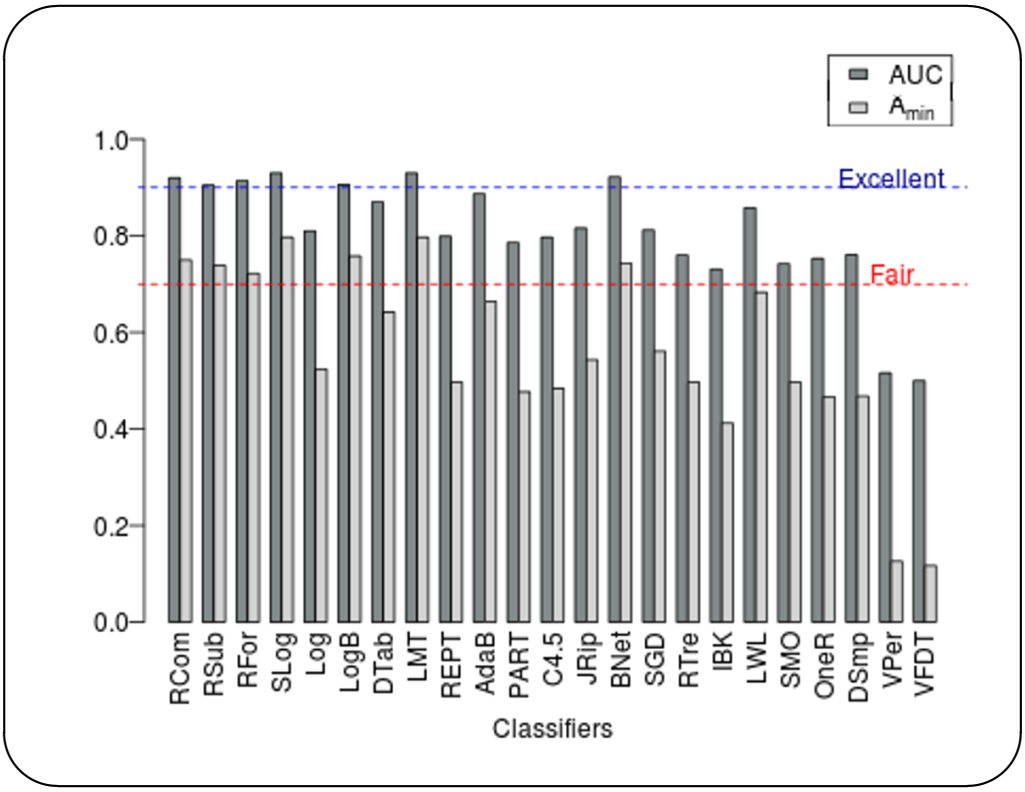


Figure 4.4: *AUC and* $A_{\min}$ *of classifiers for raw dataset* S_2.

Table 4.3: *AUC*, A_{min} and *ACC* of classification algorithms using raw dataset S_1, and after sampling, normalization and feature selection.

Classifier	Raw Dataset			Sampled			Normalized			Features Selected		
	AUC	A_{min}	*ACC*	*AUC*	A_{min}	*ACC*	*AUC*	A_{min}	*ACC*	*AUC*	A_{min}	*ACC*
RCom	.972	.937	.940	.984	.980	.943	.983	.979	.939	.982	.977	.939
RSub	.959	.923	.911	.978	.978	.928	.976	.976	.921	.975	.974	.927
RFor	.965	.917	.920	.981	.974	.937	.977	.972	.922	.978	.970	.930
SLog	.926	.841	.867	.926	.911	.854	.926	.911	.853	.926	.911	.854
Log	.945	.839	.898	.916	.925	.872	.916	.919	.872	.916	.920	.872
LogB	.930	.837	.878	.933	.918	.874	.932	.918	.868	.932	.918	.868
DTab	.922	.833	.876	.942	.933	.872	.942	.936	.873	.943	.936	.872
LMT	.927	.808	.901	.937	.898	.908	.936	.896	.905	.937	.895	.904
REPT	.891	.789	.887	.932	.912	.892	.923	.903	.887	.923	.904	.887
AdaB	.908	.767	.855	.919	.899	.850	.917	.895	.849	.917	.895	.849
PART	.884	.764	.901	.916	.892	.904	.932	.906	.910	.925	.901	.909
C4.5	.853	.747	.893	.915	.885	.914	.900	.864	.906	.898	.861	.906
JRip	.853	.747	.893	.916	.887	.905	.918	.891	.894	.911	.877	.894
BNet	.881	.703	.808	.888	.853	.830	.889	.862	.824	.889	.862	.824
SGD	.950	.683	.888	.869	.815	.869	.866	.810	.864	.866	.810	.864
RTre	.853	.669	.883	.877	.830	.877	.879	.835	.879	.875	.826	.874
IBK	.851	.668	.881	.881	.839	.881	.881	.839	.881	.881	.839	.881
LWL	.870	.644	.736	.874	.819	.773	.874	.819	.773	.874	.819	.773
SMO	.803	.628	.867	.866	.815	.866	.866	.814	.865	.866	.814	.865
OneR	.767	.553	.834	.795	.733	.794	.795	.733	.794	.795	.733	.794
DSmp	.769	.450	.733	.774	.692	.774	.774	.692	.774	.774	.692	.774
VPer	.543	.289	.638	.526	.515	.527	.802	.735	.798	.802	.736	.798
VFDT	.500	.267	.733	.638	.582	.606	.736	.668	.661	.736	.668	.661

Although the effect of feature selection is negligible on the prediction performance of the classifiers, it has two advantages; the first advantage is the provision of faster and more cost effective calssifiers, and the second is the provision of a better understanding of the underlying process that generated the data.

4.4.3 The Powerful discriminators

Wavelet-based texture measurement has shown their superiority to discriminate our instances in the top classifiers. Specifically, the *Smoothness* and *Uniformity* textures in the horizontal, diagonal and vertical wavelet detail images were used as root nodes in most base trees in the *RCom*, *RFor*, *RSub* and *C4.5* Decision tree classifiers. In addition, the fusion of invariant moments with wavelet details in many dimensions obtained high weights in the functions of *SLog* and *LMT* classifiers revealing their discriminative power. They also showed up multiple times within the decision trees built by various models. The second order histogram features extracted from the co-occurrence matrix have also played a major role in the discrimination of yeast cells, they showed up in almost every classifier, though at lower level in decision trees or

Table 4.4: Area under *ROC* and Accuracy of the top 10 classification algorithms using raw dataset S_2, and after sampling, normalization and feature selection.

Classifier	Raw Dataset			Sampled		Normalized		Features Selected	
	AUC	AC1	ACC	AC1	ACC	AC1	ACC	AC1	ACC
SLG	.930	.797	.944	.987	.948	.987	.948	.986	.948
LMT	.930	.797	.944	.981	.953	.981	.956	.979	.956
BYN	.922	.743	.918	.974	.930	.971	.927	.971	.927
RNC	.920	.750	.937	.991	.973	.992	.972	.991	.968
RNF	.914	.722	.938	.988	.966	.991	.964	.990	.962
LBS	.906	.758	.942	.975	.920	.975	.919	.975	.919
RSS	.905	.739	.938	.990	.955	.989	.958	.988	.955
BGG	.903	.763	.938	.989	.952	.990	.950	.990	.951
ABM	.887	.664	.930	.966	.910	.967	.903	.967	.903
DTB	.870	.642	.930	.953	.888	.961	.895	.964	.898

with a smaller weight in regression functions.

Considering dataset S_1, most of the top classifiers built complex models that are not easy to interpret. *RCom* has built ten random trees of sizes between 267 and 297. *RSub* built a "relatively" less complex model of ten random *REP* trees of sizes between 47 and 87. *RFor* built ten random trees each using seven random attribute values in its construction. On the other hand, the *Logistic* and *Simple Logistic* algorithms built decent regression functions. *Simple Logistic* built its function with a few attributes (22 of the total 90 considered). This makes *Logistic* regression a much more appealing classification model for our domain. The *Logistic regression* classifier has two output terms, coefficients and odds ratios. High coefficient values when predicting the negative class were noticed in many moment invariant features in the wavelet detail images. Moreover, high odds ratios were noticed in texture measurements and co-occurrence matrix features. The *Simple Logistic* has used in its function, eight features from the co-occurrence matrix, five features from the wavelet texture measurement, two features from the combination of wavelet and moment invariants, four features from moment invariants, one texture measure and two basic shape descriptors.

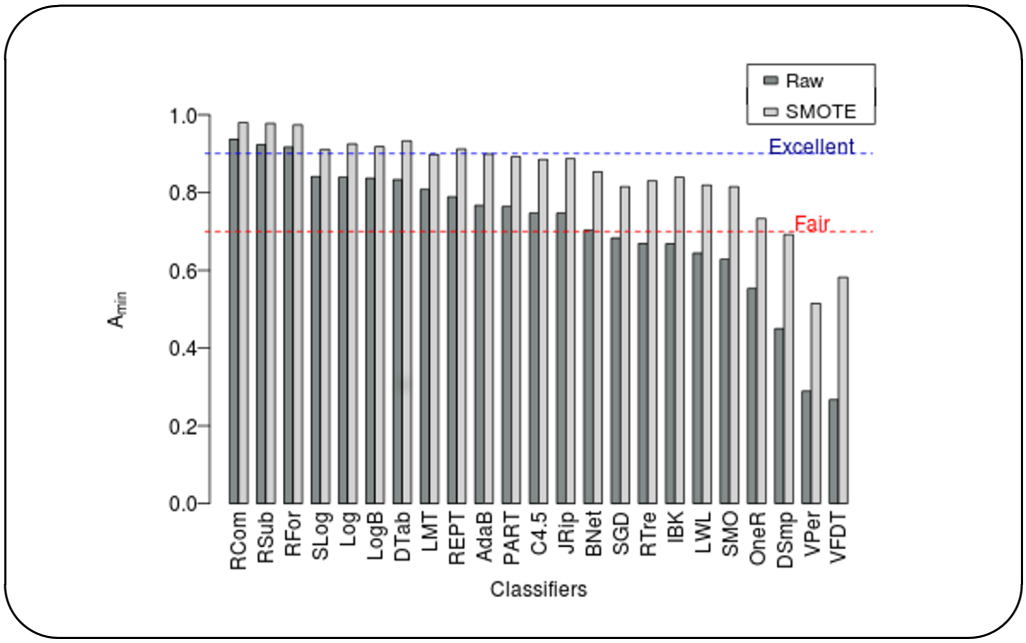


Figure 4.5: A_{min} *of classifiers for raw and sampled dataset* S_1.

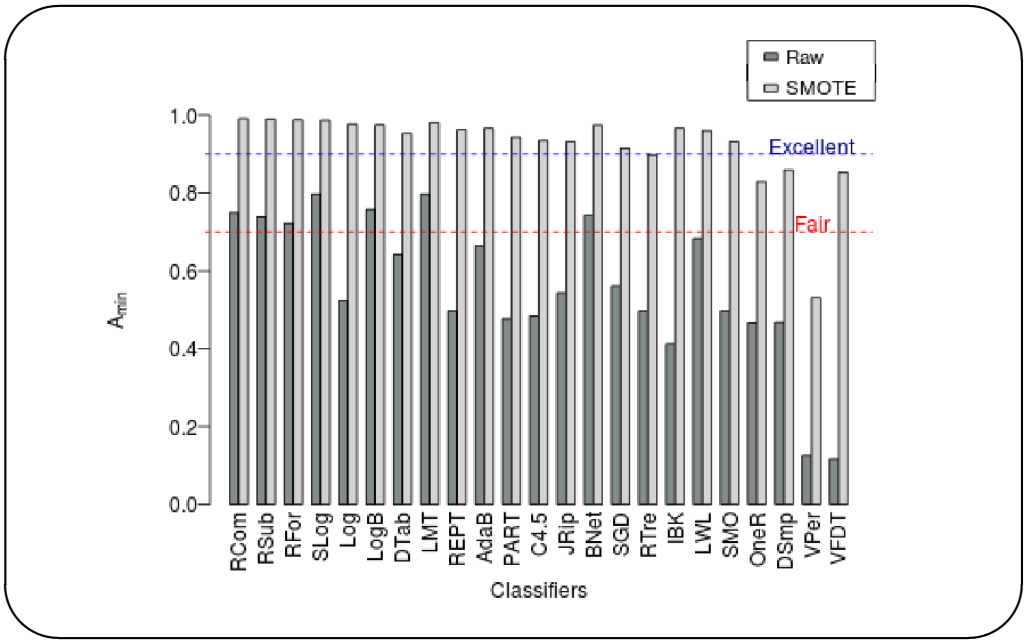


Figure 4.6: A_{min} *of classifiers for raw and sampled dataset* S_2.

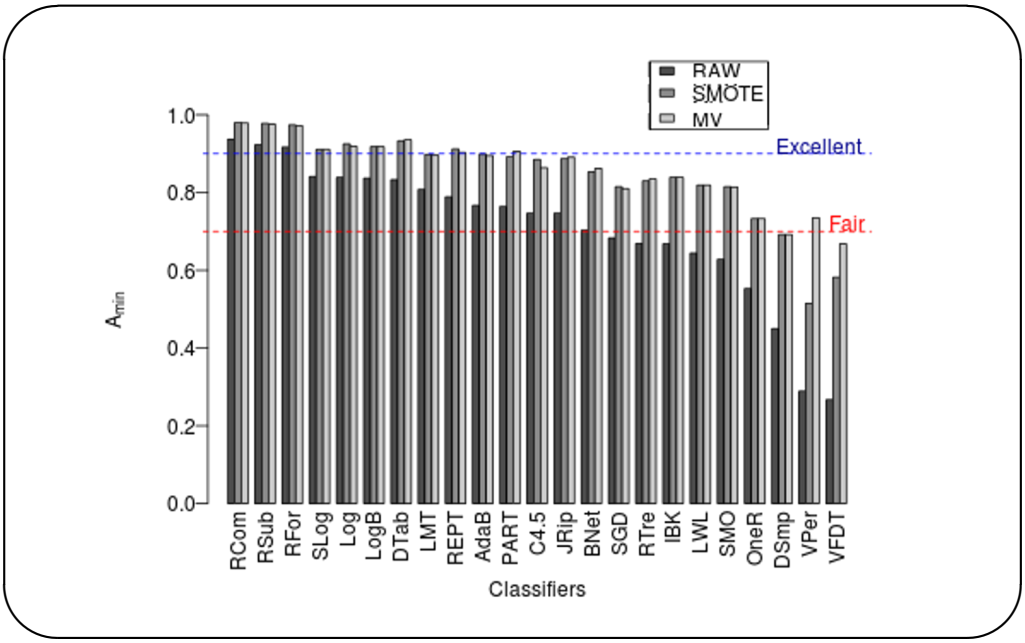


Figure 4.7: A_{min} *of classifiers for raw, sampled and scaled dataset* S_1.

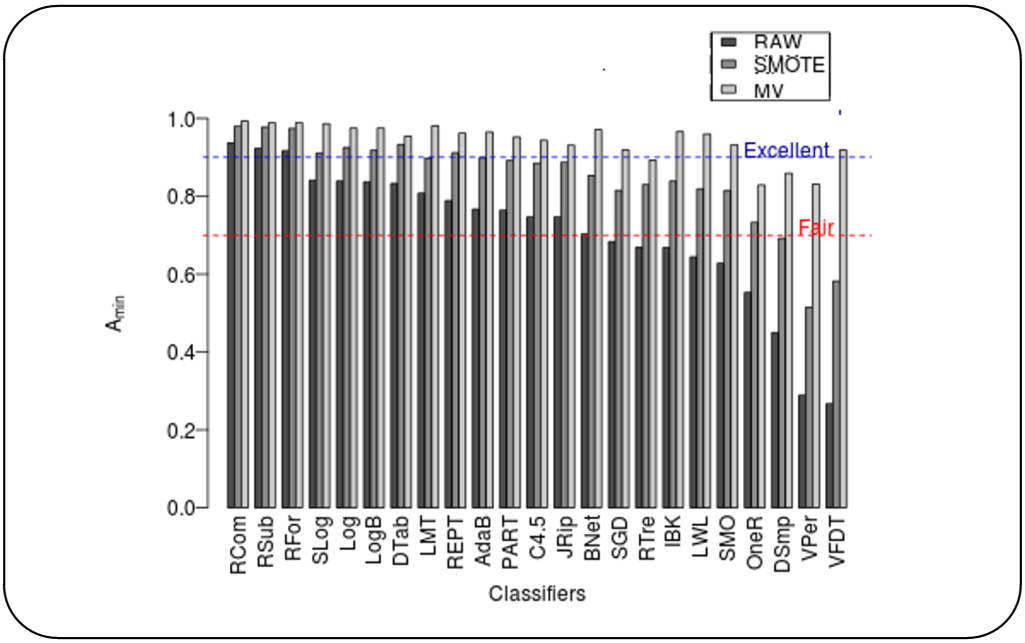


Figure 4.8: A_{min} *of classifiers for raw, sampled and scaled dataset* S_2.

Support Vector Machines (*SVMs*) are popular classifiers. The *SMO*, which is an implementation of the *SVMs*, did not rank as an excellent classifier in dataset S_1 when using the default parameters. However, this changes when optimizing its parameters by increasing the complexity constant to 5, disabling any normalization within the classifier itself, fit logistic models to *SVM* outputs and use a normalized Polynomial Kernel.

Figure 4.9 shows initially that the *SVM* classifier evaluated as a poor classifier with an A_{min} value of .628. This value is improved into "good" level after sampling with the *SMOTE* algorithm, i.e. .815. However, optimization pushed the *SMO* into the top five classifiers with an A_{min} of 0.92.

Next we study the considered feature sets for their contribution to the power of discrimination.

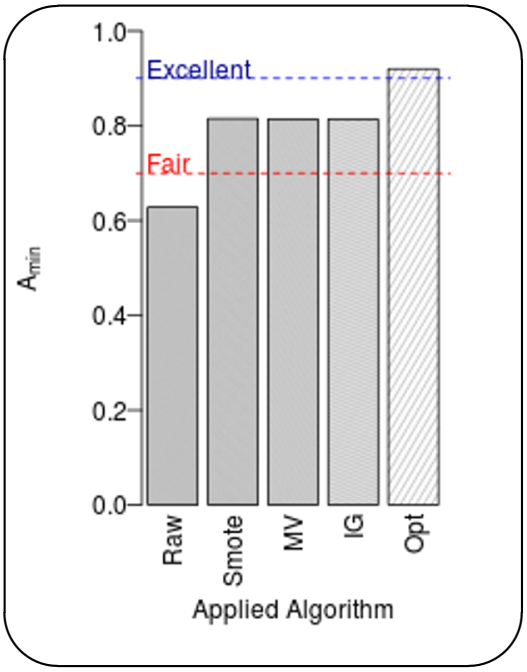


Figure 4.9: A_{min} *of SMO classifier for raw dataset* S_1*, and after the application of SMOTE, MV and IG algorithms for sampling, scaling and feature selection respectively.*

4.4.4 Feature sets performance

In order to investigate whether our composed feature sets has any added value to the discrimination power, we started by comparing the classification performance on dataset S_1 using different set of features including basic shape descriptors, invariant moments, wavelet texture measurement, invariant moments on wavelet detail images, co-occurrence matrix derived features, basic texture measurement and a full feature space combining all the feature sets. The difference is shown in Fig. 4.10. This test was performed on both Logistic and C4.5 classifiers. Logistic regression was chosen as it is the top "non-random based" classifier and C4.5 as a non-linear approach for comparison. In both classifiers, the benefits of using the full set is obvious. In the Logistic classifier, the performance of individual feature sets were not sufficient, except for the basic texture measurement which shows a very good discrimination with an *AUC* of 0.82. However, using the full features set shifted the performance of the Logistic classifier into the excellent category with an *AUC* of 0.916. In the non-linear C4.5 decision tree classifier, all the individual feature sets except the basic set have good discrimination. However, none is ranked as excellent. Only when the feature sets are fused together the classifier has an excellent discrimination rate with an *AUC* of 0.914.

Since 3^{rd} order moment invariants are more complex than there 2^{nd} order counterpart, and they might be more noise-prone [Goo96], we study whether it really adds any discrimination value by creating only two small feature spaces. The first having only the second moment

invariants while the second feature space contains the whole set of seven invariant moments. Figure 4.11 show the *ROC* graph of the performance of the `Logistic` and C4.5 classifiers on both feature spaces. The third order moments show to have an additional discriminative value in both classifiers for this dataset.

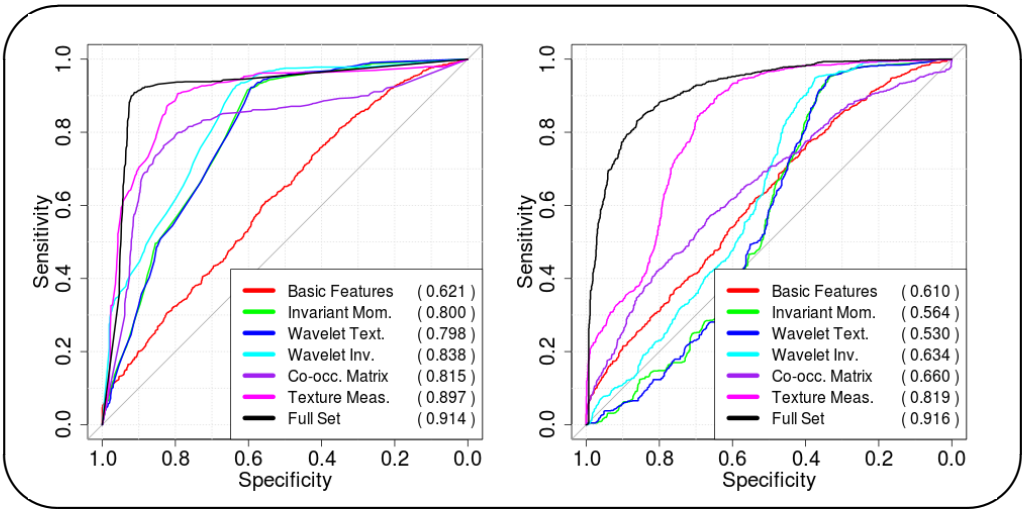


Figure 4.10: *ROC analysis and AUC value of C4.5 (left) and Logistic (right) classifiers using various feature sets*

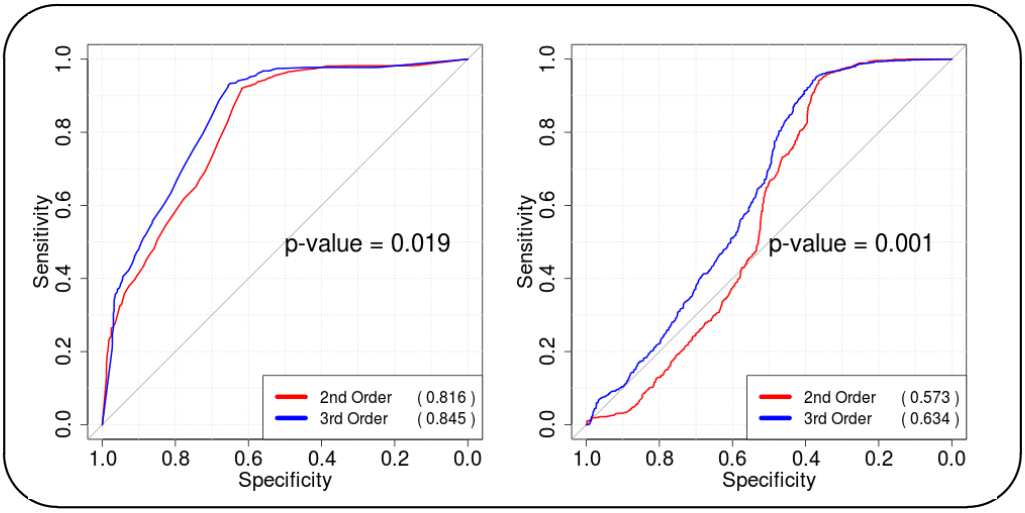


Figure 4.11: *Performance of C4.5 (left) and Logistic (right) classifiers using second and up-to third order moment invariant features*

4.5 Discussion

In this chapter, we addressed our principal research question and showed that a machine learning approach can discriminate *S. cerevisiae* yeast cells to identify essential characteristic differences between two groups treated under different conditions, and a similar approach can improve the object recognition based on its measured sophisticated features. In addition we address the sub question and showed that a combination of various feature extraction methods have a significant role in improving the accuracy of the built classification models.

In the case study of yeast cells, we were able to find a significant difference in the expression of 14-3-3 proteins for cells cultivated under different stress levels. This difference in their characteristics is a subtle pattern that is hard to be noticed using standard methods, such as investigation the measurement data and examining the basic statistical information. On the other hand, the object recognition allow us in our experiment to exclude artefacts prior to data analysis, which consequently improves the identification of subtle patterns.

In this work, we introduced a machine learning workflow to be followed in building a classification model. We showed that the extracted object features explained in Section 4.2 are advantageous to predict the group that an object belongs to. The Wavelet-based texture measurements, co-occurrence matrix derived features, moment invariant features and texture measurement derived from the image histogram, forms a set of powerful discriminators in the top classification models, from which the Logistic model was chosen as the most efficient for classifying our cells. Sampling of our dataset with the *SMOTE* method showed to have a significant effect on building the classification model system. In addition, the *MV* normalization scheme showed an extra improvement. The best feature selection algorithm tested showed a little non-significant improvement; however, it provides a more simple, faster and cost effective classification model. With this machine learning process and the chosen feature sets, it becomes possible as future work, to classify different cell strains and conditions in a high-volume high-throughput studies.

5

Role of 14-3-3 proteins and Nha1 antiporter in the response of *S.cerevisiae* to salt stress

> *In Chapter 2, we described our developed platform, i.e. YeastAnalysis. In this chapter, the developed platform is tested in studies on the role of* 14-3-3 *proteins and the* Nha1 $Na^+(K^+)/H^+$ *antiporter in the response of S. cerevisiae cells to high external* NaCl *concentrations. To this end we investigate the effect of high external* Na^+ *concentration on the levels of GFP-tagged* Bmh1, Bmh2 *and* Nha1. *For validation of the software tool the results were compared to results obtained by flow cytometry.*

This chapter is based on the following publication:

- Mohamed Tleis, Paul van Heusden and Fons J. Verbeek.*YeastAnalysis*: An image analysis platform to quantify fluorescent reporter proteins in *Saccharomyces cerevisiae* cells. (Submitted).

5.1 Introduction

Sacharomyces cerevisiae yeast as a model system is very suitable to study the function of proteins. This can be achieved through biochemistry and/or molecular biology or, if location is important through imaging. There is a class of proteins, referred to as the 14-3-3 proteins that are very important to maintain the integrity of the organism. These 14-3-3 proteins are found in all organisms and are often related to stress situations, i.e. disease [Rob02, Mar04]. Therefore, an experiment on stress in *S. cerevisiae* yeast is designed to further investigate these proteins. There we use salt stress, which can simply be induced by increasing the salt concentration in the medium.

S. cerevisiae has two genes encoding 14-3-3 proteins, *BMH1* and *BMH2*. In this study we address the function of the Bmh1 and Bmh2 14-3-3 proteins and the Nha1 cation transporter protein that modulates ion homeostasis. To address this function in the tolerance of yeast cells under salt stress, we cultivated strains expressing Nha1-GFP, Bmh1-GFP and Bmh2-GFP reporters under standard conditions and in the presence of 0.5 M NaCl. These reporter proteins are expressed by the *NHA1*, *BMH1* and *BMH2* genes tagged with the *GFP* reporter gene that expresses a green fluorescent protein. We also included a mutant strain with a deletion of *BMH1* gene (*Δbmh1*) expressing Nha1-GFP protein. Images were made by confocal laser scanning microscopy. As shown in Fig. 5.1, Nha1-GFP was mainly found in the plasma membrane, in agreement with the reported localization [Huh03]. Deletion of *BMH1* gene did not affect the localization of Nha1-GFP protein. Both Bmh1- and Bmh2-GFP were found all over the cell, also in agreement with previous reports [Huh03]. In order to fully understand the role of the 14-3-3 proteins and Nha1 protein in salt tolerance, the effect of NaCl on the levels of GFP-tagged proteins needs to be quantified. To this end we have used *YeastAnalysis* (cf. Chapter 2) to analyze microscope images using an extended set of features. In the following sections we discuss the materials and methods used in this experiment. Subsequently we discuss how the experiment analysis is performed including segmentation, measurement and data analysis. Then we discuss the results obtained from the application of *YeastAnalysis* to study the function of *BMH1*, *BMH2* and *NHA1* genes in the response to salt stress. We complete this chapter with a conclusion.

5.2 Materials and Methods

In this section we describe the materials and methods used to conduct the experiment explained in the introduction. The development of *YeastAnalysis* was already discussed in Section 2.8. Herein, we first highlight the used strains and how they were cultivated. Subsequently, we discuss the used microscopy and flow cytometry techniques.

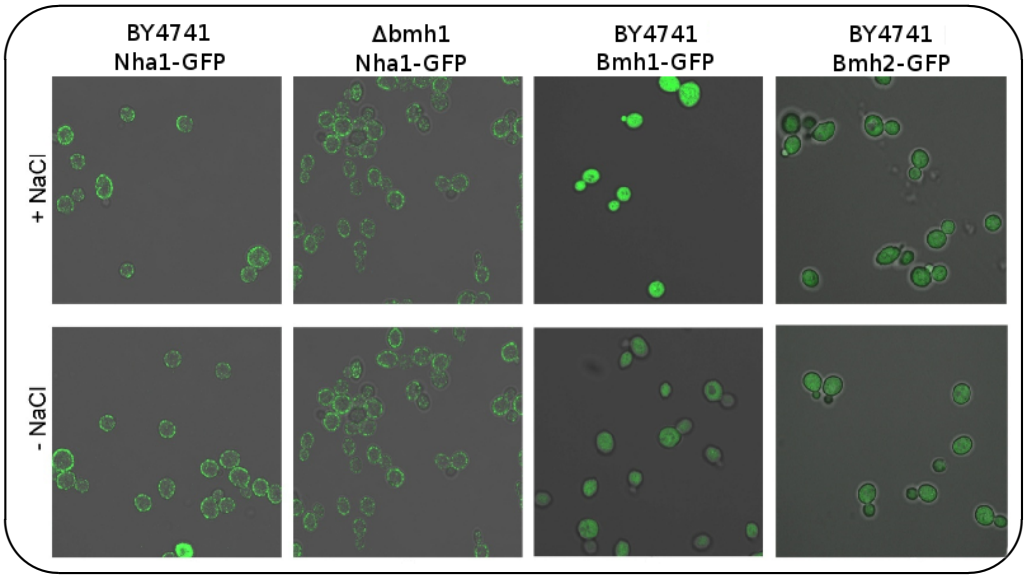


Figure 5.1: *Confocal laser scanning microscopy of* BY4741 NHA1-GFP, *Δbmh1* NHA1-GFP, BY4741 BMH1-GFP *and* BY4741 BMH2-GFP *cells grown in supplemented* MY *medium with (+*NaCl*) or without (−*NaCl*) additional* 0.5 M NaCl.

5.2.1 Yeast strains and cultivation

The yeast strains used in this study are listed in Table 5.1. *Δbmh1 NHA1-GFP* was constructed by integration of GFP downstream of the *NHA1* gene coding sequences in strain *Δbmh1* (GG3240). For this purpose a PCR fragment was generated using pYM28 [Jan04] as template and the primers pYM-NHA1-Fw (5′-GCTGCTGTTAAGTCGGCGCTATCAAAAACGCTT GGTCTCAATAAGCGTACGCTGCAGGTCGAC-3) and pYM-NHA1-Rev (5′-CGACACA TGTAAATAAAAAAGGCATTTCGTTTATATATATACTAAATCGATGAATTCGAGC TCG-3′). Transformants were selected for histidine prototrophy. Yeast transformations were performed using the LiAc method [Gie95]. Yeast was cultivated in MY medium supplemented, when required, with histidine, methionine, uracil and leucine [Zon86].

For microscopy and flow cytometry, yeast cells were grown overnight in MY medium supplemented with histidine, methionine, uracil and leucine, when required. The next morning cultures were diluted ten-fold in supplemented MY medium containing 0.5 M NaCl or in the same medium without added NaCl. After cultivation for 4 to 7 hours at 30°C, cells were analyzed by confocal microscopy and/or flow cytometry.

Strain	Genotype	Source/Reference
BY4741	*MATa his3Δ1 leu2Δ0 met15Δ0 ura3Δ0*	Euroscarf,Germany
Δbmh1 (GG3240)	*MATa his3Δ1 leu2Δ0 met15Δ0 ura3Δ0 bmh1Δ::loxP*	Wouter Hendriksen, unpublished results
NHA1 – GFP	*MATa his3Δ1 leu2Δ0 met15Δ0 ura3Δ0 NHA1-GFP (HIS3MX6)*	Life Technologies
BMH1 – GFP	*MATa his3Δ1 leu2Δ0 met15Δ0 ura3Δ0 BMH1-GFP (HIS3MX6)*	Life Technologies
BMH2 – GFP	*MATa his3Δ1 leu2Δ0 met15Δ0 ura3Δ0 BMH2-GFP (HIS3MX6)*	Life Technologies
Δbmh1 NHA1-GFP (GG3398)	*MATa his3Δ1 leu2Δ0 met15Δ0 ura3Δ0 bmh1Δ::KAN.MX NHA1-GFP (HIS3MX6)*	This study

Table 5.1: *Yeast strains used in this study.*

5.2.2 Confocal microscopy and flow cytometry

For image acquisition a Zeiss LSM 5 Exciter-AxioImager M1 confocal microscope with a Plan-Apochromat objective (63X/1.4 Oil DIC) and Zeiss ZEN 2009 software were used. GFP was imaged with excitation at 488 nm and emission at 505-530 nm. For flow cytometry, a Merck-Millipore Guava EasyCyte 5 Flow Cytometer was used. Fluorescence was determined after excitation at 488 nm and using the standard green 525/30 nm emission filter. For each flow cytometry analysis, 5000 cells were used.

5.3 Experiment Analysis

Here we discuss the steps followed in the analysis of the experiment images; i.e. segmentation, measurement and data analysis.

5.3.1 Segmentation

The first step in the analysis is to locate the individual cells in the image through segmentation. In this experiment, we apply segmentation on the bright-field channel of the image, and we use the segmentation results to analyze the other channels. In the bright-field images, the cell contours are visible as dark structures. We applied the *HCSP* segmentation algorithm described in Section 3.2 and Section 3.4.2. It first applies a 3x3 *Prewitt* filter, which gives a good initial estimate of the magnitude of the gradient of the image. i.e. it highlights the cell contours. On this gradient image, a threshold and skeletonization algorithm [Gon08] are applied to get the contours as binary structures of one pixel thickness. The following step is the utilization of Hough Transform to locate the structures that form part of a geometrical circle, for the fact that

yeast cells are nearly circular. The detected partial circumference of this circle corresponds to the detected part of the cell contour, and the center point of this circle corresponds to a point somewhere in the middle of the cell. The location of the center point is not necessarily at the center of the cell, as the next step creates a sufficient resampled polar image to evaluate the complete cell. This subsequent step is the extraction of the exact contour of the cell starting from the center point for that cell located by Hough transform. Figure 5.2(a) shows an example of a cell after locating its center point by Hough Transform. The *red* circles and *blue* lines are labels to illustrate the creation of the resampled polar image of the cell as shown in Fig. 5.2(b). The radius at an angle θ (*blue* lines in Fig. 5.2(a)) in the image plane, is transformed into a column in the polar image (vertical blue lines in Fig. 5.2(b)). The circles surrounding the center point (*red* circles of radius r in the image plane in Fig. 5.2(a)), are transformed into rows of the polar image (red horizontal lines in Fig. 5.2(b)). Next, a minimal path algorithm is applied on the polar image to find the full contour of the cell. Figure 5.2(c) shows the detected minimal path in *cyan*. Figure 5.2(d) shows the detected cell contour as *cyan* pixels backprojected to the original image. The results of segmentation of a typical microscope image are shown in Fig. 5.3.

During the segmentation process, outliers are discarded by setting limits to the minimum and maximum cell size as well as to the minimum circularity of the detected shape. The extracted cell contours or the masks are subsequently used to measure the fluorescence intensity in the overlaid fluorescence channel, and for measurements in the bright-field channel.

5.3.2 Measurement

After detection of the cells in the microscope images, the *YeastAnalysis* software is used to perform various measurements on these cells. In addition to the fluorescence intensity, several shape features of the cells are measured in the bright-field image, including size (area), perimeter, density, and features describing the textures of the cell such as variance, relative smoothness, skewness, uniformity and entropy [Gon08]. The same set of features are measured in the bright-field channel as well as in the overlaid fluorescence image channels. Table 5.2 lists how texture measurements can possibly be interpreted within images of *S. cerevisiae* cell.

For analysis of membrane proteins *YeastAnalysis* automatically estimated the region where the cell membrane protein are expressed by focusing on the pixels next to the cell contour (Fig. 5.4). All measurements are saved automatically into a CSV (comma-separated values) file as explained in Section 2.4. This measurement file is used at the following step to generate a report and for further inspection by a specialist.

5.3.3 Data Analysis

After completion of the measurements, two sets of cells are compared to study any differences between those sets. The first set is those cells cultivated in a *non*-NaCl medium, while the second set is those cultivated in a 0.5 M NaCl medium. *YeastAnalysis* automatically generates a report in *pdf* format holding basic statistics and graph charts to visualize the results.

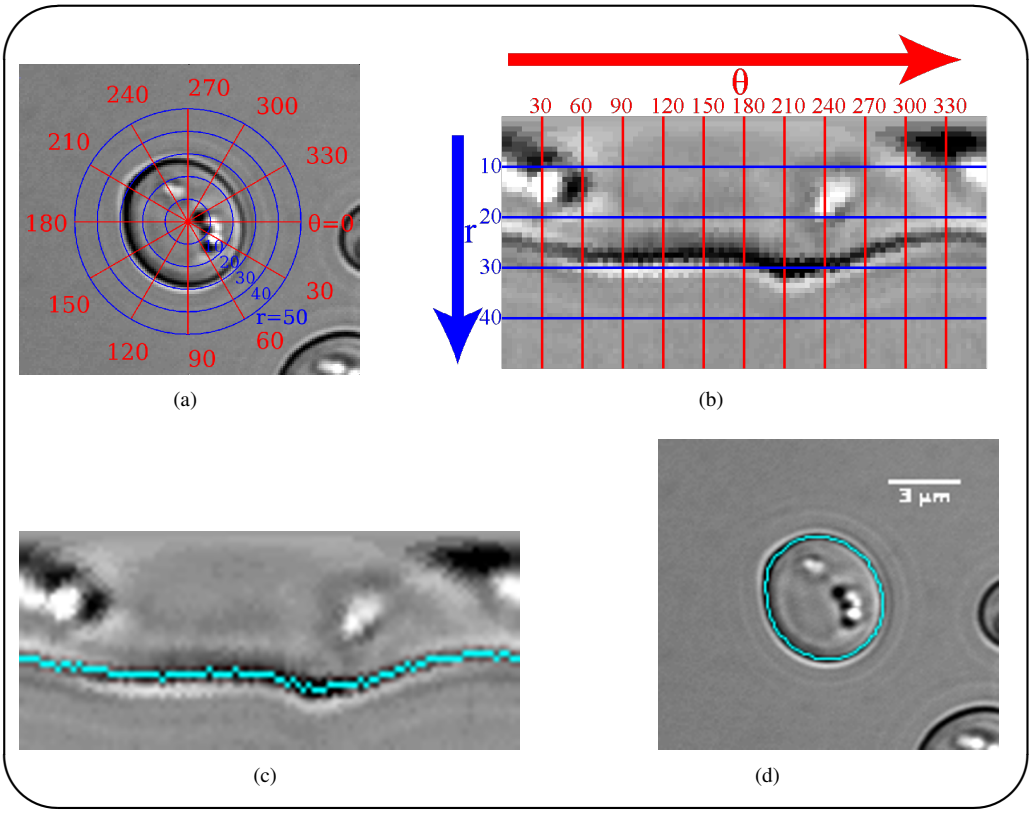


Figure 5.2: ***Detecting contours by Hough Transform and Minimal Path algorithm.*** *Image (a) shows an example of a cell after detection of the center point by Hough Transform. In (b) a polar image is generated. The columns correspond to the pixels along the radius at an angle θ of the largest possible circle (red lines). The rows correspond to the circles surrounding the center point (blue lines, blue circles in image a). Image (c) shows the detected path from the first column to the last column in the polar image, after the application of dynamic programming. Image (d) shows the detected path as the actual contour of the cell.*

The statistical information includes the number of detected cells and the mean values of the different measured features along with their standard deviations. The unpaired Student t-test is performed to report the t-value and p-value to assist in analyzing the significance of the differences between two sets of cells. To visualize the measurement results, several chart types are generated, including scatter plots, Pareto charts and box-and-whiskers plots. The next section shows the results obtained using *YeastAnalysis* to study the function of *BMH1, BMH2,* and *NHA1* genes under salt stress.

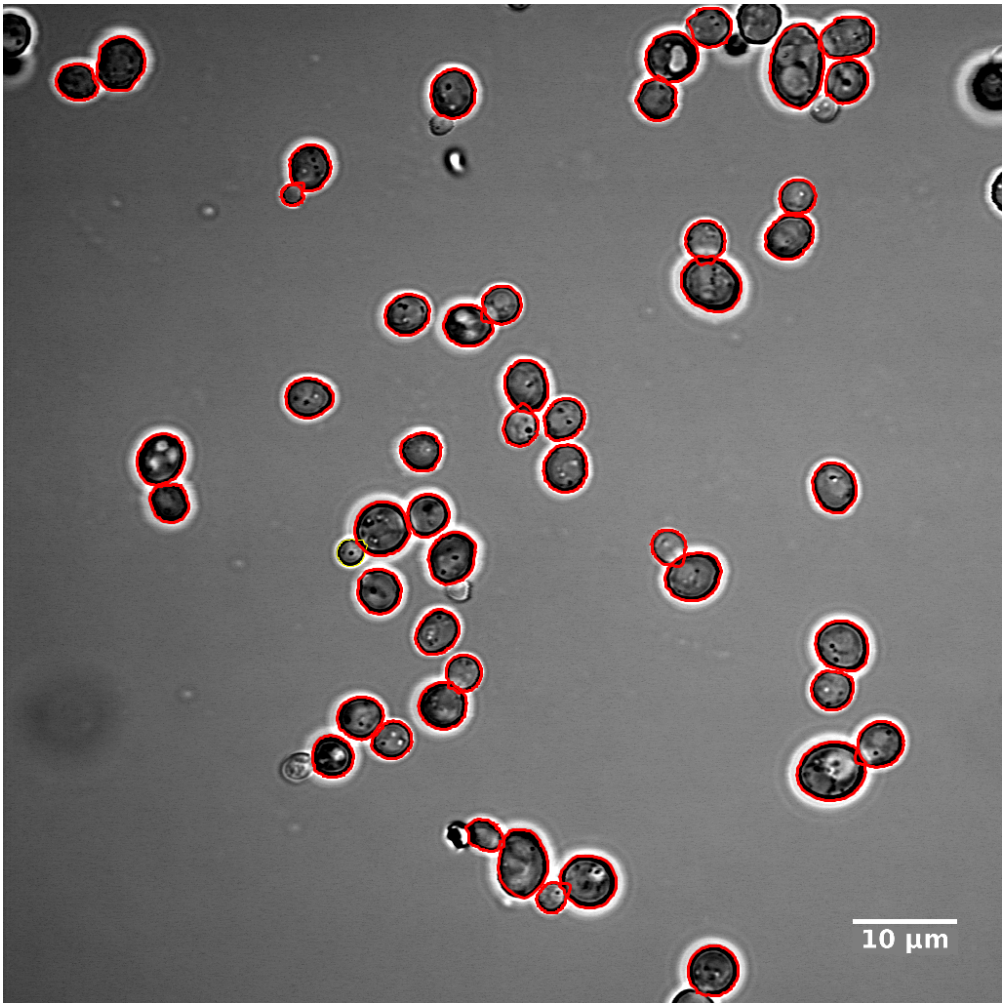


Figure 5.3: *Overlay on a bright-field image showing the extracted contours of the detected cells.*

Texture	Bright-Field Channel	Fluorescent Channel
Entropy	A higher entropy value can indicate a distorted cell as in cell death.	Lower entropy values can indicate that expressed proteins are forming simple patterns within cells. Higher values can indicate a more complex and variable patterns.
Skewness	Lower skewness values can indicate brighter cells while higher values can indicate darker cells.	Lower Skewness values can indicate less intensity while higher values can indicate higher intensity values within the cell.
Uniformity	Uniformity values go lower when there are no dark structures or organelles appearing within the cell, or when the cell is not distorted.	Uniformity values can go higher when the gene is expressed more evenly throughout the cell.
Variance	Indicates how many dark structures appear or how distorted a cell is	Indicates a constant expression throughout the cell (low variance), or an highly variable expression throughout the cell (high variance).

Table 5.2: *Interpretation of texture measurements in yeast.*

5.4 Results

We validated *YeastAnalysis* in studies on the effect of salt stress on yeast cells expressing GFP-tagged Bmh1, Bmh2 or Nha1. To this end, six to ten confocal microscope images were acquired with in total more than 200 cells for each condition. We have four different conditions in two different classes; i.e. the *non*-NaCl and the 0.5 M NaCl class. The number of samples per condition per class are depicted in Table 5.3. The acquired images in this experiment contain two channels, a fluorescence channel and a bright-field channel. The fluorescence images show the expression of *BMH1-GFP, BMH2-GFP, NHA1-GFP* and *Δbmh1 NHA1-GFP* in yeast cells cultured in the absence or presence of additional 0.5 M NaCl. We loaded these images into *YeastAnalysis* and used the *HCSP* algorithm for detection of the cells in the bright-field image. By automatic segmentation of the total acquired 195 microscope images in this study we were able to detect 92 percent of the cells on these images, whereas 2.2 percent of the detected objects did not correspond to learned shape of a healthy cell; i.e. debris and dead cells. The F1-Score [Rij79] is used to measure the effectiveness of the algorithm. The F1-Score measure is 0.95. The missing cells were added by manual segmentation and the false positives were removed. Such manual segmentation is easily performed as non-detected cells are added

through manual seeding and incorrect detections are deleted. These adjustments are facilitated through an interactive Graphical User Interface (GUI) in the *YeastAnalysis* software.

The segmented images were used for measurement of various cell features. These measurements are exported into a csv-file showing the cells expressing GFP-tagged Bmh1, Bmh2 and Nha1 cultured in the absence and presence of additional 0.5 M NaCl. For the measurements of the fluorescence channel, we selected the most common features; i.e. the total intensity, cell size, cell perimeter, internal density, intensity per pixel, intensity per square micron, circularity, variance, skewness, uniformity, relative smoothness and entropy.

The results of the measurements of the cell size and fluorescence intensity in a typical experiment on the effect of additional 0.5 NaCl on cells expressing Nha1-GFP, Bmh1-GFP

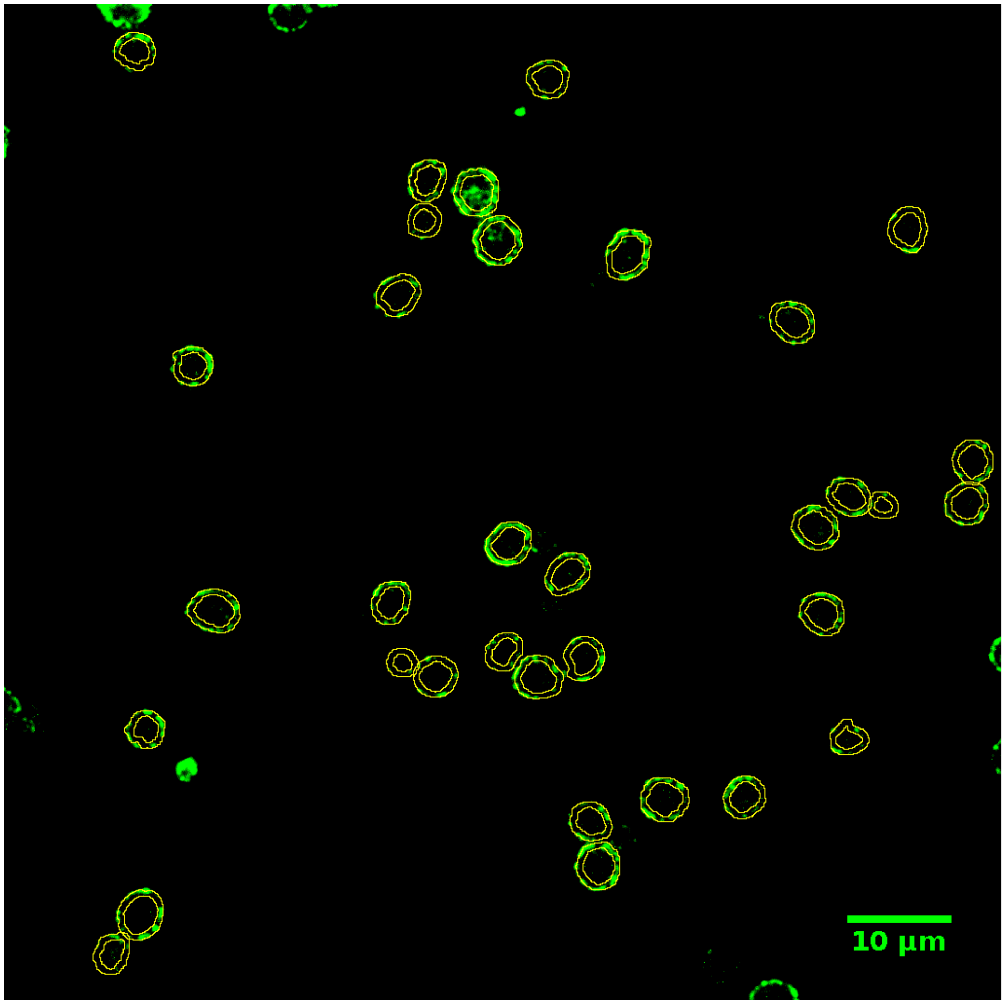


Figure 5.4: *Automatically generated overlay showing estimated cell membrane locations.*

Strain	Cell Size in μm^2		Fold Change Cell Size +NaCl/ -NaCl (P-Value)	Integrated GFP fluorescence		Fold Change GFP +NaCl/ -NaCl (P-Value)
	-NaCl	+NaCl		-NaCl	+NaCl	
NHA1-GFP	16.7±5.2 (407)	15.6±4.7 (281)	0.93 P = 0.002	4.3±2.2 (407)	4.5±2.6 (281)	1.05 P = 0.44
Δbmh1 NHA1-GFP	13.6±4.2 (915)	15.3±5.6 (219)	1.12 P < 0.001	3.2±1.7 (915)	5.6±2.9 (219)	1.75 P < 0.001
BMH1-GFP	12.7±4.2 (447)	12.6±4.5 (328)	0.99 P = 0.74	32.6±15.2 (447)	56.6±18.3 (328)	1.74 P < 0.001
BMH2-GFP	12.9±4.5 (311)	11.7±3.7 (238)	0.91 P = 0.001	20.0±10.4 (311)	33.9±13.8 (238)	1.70 P < 0.001

Table 5.3: *Determination of cell size and GFP fluorescence in microscope images using YeastAnalysis. The statistical method used to check the significance of the differences (P-Value) was the Student t-test. Numbers within parenthesis represents the number of cells used in the analysis. Arbitrary unit is used for GFP fluorescent.*

Strain	GFP fluorescence [Arbitrary Units]		Fold Change GFP fluorescence +NaCl/-NaCl
	-NaCl	+NaCl	
NHA1-GFP	2.53	2.98	1.18
Δbmh1 NHA1-GFP	1.83	2.72	1.49
BMH1-GFP	77.3	131.0	1.69
BMH2-GFP	43.5	65.6	1.51

Table 5.4: *Determination of GFP fluorescence using flow cytometry.*

or Bmh2-GFP are summarized in Table 5.3. Moreover, the effect of deletion of *BMH1* gene on Nha1-GFP expression is shown. The results show that NaCl stress results in a slight increase in the expression of Nha1-GFP in the wild type BY4741 background but a stronger increase in the mutant *Δbmh1* background, whereas the expression of both Bmh1- and Bmh2-GFP is strongly increased upon salt stress. The results further show that *Δbmh1* cells are slightly smaller than wild type BY4741 cells (P-values from *Student* t-test less than 0.001 as seen in Table 5.3). Figure 5.5 shows some visualization charts generated by *YeastAnalysis* for this analysis of *Δbmh1 NHA1-GFP* strain under salt stress. Figure 5.6 shows the analysis of their membrane intensity and size. All the experiments were repeated three times except for the cells expressing Bmh2-GFP that was repeated in duplo and similar results were obtained.

To validate the results obtained by analyzing microscope images by *YeastAnalysis* we used flow cytometry. To this end 5000 cells from each culture were analyzed and the *GFP*-fluorescence measurements are shown in Fig. 5.7. Calculation of the average GFP fluorescence

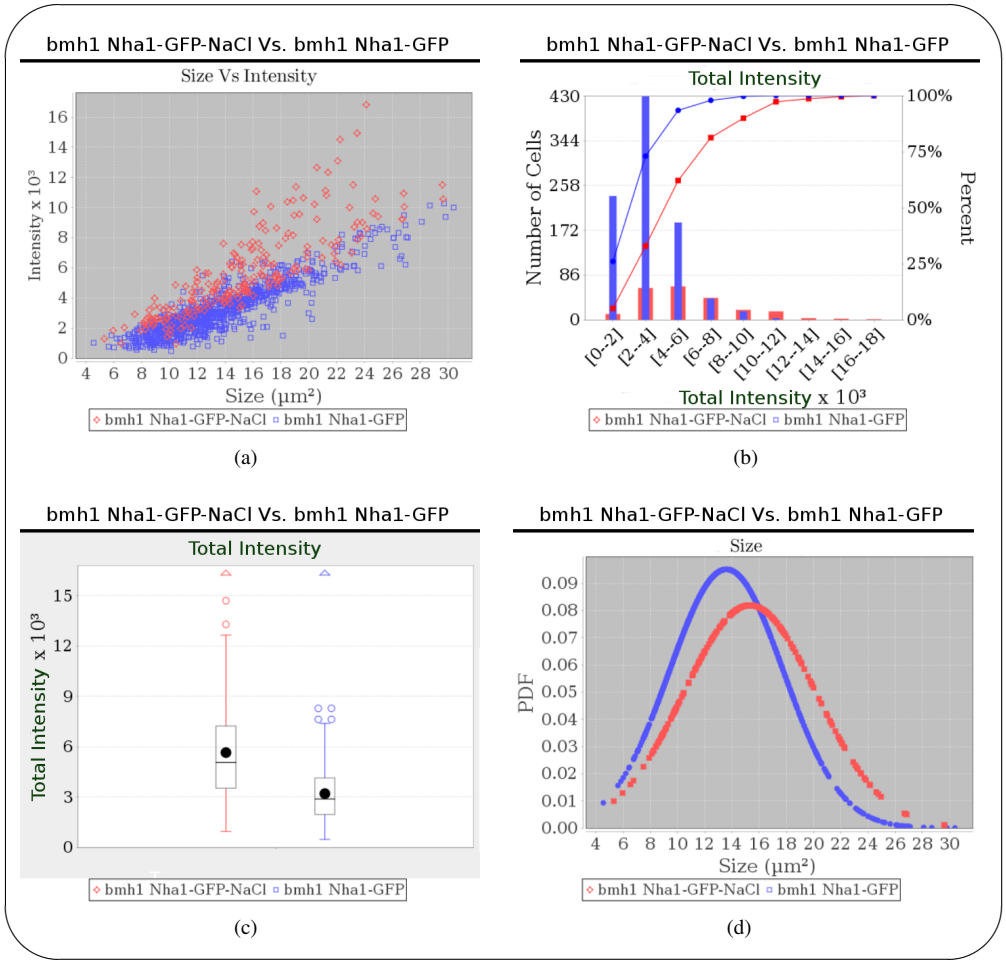


Figure 5.5: ***Visualization of data generated by YeastAnalysis.*** *Yeast strain Δbmh1 NHA1-GFP was cultivated in* MY *medium supplemented or not supplemented with* 0.5 M NaCl *and cells were analyzed by microscopy, followed by analysis of the images using YeastAnalysis. Results can be visualized in several ways. (a) - in Scatter Plot showing the cell size on the x-axis and fluorescent Intensity on y-axis. (b) - in a Pareto chart of the fluorescence intensity vs the number of cells. (c) - in a Box and Whiskers Plot. (c) - in a Scatter Plot fitted to Gaussian Distribution showing the cell size on the x-axis and the Probability Density Function (PDF) on the y-axis. Statistical analysis revealed that* bmh1 Nha1-GFP *total fluorescence intensity increased significantly upon salt stress (*1.75*-fold; P<* 0.001*). The size is increased significantly upon salt stress as well (*1.12*-fold; P<* 0.001*).*

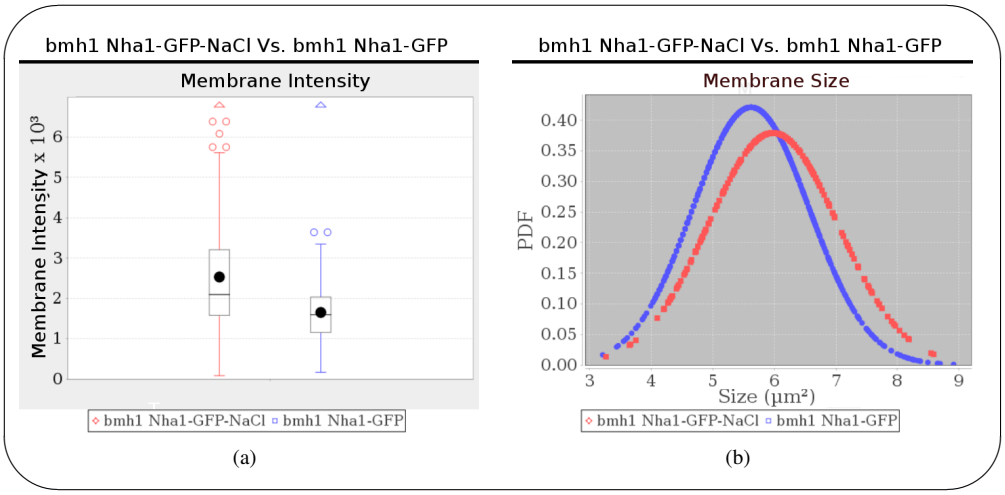


Figure 5.6: ***Membrane Intensity of Δbmh1 NHA1-GFP cell strain.*** *Statistical analysis revealed that bmh1* Nha1-GFP *fluorescence membrane intensity increased significantly upon salt stress (*1.53*-fold; P<* 0.001*). The size that this membrane protein occupies constitutes of an average of* 40 *percent of the total cell size.*

showed that the expression of Nha1- Bmh1- and Bmh2-GFP increased upon NaCl stress (Table 5.4). In summary, the results obtained by confocal microscopy and image analysis using *YeastAnalysis* and the results obtained by flow cytometry are in very good agreement.

YeastAnalysis is able to measure GFP intensity in the membrane (Fig. 5.4). In a typical experiment we showed that 63 percent of the GFP fluorescence was found around the membrane in BY4741 NHA1-GFP cells, whereas the membrane protein region constitutes, in average, 40 percent of the cell area (cf. Fig. 5.6). This result indicates that the majority of the Nha1 protein is expressed nearby the membrane, which corresponds well with the reported localization [Huh03].

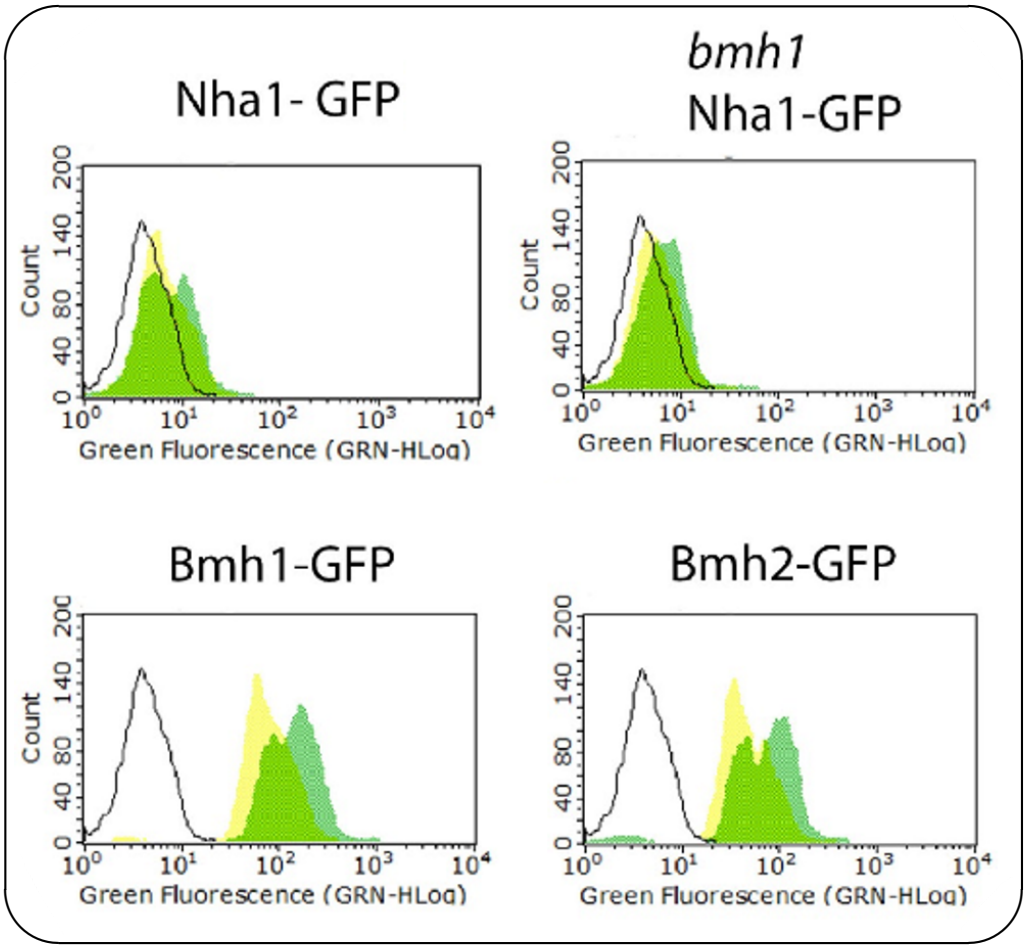


Figure 5.7: ***Flow cytometric analysis of the effect of*** 0.5 M NaCl ***on yeast cells expressing*** GFP***-tagged proteins.*** NHA1-GFP, BY4741 *expressing* Nha1-GFP*; bmh1* Nha1-GFP, *Δbmh1, expressing* Nha1-GFP*;* Bmh1-GFP, BY4741 *expressing* Bmh1-GFP*;* Bmh2-GFP, BY4741 *expressing* Bmh2-GFP. *Yellow: cells cultivated in the absence of additional* NaCl*; green: cells cultivated in the presence of* 0.5 M NaCl*; solid line, no fill:* BY4741 *cells.*

5.4.1 Yeast Vacuoles

Using the *YeastAnalysis* software, the size and number of vacuoles for an individual cell can be estimated. As some gene expression causes a fluorescence all over the cytoplasm, this fact can be used to find the size and shape of the vacuoles. In this experiment we study the vacuole of cells from *BMH1-GFP* wildtype strains. Our objective is to study the vacuole size and number of vacuolar compartments under salt stress in both strains. First we analyze the size of the central vacuoles. Subsequently we analyze the number of vacuolar compartments and their total size. After that, we repeat the experiments using images acquired from different experiments at different time points.

In our dataset we have 42 segmented cells from *BMH1-GFP* strain cultivated in 0.5 M NaCl, 131 *BMH1-GFP* cultivated under non-NaCl medium. In our analysis we compute the relative vacuole size; i.e. the vacuole size in terms of percentage of the cell size. In the first analysis we noticed that the central vacuole size occupies in average 3.5% of the cell size in *BMH1-GFP* strain, increased to 5.3% in 0.5 M NaCl. This result is depicted in Fig. 5.8 and Table 5.5(a). In order to assess the significance of this increase, we perform an unpaired Student t-test of the null hypothesis such that the means of the two samples (i.e. under non-NaCl and that of 0.5 M NaCl) are equal. The p-value from the *Student* t-test statistical analysis is 0.021 as shown in Table 5.5(a).

The mean value does not show clear significant difference in the first analysis. Therefore we perform additional analysis in which outliers are removed. From our observations of yeast images we have observed outlier cases such as very few large cells that might bias the measurement and cells with very high intensity values due to auto-fluorescence from a dead

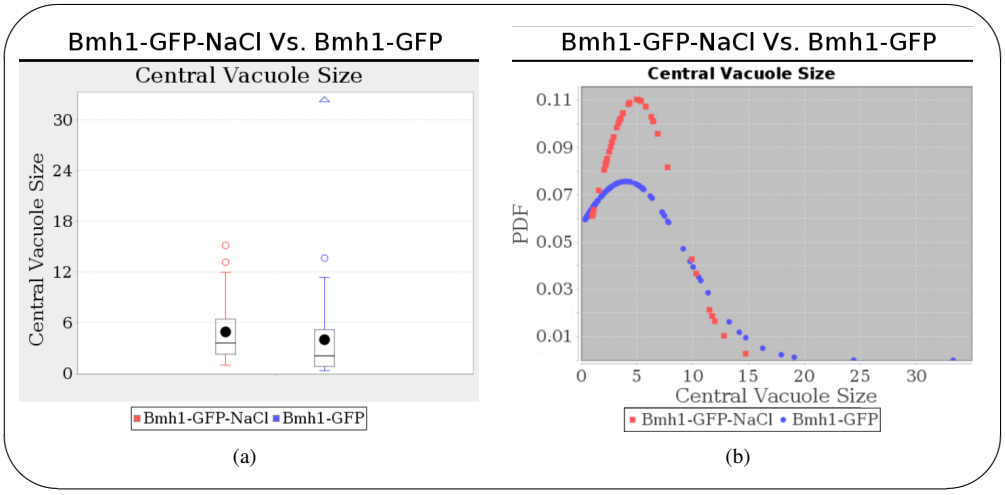


Figure 5.8: ***Central vacuole estimation in BMH1-GFP cells under salt stress.*** *(a) - Box and whiskers chart comparing BMH1-GFP under non-*NaCl *and* 0.5 M NaCl *medium. (b) - Fitting measurement in (a) to a Gaussian distribution.*

cell. Deviation from the mean is widely used to set outlier threshold, however, significant results could easily turnout to be false positives. In literature [Ley13], deviation from the median is proposed, as the median is very insensitive to the presence of outliers. The Median Absolute Deviation (*MAD*) is used as a way of dealing with the problem of outliers. To define a rejection criterion, we have adopted the recommended coefficient value of 2.5 [Ley13]. The recommended threshold value for outlier detection is at the median plus or minus 2.5 times the *MAD*. By removing these outliers, difference between the mean vacuole size in the two groups becomes more significant. This result is shown in Fig. 5.9 and Table 5.5(a). The estimated average vacuole size is increased from 2.3% to 3.6% under stress. The p-value from *Student* t-test is 0.002. Nevertheless, this analysis suggests that the central vacuole has a larger volume under salt stress. This is an interesting result, but what about all the vacuoles within the cells? Hence, we perform an additional experiment considering all estimated vacuoles.

In addition to the central vacuole we analyzed the number of vacuolar components and their total size per individual cells. Similarly to the central vacuole analysis, we compute the relative vacuole size; i.e. the vacuole size in terms of percentage of the cell size. Figure 5.10 illustrates typical yeast cells expressing Bmh1-GFP, where the central vacuole and all other vacuoles are segmented.

In this analysis we noticed interestingly, as it is clear from Fig. 5.11 that the number of vacuolar compartments significantly increased under NaCl salt stress in the *BMH1-GFP* strain. This increase is from an average of two vacuolar compartments under non-NaCl medium to five compartments under 0.5 M NaCl.

From Fig. 5.12 and Table 5.5(b), we can see that the total vacuoles size occupies in average 4.6% of the cell size in *BMH1-GFP* strain, increased to 8.7% in 0.5 M NaCl. The p-value from the *Student* t-test statistical analysis is less than 0.001 as shown in Table 5.5(b).

We performed an additional analysis by removing outliers as done with the analysis of central vacuoles. The result is shown in Fig. 5.13 and Table 5.5(b). The estimated average vacuoles size is increased from 3.1% to 7.0% under stress. The p-value from *Student* t-test

(a)

	-NaCl	+NaCl	P-value
Initial Analysis	3.5% ± 5.1 (131)	5.3% ± 4.1 (42)	0.021
No-Outliers	2.3 %± 2.0 (94)	3.6% ± 1.9 (34)	0.002

(b)

	-NaCl	+NaCl	P-value
Initial Analysis	4.6% ± 6.0 (131)	8.7% ± 4.9 (42)	< 0.001
No-Outliers	3.1% ± 2.5 (84)	7.0% ± 3.1 (34)	< 0.001

Table 5.5: *Analysis of vacuole size in* Bmh1 14-3-3 *protein under non-*NaCl *stress and* 0.5 M NaCl *stress level. The values represent the mean and standard deviation of the relative percentage of vacuole size related to the cell size. The p-value is computed from the Student t-test statistical analysis. The numbers in parentheses represent the number of cells in each condition. (a) - Descriptive analysis of central vacuole sizes before and after removal of outliers. (b) - Descriptive analysis of total vacuole sizes before and after removal of outliers.*

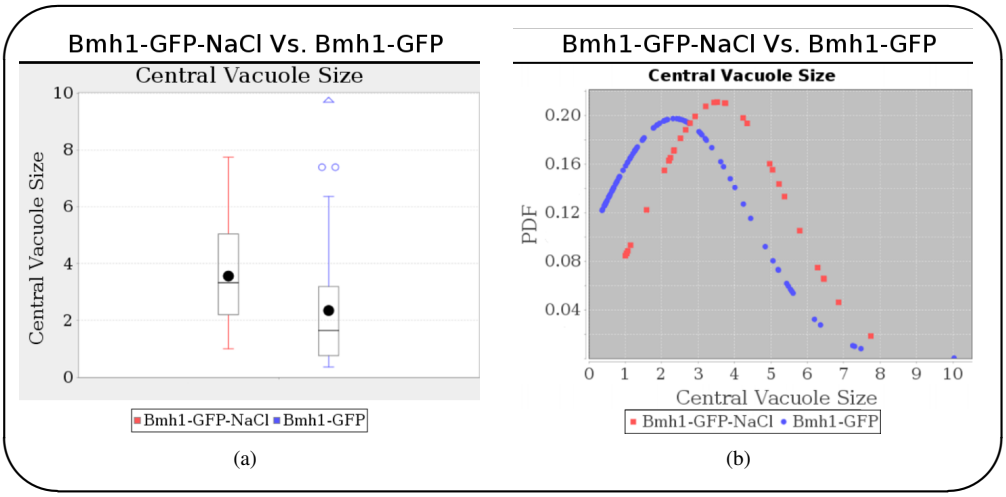


Figure 5.9: ***Central Vacuole estimation in BMH1-GFP under salt stress and after outliers removal.*** *(a) - Box and whiskers chart comparing BMH1-GFP under non-*NaCl *and* 0.5 M NaCl *medium. (b) - Fitting measurement in (a) to a Gaussian distribution.*

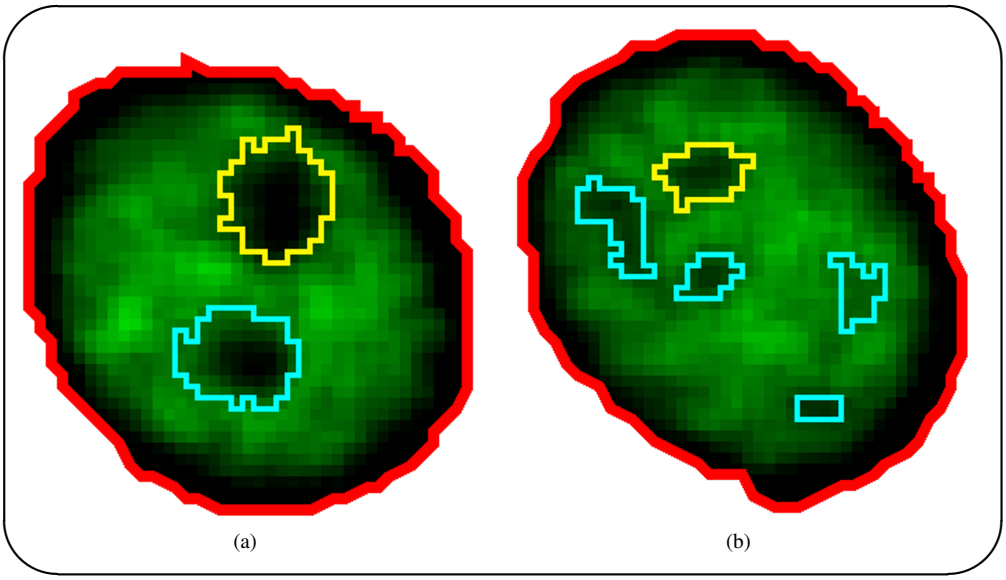


Figure 5.10: ***Estimation of vacuoles in sample yeast cells expressing*** Bmh1-GFP ***protein.*** *The detected central vacuole is marked with a yellow contour and all the other vacuoles are marked with cyan countors. (a) - Sample yeast cell with two vacuoles. (b) - A sample cell with one large central vacuole and multiple small vacuoles.*

is again less than 0.001. Nevertheless, this analysis suggests that also the total vacuoles has a larger volume under salt stress as well as the central vacuole. The result in this analysis is interesting. However, it requires more experiment validation. The increase in the number of

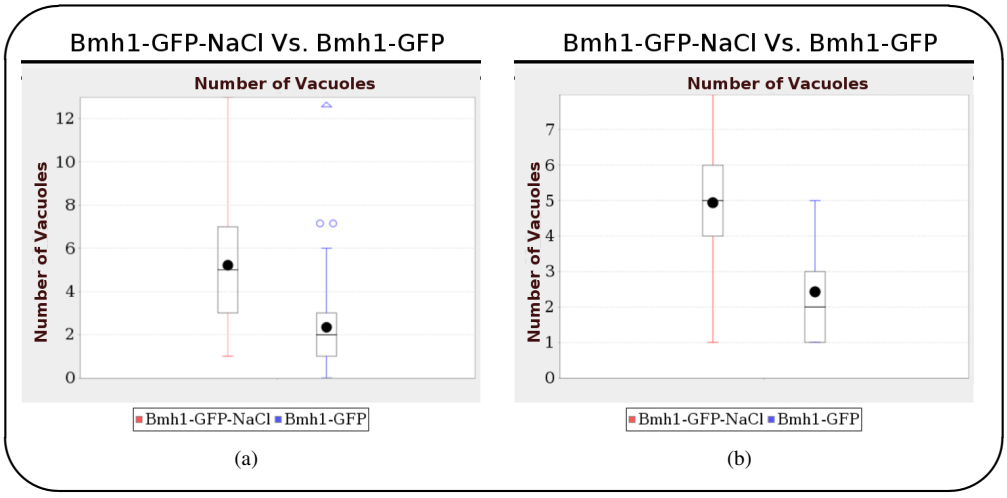


Figure 5.11: ***Number of Vacuolar Compartments in cells expressing* Bmh1** 14-3-3 ***proteins under salt stress.*** *(a) - Number of vacuoles in BMH1-GFP under non-*NaCl *and* 0.5 M NaCl *medium. (b) - Number of vacuoles in BMH1-GFP after outliers removal.*

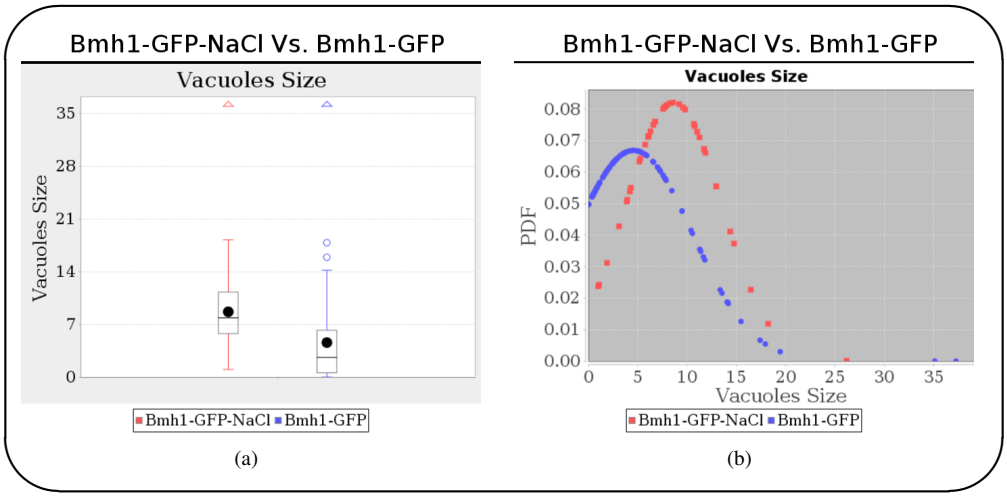


Figure 5.12: ***Total vacuole sizes estimation in BMH1-GFP cells under salt stress.*** *(a) - Box and whiskers chart comparing BMH1-GFP under non-*NaCl *and* 0.5 M NaCl *medium. (b) - Fitting measurement in (a) to a Gaussian distribution.*

vacuolar compartments might be a mechanism for water conservation in yeast cells. These small provacuoles in cells under salt treatment could have appeared as a result of dehydration. This result is consistent with another study done with plant root cells [Sán92]. However, the increase in total vacuole size and central vacuole size is opposite to what we expected. Because we expected the cellular volume to decrease in response to increase in external salinity. Hence we analyzed images acquired in three other experiments performed at different dates. Interestingly, in all the experiments the estimation of vacuoles size has increased under NaCl stress for *BMH1-GFP* strain with p-values from *Student t-test* being < 0.001, 0.03 and 0.3 with a fold change of 1.39, 1.19 and 1.12 respectively. The result of the repeated experiments for *BMH1-GFP* cells are depicted in Table 5.6. After outliers removal these numbers are shown in Table 5.7. The number of vacuoles seems to be always increasing with *BMH1-GFP* under NaCl stress, although not significantly in one experiment.

These results are unexpected. In order to validate if these results are correct, we propose to perform an experiment in which we stain the vacuoles with a bio-marker. From the literature [Rob91], there are different protocols available for the staining of the vacuoles. This experiment will help us to better understand the behaviour of cells under salt stress, and whether the number of vacuolar components and vacuole sizes increase as a mechanism that the cell adopt under stress. The stained vacuoles can be easily measured in *YeastAnalysis*, which has an option to measure separate channels of any stain/fluorescent label.

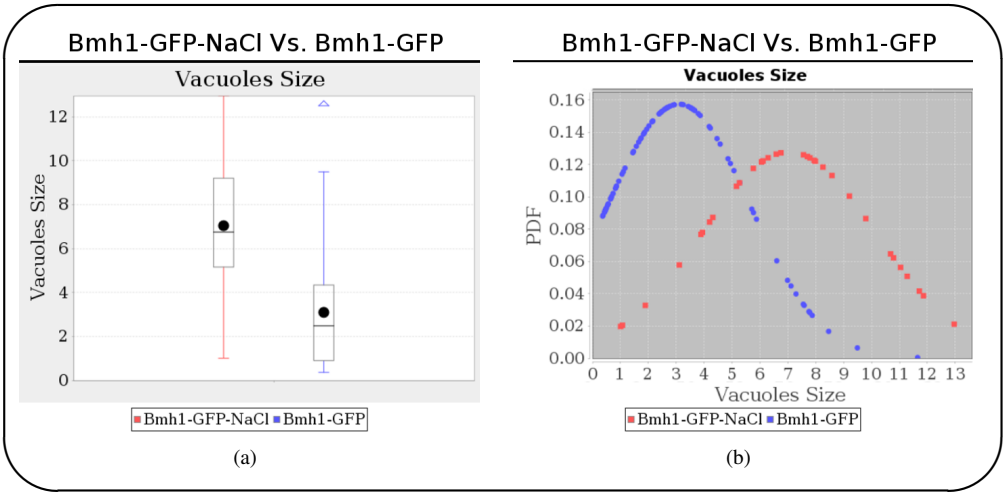


Figure 5.13: ***Total vacuole sizes estimation in BMH1-GFP cells under salt stress after outliers removal.*** *(a) - Box and whiskers chart comparing BMH1-GFP under non-*NaCl *and* 0.5 M NaCl *medium. (b) - Fitting measurement in (a) to a Gaussian distribution.*

Repetitions BMH1-GFP	Vacuoles Size		Size Fold Change +Nacl/-Nacl (P-Value)	Number Of Vacuolar Compartments		Vacuoles Fold Change +Nacl/-Nacl (P-Value)
	-NaCl	+NaCl		-NaCl	+NaCl	
Experiment 1	10.0 ± 4.4 (166)	13.9 ± 8.2 (155)	1.39 (P<0.001)	7.2 ± 4.0 (166)	8.1 ± 4.6 (155)	1.16 (P : 0.068)
Experiment 2	9.1 ± 5.1 (88)	10.8 ± 4.2 (86)	1.19 (P: 0.035)	5.4 ± 3.2 (88)	5.6 ± 2.7 (86)	1.04 (P : 0.717)
Experiment 3	6.7 ± 4.6 (74)	7.5 ± 4.0 (57)	1.12 (P: 0.324)	5.0 ± 3.8 (74)	5.3 ± 2.9 (57)	1.06 (P : 0.611)

Table 5.6: *Repeated experiments to determine vacuoles size and number of vacuolar compartments. The statistical method used to check the significant of the differences (P-Value) was the Student t-test. Numbers within parenthesis represents the number of cells used in the analysis.*

Repetitions BMH1-GFP	Vacuoles Size		Size Fold Change +Nacl/-Nacl (P-Value)	Number Of Vacuolar Compartments		Vacuoles Fold Change +Nacl/-Nacl (P-Value)
	-NaCl	+NaCl		-NaCl	+NaCl	
Experiment 1	9.7 ± 3.5 (139)	11.7 ± 4.5 (130)	1.21 (P<0.001)	6.6 ± 2.9 (139)	7.7 ± 3.8 (130)	1.17 (P : 0.008)
Experiment 2	7.7 ± 3.4 (70)	9.4 ± 3.7 (72)	1.22 (P:0.006)	5.0 ± 2.2 (70)	5.7 ± 2.6 (72)	1.14 (P : 0.075)
Experiment 3	6.1 ± 3.8 (58)	6.5 ± 2.8 (49)	1.07 (P: 0.585)	4.3 ± 2.6 (58)	5.0 ± 2.7 (49)	1.16 (P : 0.206)

Table 5.7: *After removal of outliers from the repeated experiments to determine vacuoles size and number of vacuolar compartments. The statistical method used to check the significant of the differences (P-Value) was the Student t-test. Numbers within parenthesis represents the number of cells used in the analysis.*

5.5 Conclusion

We have used *YeastAnalysis* to address the role of 14-3-3 proteins and the Nha1 cation transporter in the response of yeast cells to high salt concentrations. 14-3-3 proteins are highly conserved eukaryotic proteins binding to hundreds of different mostly phosphorylated proteins [for reviews see: [Mac04, Ait06, Mor09]]. In addition, the *S. cerevisiae* 14-3-3 proteins, encoded by *BMH1* and *BMH2*, bind to hundreds of phosphorylated proteins [Heu95, Kak07, Heu09] and play a role in the regulation of many processes including tolerance to NaCl [Pos00, Zah12]. Deletion of *BMH1*, encoding the major 14-3-3 isoform, is known to result in an increased sensitivity to Na^+, Li^+ and K^+ and to cationic drugs [Zah12]. Testing the genetic interaction between *BMH* genes and genes encoding plasma membrane cation transporters revealed a genetic interaction between *BMH1* and *NHA1*. These results show that the yeast 14-3-3 proteins and an alkali-metal cation efflux system interact and that this interaction enhances cell survival upon salt stress [Zah12]. To further understand the role of 14-3-3 proteins and the Nha1 antiporter in salt stress resistance, the effect of high external NaCl concentration on the levels of these proteins was studied here. Cultivation in the presence of 0.5 M NaCl resulted in an increased level of both Bmh1-GFP and Bmh2-GFP. This observation is in line with a role of 14-3-3 proteins in tolerance to high environmental NaCl. Moreover, analysis of vacuole sizes revealed larger vacuole size and increased number of vacuolar compartments under increased concentration of environmental NaCl; this result might be a mechanism that the cells employ under stress conditions in BMH1 – GFP cell strain. In order to validate if these results are correct, we propose to perform an experiment in which we stain the vacuoles with a bio-marker. Further analysis using sophisticated feature sets as discussed in Chapter 4 is possible, for example, to recognize the characteristic differences between all the cells cultured in increased concentration of environmental NaCl and those in low concentration of environmental NaCl. Such characteristic differences cannot be seen by standard descriptive analysis. However, such analysis was not relevant in the study performed in the current chapter.

6
Discussion

In this chapter, we will evaluate our research questions and make separate discussion for each of these research questions. In the first section, we specify the research problem and the research questions we addressed in this dissertation. In the following sections we discuss our image analysis pipeline, our segmentation algorithms, our machine learning approach to perform object recognition and pattern recognition using our combination of feature sets, and then the validation of our analysis system.

6.1 Research Problem and Questions

HERE we again clearly state our research problem (RP), which is at the center of our entire project and directly related to our goals and the associated research questions (RQs) that frame our research study. The answers to these questions creates an identifiable connections between them and the main research problem inspiring the study. Such connections is what makes our research meaningful. In the following sections we discuss these research questions.

- **RP:** There is no mature Pattern Recognition system to support objective analysis and phenotype characterization in single-cell image-based gene expression experiments.
- **RQ1:** Which components and processes are required to build a comprehensive image analysis pipeline for single-cell image based gene expression experiments?
- **RQ2:** Can a segmentation based on Hough-Transform and minimal path extraction algorithms improve the detection of ovoid-shaped objects in micro/cell biology images? Can we realize an optimization of the initial result using contour expansion algorithms?
- **RQ3:** What machine learning approach using sophisticated features can support in the object recognition process? and can it improve the identification of subtle patterns residing within the measurement data?
- **RQ4:** Can we study the role of 14-3-3 proteins and Nha1 antiporter based on imaging and image analysis? Do results from other analysis confirm our result and thereby validate our method?

6.2 Image Analysis Pipeline

RQ1: *Which components and processes are required to build a comprehensive image analysis pipeline for single-cell image based gene expression experiments?*

Fluorescent reporter proteins like GFP are widely used in biological research, also in research employing the model yeast *S. cerevisiae*. The analysis of these images is very time consuming and not completely objective. In this study we developed a novel and comprehensive image analysis platform for analysis of microscope images of cells expressing fluorescent proteins. Our application was focused on the model organism *S. cerevisiae* as a case study. The main outcome is a software platform that is currently used by our collaborators to assist them in their experiments.

The novelty of this system is the overall image analysis pipeline and the segmentation algorithm developed for yeast cell detection. This segmentation algorithm for bright-field channels has advantages over algorithms implemented in other software packages as additional staining is not required, the segmentation approach is optimized for *S. cerevisiae* cells and it is freely available.

To the best of our knowledge no similar platform exists, where the existing systems are not flexible as they either offer part of the pipeline such as the segmentation modules as in CellStat [Kva08] and CellSerpent [Bre11], or they are general and not able to offer segmentation for the image modalities that we have as in CellProfiler [Car06]. On the other hand, our proposed solution overcomes these issues and can be extendible and reusable. Our proposed system is a new and comprehensive tool that can generate measurement reports to confirm and validate experiments performed by biologists. *YeastAnalysis* system also extends previous work through providing a complete image analysis system instead of only parts of the pipeline. The strength of this platform relies on a novel segmentation algorithm, user friendly interface, automatic reports generation and selected features to be measured. The system also has flexibility in verifying the results, such as manual segmentation of individual cells, or manual correction of the measurement results and performing analysis of selected groups of cells, as well as choosing from various visualization charts. Such input from the user can be fed in a future work into the system to train it using the machine learning approach we developed. The already existing software packages that deal with *S. cerevisiae* yeast cells are usually developed for a specific task. For example, they are developed to achieve only the segmentation step [Kva08, Bre11, Pen13], to measure only a few features [Kva08, Bre11, Pen13], or to address a specific experiment [Maz13]. This complicates their ability to perform a complete analysis which is especially important for analysis of large datasets as required for systems biology. The *YeastAnalysis* platform with its novel segmentation algorithm and its complete pipeline is a promising tool for the analysis of large number of images generated in large scale experiments. This is further supported by the observation that the time required for segmenting an average 1024x1024 image with around 30 cells, using a quad core computer with *Ubuntu* operating system, is only around 5 seconds.

In conclusion, this platform contributes to improve the analysis of gene expression studies, it provides a comprehensive image analysis platform that can be used by cell biologists to analyze their experiments. This platform can be further improved and its usability can be extended in future work.

The model in Fig. 6.1 depicts an example of what can be done next to extend the usability of the system. This model can be used to help us know and understand the ideas we have for a future work. The primary objective of our model is to convey the fundamental principles and basic functionality of the system to be developed. It must be developed in such a way as to provide an easily understood system interpretation for the models users. In order to implement this model properly, it should satisfy four fundamental objectives [Str11]:

- Enhance an individual's understanding of the representative system.
- Facilitate efficient conveyance of system details between stakeholders.
- Provide a point of reference for system designers to extract system specifications.
- Document the system for future reference and provide a means for collaboration.

Our system is now a standalone application. However, for better maintainability, we can develop into a web application. In the model depicted in Fig. 6.1, a user can connect with its

account to a web application through a web interface. This application is connected to a network storage, where the users data are stored. Through this Application a user can perform either segmentation, measurement or data-analysis.

The segmentation module requires as input the images to be segmented and the segmentation method to be used in addition to the specific parameters for that method. The measurement module requires the binary masks or contour coordinates of the segmented images and as parameters all the features required to be measured and the image channels to be measured. The output of this module would be a CSV file holding all the individual cell objects measured for the requested features. The data analysis module requires the CSV file generated by a measurement process and keywords specifying what cell groups to be analyzed and what type of visualization charts to be output into the final PDF report. These steps can be performed one by one. However, when preferred, the parameters for all the modules can be fed in advance into an additional batch module that performs all these steps sequentially and generate automatically the required report.

At the lowest level in the model there is a Cell Analysis API that can communicate with each of the mentioned modules through a web services description language (WSDL). In future implementation, this model will play an important role in the overall system development life cycle.

6.3 Ovoid Objects Segmentation

RQ2: *Can a segmentation based on Hough-Transform and minimal path extraction algorithms improve the detection of ovoid-shaped objects in micro/cell biology images? Can we realize an optimization of the initial result using contour expansion algorithms?*

The segmentation algorithm is an improvement of an algorithm developed in an initial work [Kva08] by increasing the detection rate and the accuracy of the detected contours [Tle14]. Our approach consists of two steps, the first uses the Hough Transform to locate circular objects in an image, the second step is fine-tuning each detected object and extracting its exact contour where the center of the circle detected in the first step is taken as a seed point. From that point, a polar representation of the object (referred to as polar image) is generated and an algorithm is applied to extract the exact contour of the object by determining the minimal path from the first to the last column in the polar image.

The detection rate is dependent on the quality of the images. Images containing out-of-focus cells and images containing debris and dead cells are more difficult to segment. Furthermore, during confocal imaging, the intensity of the fluorescence drops dramatically when cells are out-of focus, making quantification difficult. In the case study on the role of 14-3-3 proteins and Nha1 antiporter mentioned in Chapter 5 the segmentation method was able to detect 92 percent of the cells. Additionally, the non-detected cells could be segmented using an interactive GUI by manually seeding (mouse click within) the cells. Detected artefacts such as debris or dead cells could easily be removed manually as well.

The novelty of this method is the threshold equations presented to control the Hough-

Transform to detect circular objects; in addition to the extraction of the circular shortest path method from the polar representation of the image object. Our novel segmentation method is better than the other methods and offers higher detection rate and more accurate contours. In a validation experiment the detection rate obtained an F1-score of 0.93 outperforming similar methods.

Since microscope settings are delicate and contours are biased to the inner part of the objects, the novel expansion algorithm comes into play. In one experiment the expansion algorithm could improve the accuracy of the extraction contour with an F1-score of 0.92 and a Pratt score of 0.58 outperforming other methods.

The strength of our segmentation algorithm is represented by the higher detection rate and the more precise contours compared with other methods. It is much better than many algorithms such as heavy active-contour based algorithms [Bre11]. The improved segmentation level means consequently an improved accuracy in the analysis of the measurement.

In conclusion, we presented a novel segmentation algorithm to extract the contours of ovoid-shaped and circular objects by using a variety of Hough Transform and a minimal path

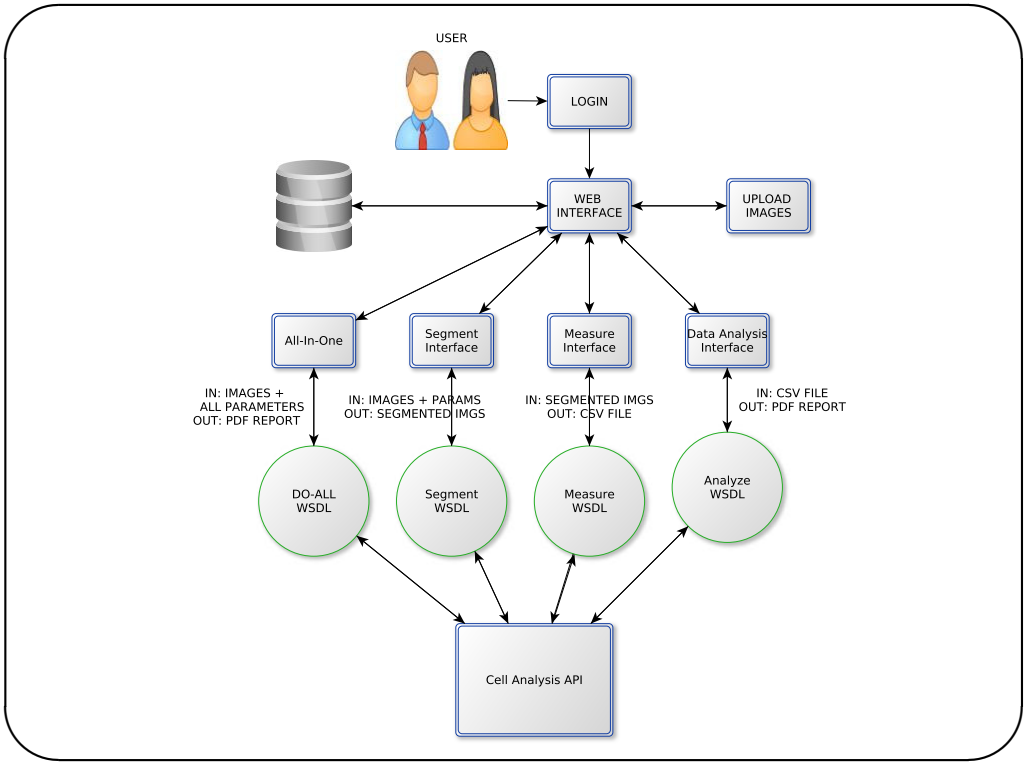


Figure 6.1: ***Model for further platform development.*** *At the lowest level in this model there is a cell analysis API that communicates with several modules through web services description language (WSDL). At the top level, users can interact with these modules through a web interface.*

algorithm. In addition to a contour expansion algorithm that optimizes the extracted contour. As a future work we can apply parallel programming to make a real-time detection method and analysis a possibility. We can also improve the detection by using noise removal algorithms and segmentation that uses the information from all the image channels.

6.4 Machine Learning and Feature Sets

RQ3: *What machine learning approach using sophisticated features can support in the object recognition process? and can it improve the identification of subtle patterns residing within the measurement data?*

The popular feature sets we chose to describe the single-cell objects has played a significant role in raising the performance of the classification models. The Wavelet-based texture measurement has shown their superiority to discriminate cell objects in the top classifiers. In addition, the fusion of invariant moments and wavelet details in many dimensions obtained high weights in some classifiers such as Simple Logistics and Linear Model Trees revealing their discriminative power. The second order histogram features extracted from the co-occurrence matrix has also played a role in the discrimination of single-cell objects. However, a complete feature set gained the best classification results on two different datasets. The complete feature set combines features from basic shape descriptors, invariant moments, wavelet texture measurement, invariant moments on wavelet detail images, co-occurrence matrix derived features, basic texture measurement. The *AUC* metric was the main evaluation criteria used. The complete feature set got an *AUC* value of 0.91 and 0.92 on the two different datasets respectively. This suggests that the presentation of complex feature sets to describe the single-cell characteristic and morphology in a more sophisticated way is an advantageous step in building classification models that aims to recognize objects and identify subtle patterns.

The use of wavelet-based texture measurement, invariant moments and second order histogram features are not new in describing texture measurement in bioimaging. However, the novelty of our work is to combine these feature sets to describe the objects in addition to our novel approach to fuse moment invariants and wavelet details. Our results show optimal classification when combining various feature sets.

The strengths of these features is their ability to describe objects characteristics and morphology is a more sophisticated way enabling its recognition and identification of subtle patterns residing within these objects. As in many disciplines, there is a drawback. This drawback is the complexity of the constructed classification models based on these features. This makes it very difficult to describe the underlying processes of the classification model. Despite this difficulty, the classification results were excellent and the constructed model could easily classify the cell objects into its correct group when compared to groundtruth.

The feature sets contribute to the improvement of classifying single-cell objects and consequently improving cell object recognition and identification of subtle patterns residing within the measurement data; i.e. recognizing the patterns that are not obvious and hard to notice within

standard measurement methods. Nevertheless, further research is still possible. Interesting research could be to try more techniques and feature sets known to be used with texture measurement and trying to fuse different methods together. For example, if we where to redo this research we would consider methods based on Zernike moments, Discrete Cosine Transform and Linear Binary Patterns, as these methods are mentioned in the literature and are successful in texture retrieval [Sim04, Bae97, Mäe03]. Moreover, an interesting research topic would be to use fewer features that can gain a similar classification power. This would enable a more easy understanding of the underlying process that is generating the data.

Using the aforementioned features, we presented a machine learning system that was able to discriminate cell objects to identify essential characteristic differences between two groups treated under different conditions, and a similar approach that improves the object recognition based on the object's measured sophisticated features. This implies that we can apply the same approach in further biological studies and especially in high-throughput screening.

Machine learning systems are gaining popularity within bioimaging and biomedical studies because as models are exposed to new data, they are able to independently adapt. They learn from previous computations to produce reliable, repeatable decisions and results. The idea of applying machine learning is not a new idea by itself. However, our novel machine learning workflow is the combination of selected sophisticated and complex features, the sampling of the dataset using *SMOTE* method (cf. Section 4.3.1) after measuring individual features, normalization of the data features using the *MV* scheme in addition to the evaluation of numerous linear and non-linear classification approaches in an automatic way to elect a classifier to construct our model.

The strength of this approach resides in the selected features as a powerful discriminative power. In addition, the evaluation of the best existing classification models assures the selection of the best model to be used as our classifier. One issue with this system is the parameters that most models utilize. With default parameters, an excellent classifier might not show its strength as we showed for the *SVM* classifier in Chapter 4. Despite this fact, the performance of the classification models were superior with an excellent classification level, i.e $AUC \geq 0.90$.

With such machine learning approach we expect an easier identification of objects and patterns within these objects. This fact will contribute to improve the understandability of protein and gene behaviours in cell/micro biological studies. Further work in this field is still possible. Applying developmental techniques to create an optimal classifier is an interesting domain. Moreover, an automatic parameter optimization and selection for the existing popular classifiers is also a possibility for a future work. In addition, adding more feature sets as mentioned in the previous section and testing the classifiers on global existing datasets would be an important project. Next to that, an implementation of such approach in a real-time image analysis might have an additional value to biologists performing there experiments.

To summarize the discussion, we state that a feature set based on wavelet-based texture measurements, moment invariants, first and second order histogram features, basic texture measurement and fusion of moment invariants with wavelet details has an optimal classifying results in classifying single-cell based gene expression data. In addition, we offered a reusable machine learning workflow that offers an excellent performance to recognize biological objects

and identify subtle patterns within the measurement of these objects. There is still more work to explore, and the application of such approach in a highthroughput study is the obvious thing to do next.

6.5 Image Based Proteomics

RQ4: *Can we study the role of* 14-3-3 *proteins and* Nha1 *antiporter based on imaging and image analysis? Do results from other analysis confirm our result and thereby validate our method?*

We have used *YeastAnalysis* to address the role of 14-3-3 proteins and the *Nha1* transporter in the response of yeast cells under salt stress. 14-3-3 proteins are highly conserved eukaryotic proteins binding to hundreds of different mostly phosphorylated proteins. Also the *S. cerevisiae* 14-3-3 proteins, encoded by *BMH1* and *BMH2*, bind to hundreds of phosphorylated proteins [Heu95, Kak07, Heu09] and play a role in the regulation of many processes including tolerance to NaCl [Pos00, Zah12]. Deletion of *BMH1*, encoding the major 14-3-3 isoform, resulted in an increased sensitivity to Na^+, Li^+ and K^+ and to cationic drugs [Zah12]. Testing the genetic interaction between BMH genes and genes encoding plasma membrane cation transporters revealed a genetic interaction between *BMH1* and *NHA1*. In addition, using bimolecular fluorescence complementation (BiFC) a physical interaction between 14-3-3 proteins and the Nha1 antiporter was shown. These results show that the yeast 14-3-3 proteins and an alkali-metal cation efflux system interact and that this interaction enhances cell survival upon salt stress [Zah12]. To further understand the role of 14-3-3 proteins and the Nha1 antiporter in salt stress resistance, the effect of high external NaCl concentration on the levels of these proteins is studied here. Cultivation in the presence of 0.5 M NaCl resulted in an increased level of both Bmh1-GFP and Bmh2-GFP. This observation is in line with a role of 14-3-3 proteins in tolerance to high environmental NaCl.

The results obtained by *YeastAnalysis* were further validated using Flow Cytometric analysis, which were in good agreement. This validation suggests the beneficial advantage of using this system in further gene expression analysis experiments.

Bibliography

[Abd79] I. Abdou and W. Pratt. Quantitative design and evaluation of enhancement/thresholding edge detectors. *Proceedings of the IEEE*, 67(5):753–763, 1979. (cited on page 53).

[Aha91] D. Aha and D. Kibler. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991. (cited on page 72).

[Ait06] A. Aitken. 14-3-3 proteins: a historic overview. In *Seminars in cancer biology*, volume 16, pages 162–172. Elsevier, 2006. (cited on page 102).

[Ama08] P. P. Amaral, M. E. Dinger, T. R. Mercer, and J. S. Mattick. The eukaryotic genome as an rna machine. *Science*, 319(5871):1787–1789, 2008. (cited on page 3).

[apa16] Apache commons mathematics, 2016. http://commons.apache.org/proper/commons-math/. (cited on page 32).

[App03] B. Appleton and C. Sun. Circular shortest paths by branch and bound. *Pattern Recognition*, 36(11):2513 – 2520, 2003. (cited on pages 41 and 44).

[Ara15] A. Arasu, S. Chaudhuri, Z. Chen, K. Ganjam, R. Kaushik, and V. Narasayya. Data cleaning, 2015. http://research.microsoft.com/en-us/projects/datacleaning. (cited on page 13).

[Atk96] C. Atkeson, A. Moore, and S. Schaal. Locally weighted learning. *AI Review*, 1996. (cited on page 72).

[Bae97] H.-J. Bae and S.-H. Jung. Image retrieval using texture based on dct. In *Proceedings of 1997 International Conference on Information, Communications and Signal Processing*, pages 1065–1068. IEEE, 1997. (cited on page 109).

[Ban13] P. Bankhead. *Analyzing fluorescence microscopy images with ImageJ*. Nikon Imaging Center, Heidelberg University, Queen's University Belfast, June 2013. (cited on page 9).

[Bar95] A. Baraldi and F. Parmiggiani. An investigation of the textural characteristics associated with gray level cooccurrence matrix statistical parameters. *IEEE Transactions on Geoscience and Remote Sensing*, 33(2):293–304, 1995. (cited on page 12).

[Bes10] T. Beshah and S. Hill. Mining road traffic accident data to improve safety: Role of road-related factors on accident severity in ethiopia. In *AAAI Spring Symposium: Artificial Intelligence for Development*. Citeseer, 2010. (cited on page 70).

[BIP08] I. BIPM, I. IFCC, I. IUPAC, and O. ISO. The international vocabulary of metrology—basic and general concepts and associated terms (vim). *JCGM (Joint Committee for Guides in Metrology)*, 2008. (cited on page 11).

[Bir11] P. W. Bircham, D. R. Maass, C. A. Roberts, P. Y. Kiew, Y. S. Low, M. Yegambaram, J. Matthews, C. A. Jack, and P. H. Atkinson. Secretory pathway genes assessed by high-throughput microscopy and synthetic genetic array analysis. *Molecular BioSystems*, 7(9):2589–2598, 2011. (cited on page 18).

[Bla99] W. P. Blackstock and M. P. Weir. Proteomics: quantitative and physical mapping of cellular proteins. *Trends in biotechnology*, 17(3):121–127, 1999. (cited on page 5).

[Bla00] Y. M. Blanter and M. Büttiker. Shot noise in mesoscopic conductors. *Physics reports*, 336(1):1–166, 2000. (cited on page 56).

[Boa09] I. Boaventure and A. Gonzaga. Method to evaluate the performance of edge detector. In *Proceedings of the The Brazilian Symposium on Computer Graphics and Image Processing*, 2009. (cited on page 53).

[Bra11] D. Bratosin and M. Sidoroff. New horizons in flow cytometry by cytomics and bioinformatics. a minireview. *Studia Universitatis Vasile Goldis Seria Stiintele Vietii (Life Sciences Series)*, 2011. (cited on page 2).

[Bre01] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. (cited on page 72).

[Bre11] K. Bredies and H. Wolinski. An active-contour based algorithm for the automated segmentation of dense yeast populations on transmission microscopy images. *Computing and Visualization in Science*, 14(7):341–352, 2011. (cited on pages 18, 105 and 107).

[Bro97] R. G. Brown and A. M. Gleason. *Advanced Mathematics: Precalculus with Discrete Mathematics and Data Analysis*. McDougal Littell, Evanston, Illinois, 1997. (cited on page 37).

[Buc97] M. Buckley and J. Yang. Regularised shortest-path extraction. *Pattern Recognition Letters*, 18(7):621 – 629, 1997. (cited on page 40).

[Cao14] L. Cao. *Biological model representation and analysis*. PhD thesis, Section imaging and bioinformatics, Liacs, Leiden University, Nov. 2014. (cited on pages 64 and 66).

[Car06] A. E. Carpenter, T. R. Jones, M. R. Lamprecht, C. Clarke, I. H. Kang, O. Friman, D. A. Guertin, J. H. Chang, R. A. Lindquist, J. Moffat, et al. Cellprofiler: image analysis software for identifying and quantifying cell phenotypes. *Genome biology*, 7(10):R100, 2006. (cited on pages 51 and 105).

[Car12] D. M. Carabias. *Analysis of image thresholding methods for their application to augmented reality enviroments*. PhD thesis, Facultad de informatica, Universidad computense de Madrid, June 2012. (cited on page 23).

[Cat13] P. Cattin. Image restoration: Introduction to signal and image processing. *MIAC, University of Basel*, 11, 2013. (cited on page 56).

[Ces92] S. le Cessie and J. C. van Houwelingen. Ridge estimators in logistic regression. *Applied Statistics*, 41(1):191–201, 1992. (cited on page 72).

[Cha94] M. Chalfie, Y. Tu, G. Euskirchen, W. W. Ward, and D. C. Prasher. Green fluorescent protein as a marker for gene expression. *Science*, 263(5148):802–805, 1994. (cited on pages 5 and 6).

[Cha06] S. Chabrier, H. Laurent, C. Rosenberger, and Y.-J. Zhang. *14th European Signal Processing Conference (EUSIPCO)*, 2006. (cited on page 52).

[Cha09] M. Chalfie. Gfp: lighting up life (nobel lecture). *Angewandte Chemie International Edition*, 48(31):5603–5611, 2009. (cited on pages 5 and 6).

[Chi13] B. F. Chimieski and R. D. R. Fagundes. Association and classification data mining algorithms comparison over medical datasets. *Journal of Health Informatics*, 5(2), 2013. (cited on page 71).

[Cho15] Y. T. Chong, J. L. Koh, H. Friesen, K. Duffy, M. J. Cox, A. Moses, J. Moffat, C. Boone, and B. J. Andrews. Yeast proteome dynamics from single cell imaging and automated analysis. *Cell*, 161(6):1413–1424, 2015. (cited on page 18).

[Coh95] W. W. Cohen. Fast effective rule induction. In *Twelfth International Conference on Machine Learning*, pages 115–123. Morgan Kaufmann, 1995. (cited on page 72).

[dic16] The free on-line dictionary of computing, 2016. http://dictionary.reference.com/browse/pattern recognition. (cited on page 8).

[Don13a] J. P. Donate, P. Cortez, G. G. Sanchez, and A. S. De Miguel. Time series forecasting using a weighted cross-validation evolutionary artificial neural network ensemble. *Neurocomputing*, 109:27–32, 2013. (cited on page 70).

[Don13b] A. Doncic, U. Eser, O. Atay, and J. M. Skotheim. An algorithm to automate yeast segmentation and tracking. *Plos One*, 8(3):e57970. doi:10.1371/journal.pone.0057970, 2013. (cited on pages 8 and 11).

[Faw06] T. Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006. (cited on pages 66 and 70).

[Fel10] H. Feldmann. *Yeast. Molecular and Cell Biology*. Willey-BlackWell, 2010. (cited on pages 7 and 34).

[Fer99] A. Ferrand and M. Pages. Image management in sport organisations: the creation of value. *European journal of Marketing*, 33,3/4:387–402, 1999. (cited on page 10).

[Few04] S. Few. Enie, meenie, minie, moe: selecting the right graph for your message. *Intelligent Enterprise*, 2004. (cited on page 13).

[Fra98] E. Frank and I. H. Witten. Generating accurate rule sets without global optimization. In J. Shavlik, editor, *Fifteenth International Conference on Machine Learning*, pages 144–151. Morgan Kaufmann, 1998. (cited on page 72).

[Fre96] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Thirteenth International Conference on Machine Learning*, pages 148–156, San Francisco, 1996. Morgan Kaufmann. (cited on page 72).

[Fre98] Y. Freund and R. E. Schapire. Large margin classification using the perceptron algorithm. In *11th Annual Conference on Computational Learning Theory*, pages 209–217, New York, NY, 1998. ACM Press. (cited on page 72).

[fre16] The free dictionary, 2016. http://encyclopedia2.thefreedictionary.com/speckle. (cited on page 56).

[Fri97] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine learning*, 29(2-3):131–163, 1997. (cited on page 72).

[Fri98] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. Technical report, Standford University, 1998. (cited on page 72).

[Fri08] V. Friedman. Data visualization and infographics. *Graphics, Monday Inspiration*, 14, 2008. (cited on page 13).

[Ger86] J. J. Gerbrands, E. Backer, and W. Van Der Hoeven. Quantitative evaluation of edge detection by dynamic programming. In *Pattern Recognition in Practice II*, pages 91–99. Amsterdam, The Netherlands, 1986. (cited on page 44).

[Gha03] S. Ghaemmaghami, W.-K. Huh, K. Bower, R. W. Howson, A. Belle, N. Dephoure, E. K. O'Shea, and J. S. Weissman. Global analysis of protein expression in yeast. *Nature*, 425(6959):737–741, 2003. (cited on page 7).

[Gie95] R. D. Gietz, R. H. Schiestl, A. R. Willems, and R. A. Woods. Studies on the transformation of intact yeast cells by the liac/ss-dna/peg procedure. *Yeast*, 11(4):355–360, 1995. (cited on page 85).

[git16] Yeastanalysis source code on gitlab, 2016. https://gitlab.liacs.nl/mtleis/YeastAnalysis. (cited on page 32).

[God05] W. Godfrey, D. Hill, J. Kilgore, G. Buller, J. Bradford, D. Gray, I. Clements, K. Oakleaf, J. Salisbury, M. Ignatius, et al. Complementarity of flow cytometry and fluorescence microscopy. *Microscopy and Microanalysis*, 11(S02):246–247, 2005. (cited on page 5).

[Gon08] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Pearson Education, third edition, 2008. (cited on pages 10, 11, 21, 22, 23, 25, 26, 27, 34, 62, 63, 86 and 87).

[Goo96] L. van Gool, T. Moons, and D. Ungureanu. Affine/photometric invariants for planar intensity patterns. In *Computer Vision—ECCV'96*, pages 642–651. Springer, 1996. (cited on page 79).

[Gov11] P. Govind. Model organisms used in molecular biology or medical research. *International Research Journal of Pharmacy*, 2011. (cited on page 7).

[Gub09] J. Gubbi, S. Marusic, and M. Palaniswami. Smoke detection in video using wavelets and support vector machines. *Fire Safety Journal*, 44(8):1110–1115, 2009. (cited on pages 12 and 62).

[Han09] D. J. Hand. Measuring classifier performance: a coherent alternative to the area under the roc curve. *Machine learning*, 77(1):103–123, 2009. (cited on page 71).

[Han10] B. Hanczar, J. Hua, C. Sima, J. Weinstein, M. Bittner, and E. R. Dougherty. Small-sample precision of roc-related estimates. *Bioinformatics*, 26(6):822–830, 2010. (cited on page 71).

[Har79] R. M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, 1979. (cited on pages 62 and 64).

[He09] H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009. (cited on page 68).

[Hei96] R. Heim and R. Y. Tsien. Engineering green fluorescent protein for improved brightness, longer wavelengths and fluorescence resonance energy transfer. *Current Biology*, 6(2):178–182, 1996. (cited on page 6).

[Her88] I. Herskowitz. Life cycle of the budding yeast saccharomyces cerevisiae. *Microbiological reviews*, 52(4):536, 1988. (cited on page 7).

[Her11] C. Herder, J. Baumert, A. Zierer, M. Roden, C. Meisinger, M. Karakas, L. Chambless, W. Rathmann, A. Peters, W. Koenig, et al. Immunological and cardiometabolic risk factors in the prediction of type 2 diabetes and coronary events: Monica/kora augsburg case-cohort study. *PLoS One*, 6(6):e19852, 2011. (cited on page 71).

[Heu95] G. P. H. van Heusden, D. J. Griffiths, J. C. Ford, P. A. Schrader, A. M. Carr, H. Y. Steensma, et al. The 14-3-3 proteins encoded by the bmh1 and bmh2 genes are essential in the yeast saccharomyces cerevisiae and can be replaced by a plant homologue. *European journal of biochemistry*, 229(1):45–53, 1995. (cited on pages 102 and 110).

[Heu09] G. P. H. van Heusden. 14-3-3 proteins: insights from genome-wide studies in yeast. *Genomics*, 94(5):287–293, 2009. (cited on pages 102 and 110).

[Ho98] T. K. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998. (cited on page 72).

[Hol93] R. C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine learning*, 11(1):63–90, 1993. (cited on page 72).

[Hou13] X. Hou, A. Yuille, and C. Koch. Boundary detection benchmarking: Beyond f-measures. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2123–2130, 2013. (cited on pages 53 and 54).

[Hu62] M.-K. Hu. Visual pattern recognition by moment invariants. *IEEE Transactions on Information Theory*, 8(2):179–187, 1962. (cited on page 63).

[Hub07] D. W. Hubbard. How to measure anything. *Finding the Value of "Intangibles" in Business*, 2007. (cited on page 11).

[Huh03] W.-K. Huh, J. V. Falvo, L. C. Gerke, A. S. Carroll, R. W. Howson, J. S. Weissman, and E. K. O'Shea. Global analysis of protein localization in budding yeast. *Nature*, 425(6959):686–691, 2003. (cited on pages 5, 7, 84 and 94).

[Hul01] G. Hulten, L. Spencer, and P. Domingos. Mining time-changing data streams. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 97–106. ACM Press, 2001. (cited on page 72).

[Iba92] W. Iba and P. Langley. Induction of one-level decision trees. In *Proceedings of the ninth international conference on machine learning*, pages 233–240, 1992. (cited on page 72).

[Inc12] Z. Inc. Zeiss 5 exciter laser scanning microscope manual, 2012. (cited on page 49).

[ite16] itext programmable pdf, 2016. http://itextpdf.com/. (cited on page 32).

[Jaf05] K. Jafari-Khouzani and H. Soltanian-Zadeh. Rotation-invariant multiresolution texture analysis using radon and wavelet transforms. *IEEE Transactions on Image Processing*, 14(6):783–795, 2005. (cited on page 63).

[Jah12] R. R. Jahan-Tigh, C. Ryan, G. Obermoser, and K. Schwarzenberger. Flow cytometry. *Journal of Investigative Dermatology*, 132(10):e1, 2012. (cited on page 5).

[Jan04] C. Janke, M. M. Magiera, N. Rathfelder, C. Taxis, S. Reber, H. Maekawa, A. Moreno-Borchart, G. Doenges, E. Schwob, E. Schiebel, et al. A versatile toolbox for pcr-based tagging of yeast genes: new fluorescent proteins, more markers and promoter substitution cassettes. *Yeast*, 21(11):947–962, 2004. (cited on page 85).

[jex16] Java excel api, 2016. http://jexcelapi.sourceforge.net/. (cited on page 32).

[jfr16] Jfreechart api, 2016. http://www.jfree.org/jfreechart/. (cited on page 32).

[Jud11] C. M. Judd, G. H. McClelland, and C. S. Ryan. *Data analysis: A model comparison approach*. Routledge, 2011. (cited on page 13).

[Kae05] M. Kaeberlein, R. W. Powers, K. K. Steffen, E. A. Westman, D. Hu, N. Dang, E. O. Kerr, K. T. Kirkland, S. Fields, and B. K. Kennedy. Regulation of yeast replicative life span by tor and sch9 in response to nutrients. *Science*, 310(5751):1193–1196, 2005. (cited on page 7).

[Kae10] M. Kaeberlein. Lessons on longevity from budding yeast. *Nature*, 464(7288):513–519, 2010. (cited on page 7).

[Kak07] K. Kakiuchi, Y. Yamauchi, M. Taoka, M. Iwago, T. Fujita, T. Ito, S.-Y. Song, A. Sakai, T. Isobe, and T. Ichimura. Proteomic analysis of in vivo 14-3-3 interactions in the yeast saccharomyces cerevisiae. *Biochemistry*, 46(26):7781–7792, 2007. (cited on pages 102 and 110).

[Kam13] I. Kaminer, J. Nemirovsky, and M. Segev. Optimizing 3d multiphoton fluorescence microscopy. *Optics letters*, 38(19):3945–3948, 2013. (cited on page 4).

[Key01] L. Keyes and A. Winstanley. Using moment invariants for classifying shapes on large-scale maps. *Computers, Environment and Urban Systems*, 25(1):119–130, 2001. (cited on page 63).

[Kim11] S. Kim, H. Zhang, R. Wu, and L. Gong. Dealing with noise in defect prediction. In *33rd. International Conference on Software Engineering (ICSE)*, pages 481–490. IEEE, 2011. (cited on page 70).

[Koc01] M. Kociołek, A. Materka, M. Strzelecki, and P. Szczypiński. Discrete wavelet transform–derived features for digital image texture analysis. In *Proceedings of Interational Conference on Signals and Electronic Systems*, pages 163–168, 2001. (cited on page 64).

[Koh95] R. Kohavi. The power of decision tables. In *Machine Learning: ECML-95*, pages 174–189. Springer, 1995. (cited on page 72).

[Kva08] M. Kvarnstrom, K. Logg, A. Diez, K. Bodvard, and M. Kall. Image analysis algorithms for cell contour recognition in budding yeast. *Optics Express*, 16(17):12–943, 2008. (cited on pages 18, 39, 54, 105 and 106).

[Lah09] M. Lahtinen, K. Holli, L. Harrison, P. Dastidar, S. Soimakallio, and H. Eskola. Software phantoms for texture analysis. In *World Congress on Medical Physics and Biomedical Engineering*, pages 308–311, Munich, Germany, 2009. Springer. (cited on page 64).

[Lak11] V. Lakafosis, A. Traille, H. Lee, E. Gebara, M. M. Tentzeris, G. DeJean, and D. Kirovski. Rfid-coa: The rfid tags as certificates of authenticity. In *IEEE International Conference on RFID (RFID)*, pages 207–214. IEEE, 2011. (cited on page 70).

[Lan05] N. Landwehr, M. Hall, and E. Frank. Logistic model trees. *Machine Learning*, 59(1-2):161–205, 2005. (cited on page 72).

[Lee83] J.-S. Lee. Digital image smoothing and the sigma filter. *CVGIP(24)*, No. 2:255–269, 1983. (cited on page 21).

[Ler93] R. Lerski, K. Straughan, L. Schad, D. Boyce, S. Blüml, and I. Zuna. Viii. mr image texture analysis—an approach to tissue characterization. *Magnetic resonance imaging*, 11(6):873–887, 1993. (cited on page 64).

[Ley13] C. Leys, O. Klein, P. Bernard, and L. Licata. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology*, 49(4):764–766, 2013. (cited on page 97).

[Li14] S. Li, Y. Zhang, J. Xu, L. Li, Q. Zeng, L. Lin, Z. Guo, Z. Liu, H. Xiong, and S. Liu. Noninvasive prostate cancer screening based on serum surface-enhanced raman spectroscopy and support vector machine. *Applied Physics Letters*, 105(9):091104, 2014. (cited on page 70).

[Lob08] J. M. Lobo, A. Jiménez-Valverde, and R. Real. Auc: a misleading measure of the performance of predictive distribution models. *Global ecology and Biogeography*, 17(2):145–151, 2008. (cited on page 71).

[Mac04] C. Mackintosh. Dynamic interactions between 14-3-3 proteins and phosphoproteins regulate diverse cellular processes. *Biochemistry*, 381:329–342, 2004. (cited on page 102).

[Mäe03] T. Mäenpää. *The Local binary pattern approach to texture analysis: Extenxions and applications*. Oulun yliopisto, 2003. (cited on page 109).

[Mar04] J. Martinez-Urtaza, J. Peiteado, A. Lozano-León, and O. Garcia-Martin. Detection of salmonella senftenberg associated with high saline environments in mussel processing facilities. *Journal of Food Protection®*, 67(2):256–263, 2004. (cited on page 84).

[Mas01] B. Masters and P. So. Confocal microscopy and multi-photon excitation microscopy of human skin in vivo. *Optics express*, 8(1):2–10, 2001. (cited on page 4).

[Mat98] A. Materka, M. Strzelecki, et al. Texture analysis methods–a review. *Technical university of lodz, institute of electronics, COST B11 report, Brussels*, pages 9–11, 1998. (cited on pages 12, 62, 63 and 64).

[Maz13] A. Mazumder, L. Q. Pesudo, S. McRee, M. Bathe, and L. D. Samson. Genome-wide single-cell-level screen for protein abundance and localization changes in response to dna damage in s. cerevisiae. *Nucleic acids research*, page gkt715, 2013. (cited on pages 18 and 105).

[Mic99] J. Michell. *Measurement in psychology: A critical history of a methodological concept*, volume 53. Cambridge University Press, New York, 1999. (cited on page 11).

[Mor09] D. K. Morrison. The 14-3-3 proteins: integrators of diverse signaling cues that impact cell fate and cancer development. *Trends in cell biology*, 19(1):16–23, 2009. (cited on page 102).

[Moy08] M. A. Moyad. Brewer's/baker's yeast (saccharomyces cerevisiae) and preventive medicine: Part ii. *Urol Nurs*, 28(1):73–75, 2008. (cited on page 5).

[Mur05] R. F. Murphy. Cytomics and location proteomics: automated interpretation of subcellular patterns in fluorescence microscope images. *Cytometry Part A*, 67(1):1–3, 2005. (cited on pages 6 and 7).

[Mur08] M. Muratori, G. Forti, and E. Baldi. Comparing flow cytometry and fluorescence microscopy for analyzing human sperm dna fragmentation by tunel labeling. *Cytometry Part A*, 73(9):785–787, 2008. (cited on page 5).

[Nei00] M. Nei and S. Kumar. *Molecular evolution and phylogenetics*. Oxford University Press, 2000. (cited on page 3).

[Nic01] J. A. Nickoloff and J. E. Haber. Mating-type control of dna repair and recombination in saccharomyces cerevisiae. In *DNA Damage and Repair*, pages 107–124. Springer, 2001. (cited on page 7).

[Nie81] H. Niemann. *Pattern analysis*. Springer-Verlag Berlin, 1981. (cited on page 64).

[Ong05] S.-E. Ong and M. Mann. Mass spectrometry–based proteomics turns quantitative. *Nature chemical biology*, 1(5):252–262, 2005. (cited on page 3).

[Pap65] A. Papolulis. *Random Variables and Stochastic Processes*. McGraw Hill Book Company, 1965. (cited on page 62).

[Par97] S. Parker. *Yeast. McGraw-Hill Encyclopedia of Science and Technology*. McGraw Hill, Pennsylvania, 8th. edition, 1997. (cited on page 7).

[Paw10] J. Pawley. *Handbook of biological confocal microscopy*. Springer, 2010. (cited on page 4).

[Ped91] E. J. Pedhazur and L. P. Schmelkin. *Measurement, design, and analysis: An integrated approach*. Hillsdale, NJ: Lawerence Erlbaum Associates, 1st. edition, 1991. (cited on page 11).

[Pen13] J.-Y. Peng, Y.-J. Chen, M. D. Green, S. A. Sabatinos, S. L. Forsburg, and C.-N. Hsu. Pombex: Robust cell segmentation for fission yeast transillumination images. *PloS one*, 8(12):e81434, 2013. (cited on pages 18 and 105).

[Phi01] G. J. Phillips. Green fluorescent protein–a bright idea for the study of bacterial protein localization. *FEMS microbiology letters*, 204(1):9–18, 2001. (cited on pages 5 and 6).

[Pla98] J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schoelkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1998. (cited on page 72).

[Pos00] F. Posas, J. R. Chambers, J. A. Heyman, J. P. Hoeffler, E. de Nadal, and J. Arino. The transcriptional response of yeast to saline stress. *Journal of Biological Chemistry*, 275(23):17249–17255, 2000. (cited on pages 102 and 110).

[Pre11] A. J. Pretorius, M.-A. Bray, A. E. Carpenter, and R. A. Ruddle. Visualization of parameter space for image analysis. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2402–2411, 2011. (cited on page 51).

[Qui93] R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA, 1993. (cited on page 72).

[Ras13] G. Rastrelli, G. Corona, L. Vignozzi, E. Maseroli, A. Silverii, M. Monami, E. Mannucci, G. Forti, and M. Maggi. Serum psa as a predictor of testosterone deficiency. *The journal of sexual medicine*, 10(10):2518–2528, 2013. (cited on page 70).

[Ras16] W. Rasband. Imagej, 2016. U. S. National Institutes of Health, Bethesda, Maryland, USA, http://imagej.nih.gov/ij/ 1997-2016. (cited on pages 21, 22, 23, 26 and 32).

[Rij79] C. van Rijsbergen. *Information Retrieval*. Department of Computer Science, University of Glasgow, 2nd. edition, 1979. (cited on page 90).

[Rob91] C. J. Roberts, C. K. Raymond, C. T. Yamashiro, and T. H. Stevens. Methods for studying the yeast vacuole. In *Guide to Yeast Genetics and Molecular Biology*, volume 194 of *Methods in Enzymology*, pages 644 – 661. Academic Press, 1991. (cited on page 100).

[Rob02] M. R. Roberts, J. Salinas, and D. B. Collinge. 14-3-3 proteins and the response to abiotic and biotic stress. *Plant molecular biology*, 50(6):1031–1039, 2002. (cited on page 84).

[Row00] R. E. Rowland and E. M. Nickless. Confocal microscopy opens the door to 3-dimensional analysis of cells. *Bioscene*, 26(3):3–7, 2000. (cited on page 4).

[Sán92] I. Sánchez-Aguayo and A. L. González-Utor. Quantitative determination of changes induced by nacl in vacuoles and cellular size of lycopersicon esculentum root cells. *Plant, Cell & Environment*, 15(7):867–870, 1992. (cited on page 100).

[Sch95] M. D. Schaeberle. Raman chemical imaging: Noninvasive visualization of polymer blend architecture. *Analytical chemistry*, 67(23):4316–4321, 1995. (cited on page 10).

[Sch07] M. Schmid and T. Collins. Sigma filter plus, 2007. http://rsbweb.nih.gov/ij/plugins/sigma-filter.html. (cited on pages 21 and 32).

[Sch13] R. Schutt and C. O'Neil. *Doing data science: Straight talk from the frontline*. " O'Reilly Media, Inc.", 2013. (cited on page 13).

[Sen13] J. Senthilnath, V. Das, S. Omkar, and V. Mani. Clustering using levy flight cuckoo search. In *Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012)*, pages 65–75. Springer, 2013. (cited on page 70).

[Sha97] S. L. Shaw, E. Yeh, K. Bloom, and E. Salmon. Imaging green fluorescent protein fusion proteins in saccharomyces cerevisiae. *Current Biology*, 7(9):701–704, 1997. (cited on pages 5 and 6).

[Sha01] L. G. Shapiro and G. C. Stockman. *Computer Vision*. Prentice-Hall, Upper Saddle River, 2001. (cited on page 12).

[Shi79] O. Shimomura. Structure of the chromophore of aequorea green fluorescent protein. *Febs Letters*, 104(2):220–222, 1979. (cited on page 5).

[Shr13] G. Shruthi and R. K. A N. Image reconstruction using discrete wavelet transform. *IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)*, 2(4):14–20, 2013. (cited on page 64).

[Sim04] D.-G. Sim, H.-K. Kim, and R.-H. Park. Invariant texture retrieval using modified zernike moments. *Image and Vision Computing*, 22(4):331–342, 2004. (cited on page 109).

[Sol11] C. Solomon and T. Breckon. *Fundamentals of Digital Image Processing: A practical approach with examples in Matlab*. John Wiley & Sons, 2011. (cited on page 10).

[Stă13] A. Stănilă. Brewer's yeast: an alternative for heavy metal biosorption from waste waters. *ProEnvironment/ProMediu*, 6(15), 2013. (cited on page 7).

[Štr09] V. Štruc and N. Pavešic. A comparison of feature normalization techniques for pca-based palmprint recognition. In *Proceedings of International Conference MATHMOD*, pages 2450–2453, 2009. (cited on page 68).

[Str11] J. Strickland. *Simulation Conceptual Modeling*. Lulu.com, 2011. (cited on page 105).

[Sun03] C. Sun and S. Pallottino. Circular shortest path in images. *Pattern Recognition*, 36(3):709–719, 2003. (cited on page 41).

[The10] F. J. Theis and A. Meyer-Bse. *Biomedical Signal Analysis: Methods and Applications*. MIT Press, 2010. (cited on page 53).

[Tho96] C. R. Thomas and G. C. Paul. Applications of image analysis in cell technology. *Current Opinion in Biotechnology*, 7(1):35–45, February 1996. (cited on page 10).

[Tho06] R. Thomas. Flow cytometry, October 6 2006. US Patent App. 11/544,239. (cited on page 4).

[Tle12] M. S. Tleis and F. J. Verbeek. Yeast cell features extraction plugin. In *ImageJ Conference 2012*, Mondorf, Luxemburg, October 2012. (cited on page 30).

[Tle13] M. S. Tleis, G. Anemaet, P. van Heusden, and F. Verbeek. Image analysis platform for yeast biologists. In *The 2nd International Conference on Advances in Biomedical Engineering 2013 (ICABME'13)*, pages 57–60, Tripoli, Lebanon, September 2013. (cited on page 11).

[Tle14] M. S. Tleis and F. J. Verbeek. Extracting contours of oval-shaped objects by hough transform and minimal path algorithms. In *Sixth International Conference on Digital Image Processing*, page 915903. International Society for Optics and Photonics, 2014. (cited on pages 44, 45, 61 and 106).

[Tsa12] P. Tsai, M. Torabinejad, D. Rice, and B. Azevedo. Accuracy of cone-beam computed tomography and periapical radiography in detecting small periapical lesions. *Journal of endodontics*, 38(7):965–970, 2012. (cited on page 71).

[van07] P. van der Putten, L. Bertens, J. Liu, F. Hagen, T. Boekhout, and F. J. Verbeek. Classification of yeast cells from image features to evaluate pathogen conditions. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 6506 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, January 2007. (cited on pages 11 and 63).

[Ver02] F. J. Verbeek and P. J. Boon. High-resolution 3d reconstruction from serial sections: microscope instrumentation, software design, and its implementations. In *International Symposium on Biomedical Optics*, pages 65–76. International Society for Optics and Photonics, 2002. (cited on pages 52 and 56).

[Ver04] F. J. Verbeek, D. Rodrigues, H. Spaink, and S. A. Data submission of 3d image sets to a bio-molecular database using active shape models and a 3d reference model for projection. In *Proceedings of Internet Imaging V*, San Jose, CA., December 2004. SPIE. (cited on page 32).

[Ver14] N. K. Verma, L. Rao, and S. K. Sharma. Motor imagery eeg signal classification on dwt and crosscorrelated signal features. In *9th International Conference on Industrial and Information Systems (ICIIS)*, pages 1–6. IEEE, 2014. (cited on page 68).

[Vin88] L. Vincent. Minimal path algorithms for the robust detection of linear features in gray images. *Proceedings of the International Symposium on Mathematical Morphology*, 1988. (cited on page 40).

[Vos82] A. M. Vossepoel and A. W. M. Smeulders. Vector code probability and metrication error in the representation of straight lines of finite length. *Computer Graphics and Image Processing*, 20:347–364, 1982. (cited on page 26).

[Wal04] L. Walker, M. Aldhous, H. Drummond, B. Smith, E. Nimmo, I. Arnott, and J. Satsangi. Anti-saccharomyces cerevisiae antibodies (asca) in crohn's disease are associated with disease severity but not nod2/card15 mutations. *Clinical & Experimental Immunology*, 135(3):490–496, 2004. (cited on page 5).

[Wel07] M. Wellen. *Spatio-temporal gene expression analysis from 3D in situ hybridization images*. PhD thesis, Section imaging and bioinformatics, Liacs, Leiden University, November 2007. (cited on page 3).

[Wes76] J. S. Weszka, C. R. Dyer, and A. Rosenfield. A comparative study of texture measures for terrain classification. *IEEE Transactions on Systems Maintenance Cybernet*, pages 269–285, 1976. (cited on page 64).

[wik15] Wikipedia, Sep 2015. http://en.wikipedia.org/wiki/Feature (computer vision). (cited on page 12).

[Win99] E. A. Winzeler, D. D. Shoemaker, A. Astromoff, H. Liang, K. Anderson, B. Andre, R. Bangham, R. Benito, J. D. Boeke, H. Bussey, et al. Functional characterization of the s. cerevisiae genome by gene deletion and parallel analysis. *Science*, 285(5429):901–906, 1999. (cited on page 7).

[Yan97] F. Yang, L. G. Moss, and G. N. Phillips Jr. The molecular structure of green fluorescent protein. *Department of biochemistry and cell biology, Rice University*, 1997. (cited on page 5).

[Yan12] K. Yan and F. Verbeek. Segmentation for high-throughput image analysis: Watershed masked clustering. In T. Margaria and B. Steffen, editors, *Leveraging Applications of Formal Methods, Verification and Validation. Applications and Case Studies*, volume 7610 of *Lecture Notes in Computer Science*, pages 25–41. Springer Berlin Heidelberg, 2012. (cited on page 50).

[Yan15] W. Yang, J. Fu, M. Yu, D. Wang, Y. Rong, P. Yao, A. K. Nuessler, H. Yan, and L. Liu. Effects of three kinds of curcuminoids on anti-oxidative system and membrane deformation of human peripheral blood erythrocytes in high glucose levels. *Cellular Physiology and Biochemistry*, 35(2):789–802, 2015. (cited on page 4).

[Yus05] R. Yuste. Fluorescence microscopy today. *Nature methods*, 2(12):902–904, 2005. (cited on page 9).

[Zac77] G. Zack, W. E. Rogers, and S. A. Latt. Automatic measurement of sister chromatid exchange frequency. *Journal of Histochemistry and Cytochemistry*, 25 (7):741–753, 1977. (cited on page 22).

[Zah12] J. Zahradka, G. P. H. van Heusden, and H. Sychrova. Yeast 14-3-3 proteins participate in the regulation of cell cation homeostatis via interaction with nha1 alkali-metal-cation/proton antiporter. *Biochimica et Biophysica Acta*, 1820:849–858, 2012. (cited on pages 8, 102 and 110).

[Zho10] Y. Zhou, X. Wu, T. Wang, T. Ming, P. Wang, L. Zhou, and J. Chen. A comparison study of detecting gold nanorods in living cells with confocal reflectance microscopy and two-photon fluorescence microscopy. *Journal of microscopy*, 237(2):200–207, 2010. (cited on page 4).

[Zon86] B. J. Zonneveld. Cheap and simple yeast media. *Journal of microbiological methods*, 4(5):287–291, 1986. (cited on page 85).

Summary

Image analysis of objects in the microscope scale requires accuracy so that measurements can be used to differentiate between groups of objects that are being studied. This thesis deals with measurements in yeast biology that are obtained through microscope images. We study the algorithms and workflow of image analysis of yeast cells in order to understand and improve the measurement accuracy. The *Saccharomyces cerevisiae* cell is widely used as a model organism in the life sciences. It is essential to study the gene and protein behaviour within these cells, and consequently making it possible to find treatment and solutions for genetic and hereditary diseases. This is possible since many processes that occurs at the molecular level in this organism are similar to those in human cells.

In the research group Imaging and Bioinformatics, we have developed a framework for analysis of yeast cells. This framework is intended to serve as a support for research in yeast biology. The framework is integrated in one application and presented via a *GUI*. The application integrates modules and algorithms including segmentation, measurement, analysis and visualization.

In **Chapter 1**, we present our research objective: i.e. the problem on how pattern recognition systems can support objective analysis and phenotype characterization of single-cell in image-based gene expression experiments. In addition, we offer a basic background and definitions necessary to follow up in this thesis. Specifically, we introduce the necessary background about cytomic studies and pattern recognition methods followed within our research.

In **Chapter 2**, we present a complete framework for image based experimental read-out in yeast. This framework demonstrates how an automated platform based on a complete image analysis pipeline can assist biologists in their experiments. Moreover, this chapter discusses the individual modules that are integrated in the complete framework: i.e. including the segmentation, measurement, data analysis as well as the GUI that combines these modules together.

In **Chapter 3**, we discuss our novel approach to segmentation based on Hough transform and minimal path algorithms. We show how these can improve the segmentation of ovoid objects, i.e. yeast cells. We start by defining Hough transform and minimal path algorithms. Subsequently we present our general approach to detect ovoid objects in microscope images by detecting circular arcs using a variety of the Hough transform. In addition, we discuss the application of minimal path algorithms to extract the exact contour of detected objects from a polar representation of the image surrounding the object. Furthermore, this chapter presents an additional novel algorithm to expand the extracted contours of ovoid objects. Such expansion is

sometimes necessary settings due to the inherent fuzzy nature of edges and delicate microscope settings. This chapter explains how the polar representation of images is used to expand the initially detected contours by applying circular shortest paths. In addition, it explains the three introduced parameters to control the expansion process. These parameters are *resistance, limit* and *convergence*. Finally, results and comparison with other methods are evaluated using a dataset of *S. cerevisiae* cells.

In **Chapter 4**, we specifically address machine learning where we introduce features to be used in a machine learning approach to automatically identify cell groups cultivated in two different media. We use the same approach to classify cell objects from atrefacts. First we discuss the feature extraction techniques including first-order histogram features, texture measurement, moment invariants, co-occurrence matrix based features and multi-scale wavelet-based texture measurement. Subsequently, various classification methods are evaluated to build a model imported into the yeast analysis platform. This model is trained for the automatic discrimination of cell groups. This discrimination can be used to show that there are different gene expression patterns between cells cultivated under different stress levels. Moreover, the same classification methods are evaluated to build another model for the identification of cell objects. This model is used to discriminate the segmented objects in images into intact cell objects or artefacts such as debris and dead cells.

In **Chapter 5**, the designed image analysis platform is used in a case study to determine the effect of sodium chloride on 14-3-3 genes including *Bmh1* and *Bmh2* in addition to the *Nha1* antiporter. The study also includes a mutant of *BMH1 (Δbmh1)* to study the expression of *Nha1* under different osmotic stress levels. The result obtained from using the yeast analysis software is also validated with that obtained from flow cytometry.

In **Chapter 6**, we present conclusions and lessons learned from this research. Subsequently, we give scope for further research and applications.

Samenvatting

Beeldanalyse van objecten op microscopische schaal vereist nauwkeurigheid opdat de metingen uit de beeldanalyse kunnen worden gebruikt om te verschillen te vinden in groepen van objecten die worden bestudeerd. Dit proefschrift gaat over metingen verkregen uit microscope beelden gedaan zijn ten behoeve van onderzoek in de gistbiologie. Wij hebben algoritmes en workflow van beeldanalyse van gist-cellen bestudeerd teneinde de meetnauwkeurigheid te begrijpen en verbeteren. De Saccharomyces cerevisae (bakkersgist) cel is een algemeen gebruikt model systeem in de levenswetenschappen. Het is essentieel het gedrag van genen en eiwitten in deze cellen te bestuderen zodat het daarmee kan bijdragen aan het vinden van behandelingen en oplossingen voor genetische en erfelijke ziekten. Dit is mogelijk omdat veel van de processen die zich op het moleculiare niveau afspelen in de gistcel vergelijkbaar zijn met de processen in de menselijke cel.

In de onderzoeksgroep Imaging & BioInformatica van het LIACS hebben we een platform ontwikkeld voor het doen van beeldanalyse van gistcellen. Dit platform dient ter ondersteuning van het gistbiologie onderzoek; het is ingebed in een softwaretoepassing en wordt bediend door middel van een Grafische Gebruikers Omgeving. De softwaretoepassing integreert modules and algoritmes voor segmentatie, metingen, data-analyse en visualisatie.

In hoofdstuk 1, wordt de motivatie voor het onderzoek uiteengezet, meer specifiek: hoe kunnen systemem voor patroon herkenning worden ingezet voor het onderzoek naar genexpressie in cellen gebaseerd op beelden vekregen uit experimenten. Daarnaast presenteren we de basis achtergronden en definities die nodig zijn voor de vervolghoofdstukken in dit proefschrift. Daarbij wordt speciaal aandacht gegeven aan cytomics en methodieken uit patroonherkennen zoals dezen zijn gebruikt in het beschreven onderzoek in dit proefschrift.

In hoofdtstuk 2 wordt een complete omgeving voor beeld-gebaseerd experimenteel gist onderzoek gepresenteerd. Deze omgeving demonstreert hoe een complete reeks van bewerkingen (ook wel pipe-line genoemd) biologen kan helpen in het doen en verwerken van experimenten. In dit hoofdstuk worden ook de afzonderlijke componenten geïntegreerd in de reeks van bewerkingen behandeld: te weten segmentatie, metingen, data analyse en het Grafische Gebruikers Omgeving waarbinnen al deze componenten als een applicatie gepresenteerd worden.

In hoofdstuk 3 bespreken we onze nieuwe aanpak voor segmentatie waarbij gebruik gemaakt wordt van de Hough Transformatie en minimale pad-lengte algoritmes. We laten zien hoe deze algoritmes de segmentatie van ei-vormige (ovoide) objecten, zoals gist cellen, kan verbeteren. We beginnen met een uitleg van de basis van de Hough Transformatie en minimale padlengte algoritmen. Daarop presenteren we onze generieke aanpak om eivormige objecten te vinden

in microscoop beelden door cirkelboogsegmenten te zoeken gebruikmakend van onze eigen variant van het Hough Transformatie algoritme. Daarnaast gaan we in op het toepassen van minimale padlengte algortimen om na een eerste schatting de exacte contour van de objecten te vast te stellen. Hierbij wordt gebruik gemaakt van een polaire representatie van het deel van het beeld waar het object is gevonden. Dit hoofdstuk beschrijft een additioneel algoritme voor het verruimen van de gevonden contouren van de eivormige objecten. Een dergelijke verruiming is soms nodig als gevolg van onduidelijkheid van de randen en ook door bepaalde instellingen/keuzes in de microscopie. Eerst wordt uitgelegd hoe de polaire representatie van de beelden wordt gebruikt om het initiëele resultaat te verbeteren door het toepassen van ons circulaire kortste pad algoritme. Daarna wordt uitgelegd hoe, door gebruik te maken van drie condities, dit proces precies gestuurd kan worden. Deze condities zijn: weerstand, limiet en convergentie. Het hoofdstuk sluit af met een demonstratie van resultaten en vergelijkt deze resultaten met andere methoden waarbij gebruik gemaakt wordt van een set van "test-beelden" van S. cerevisiae cellen.

In hoofdstuk 4 gaan we specifiek in op de "machine learning" aspecten en we introduceren de kenmerken die we gebruiken om, in een "machine learning" aanpak, automatisch groepen van cellen te kunnen herkennen die gecultiveerd zijn in twee verschillende media. We gebruiken eenzelfde aanpak om cellen te classificeren tegenover artefacten. We beginnen met het uitleggen van technieken voor kermerk-extractie, waaronder eerste-orde histogram kenmerken, textuur metingen, invariante momenten, co-occurent matrix kenmerken en multi-scale wavelet textuur kenmerken. Daaropvolgend worden verschillende classificatie methoden geëvalueerd teneinde een model te bouwen dat kan worden gebruikt in het platform dat we in Hoofdstuk twee hebben geïntroduceerd. Dit model is getraind om de automatische de verschillende celgroepen te kunnen herkennen. Deze automatische herkenning kan worden ingezet om te laten zien dat er veschillende patronen van genexpressie zijn wanneer cellen in verschillende niveaus van stress worden gecultiveerd. Dezelfde classificatie methoden worden bovendien geëvalueerd om cellen te kunnen identificeren tegenover andere objecten. Op deze manier worden rommel en resten van dode cellen gescheiden van levende cellen.

In hoofdstuk 5 wordt het beeldanalyse platform gebruikt in een toepassing waarbij het effect van keukenzout (NaCl) wordt gedemonstreerd op de expressie van de 14-3-3 genen, te weten Bmh1 en Bmh2, in samenhang met Nha1. In deze studie wordt daarnaast ook nog gebruik gemaakt van een mutant-stam van het BMH1 gene waardoor de expressie van Nha1 onder verschillende niveaus van osmotische stress kan worden bestudeerd. De resultaten die uit de beeldanalyse zijn verkregen zijn gevalideerd met een complementaire study waarbij gebruik gemaakt is van flow-cytometrie.

In hoofdstuk 6 sluiten we af met de conclusies en de lessen geleerd in dit onderzoek. Tenslotte geven we een blik op de toekomst met suggesties voor verder onderzoek.

List of Publications

[1] Mohamed Tleis, Paul van Heusden and Fons J. Verbeek.*YeastAnalysis*: An image analysis platform to quantify fluorescent reporter proteins in *Saccharomyces cerevisiae* cells. (Submitted).

[2] Mohamed S. Tleis and Fons J. Verbeek. "Machine Learning approach to discriminate *Saccharomyces cerevisiae* yeast cells using sophisticated image features." Journal of Integrative Bioinformatics, 12(3):276, 2015.

[3] Mohamed Tleis and Fons J. Verbeek. "Contour Expansion preceded by the Application of Hough transform and Minimal Path Algorithm". In Fifth International Conference on Image Processing Theory, Tools and Applications (IPTA). pp. 149-154. IEEE, 2015.

[4] Mohamed S. Tleis and Fons J. Verbeek. "Machine Learning approach to segment *Saccharomyces cerevisiae* yeast cells." In third International Conference on Advances in Biomedical Engineering (ICABME). pp. 278-281. IEEE, 2015.

[5] Mohamed Tleis and Fons J. Verbeek. "Extracting contours of oval-shaped objects by Hough transform and minimal path algorithms." In Sixth International Conference on Digital Image Processing, pp. 915903-5. International Society for Optics and Photonics, 2014. *(***Excellent Paper Award)**.

[6] Mohamed Tleis, Ginny Anemaet, Paul van Heusden, and Fons J. Verbeek. Image analysis platform for yeast biologists. In The 2nd International Conference on Advances in Biomedical Engineering 2013 (ICABME'13), pages 57–60, Tripoli, Lebanon, September 2013.

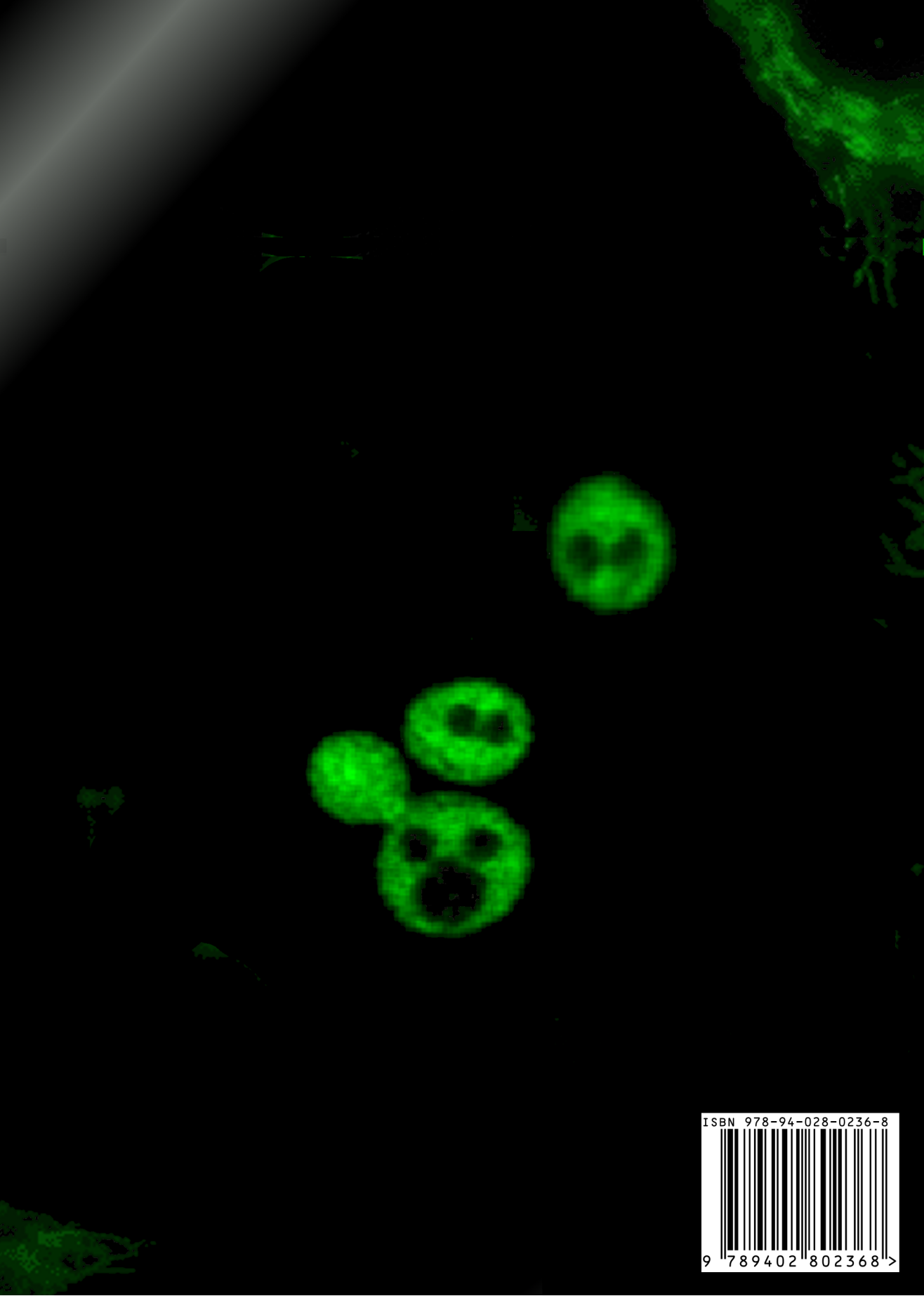
[7] Mohamed Tleis and Fons J. Verbeek. Yeast-Cell features extraction plugin. In ImageJ Conference 2012, Mondorf, Luxembourg, October 2012.

About the Author

Mohamed Tleis was born on December 05 1982, in Brital, a small village in the Bekaa valley in Lebanon. He studied Computer Science at the American University of Science and Technology in Beirut, and completed his Master's degree in 2007 with a thesis on Arabic Speech Recognition.

In April 2007, he started working for the IT department of the Lebanese University Central Administration as a Server and Network Administrator. During this period he followed many training programs at a Microsoft certified center. In addition, he followed many training programs in Library Cataloging systems. He also trained Librarians from all over the country on using the cataloging system.

In June 2011 he started his PhD at Leiden University in the imaging and Bioinformatics group of the Leiden Institute of Advanced Computer Science (LIACS). His PhD program was partially supported by the *EU* in an *Erasmus Mundus* Program for the Middle East region, i.e. *JoSyLeEN*, and later by *Raymond-Sackler* organisation and the *Landelijke Stitching voor Blinden en Slechtzienden* organization (*LSBS*). This research was performed under supervision of Fons Verbeek and focused on automated and accurate methods for image analysis of yeast cells.



ISBN 978-94-028-0236-8