Karlsruher Institut für Technologie

Institute for Applied Computer Sciences (IAI)
Prof. Dr. Veit Hagenmeyer

Development of Robust and Efficient Algorithms for Image Processing and Analysis on Multidimensional Image Data using Feedback Concepts with Challenging Applications

Doctoral Thesis
by
M.Sc. Arif ul Maula Khan
(Matr.-Nr: 1507951)

Head:	Prof. Dr. Veit Hagenmeyer
Supervisors:	apl. Prof. Dr.-Ing. Ralf Mikut
Co-Supervisors:	Prof. Dr. Jivka Ovtcharova
	PD Dr. Carsten Weiss
	PD Dr.-Ing. Markus Reischl

Submission Date:	16.02.2017
Oral Examination Date:	16.05.2017

Institute of Applied Computer Sciences (IAI)
Prof. Dr. Veit Hagenmeyer

Development of Robust and Efficient Algorithms for Image Processing and Analysis on Multidimensional Image Data using Feedback Concepts with Challenging Applications

Zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
Der Fakultät für Maschinenbau
Karlsruher Institut für Technologie (KIT)

genehmigte
Dissertation
von

Arif ul Maula Khan

Tag der mündlichen Prüfung: 16.05.2017
Hauptreferent: apl. Prof. Dr. -Ing. Ralf Mikut
Korreferent: Prof. Dr. Jivka Ovtcharova
PD Dr. Carsten Weiss
PD Dr.-Ing. Markus Reischl

Declaration

I hereby guarantee that, the following work has been completed independently only using the helping material referred to in this dissertation. All forms, contents and references have been made clear in this thesis according to the requirements of the Karslruhe Institute of Technology (KIT).

I hereby also declare that, I have no failed or pending Ph.D. examination procedures outside the faculty of Mechanical Engineering of the Karlsruhe Institute of Technology (KIT).

Karlsruhe, 22nd May 2017

Khan, Arif ul Maula

Gratitude

Thanking my Professor Mr. Georg Bretthauer for overseeing my Ph.D. thesis.

Special thanks to my supervisors: apl. Prof. Dr. Ralf Mikut and Dr. Markus Reischl for giving me adequate support and suitable supervision in the entire doctoral work, and to my co-supervisors: PD Dr. Carsten Weiss and Prof. Dr. Ruediger Rudolf for being my co-operation partners and advisers of my Thesis Advisory Committee (TAC).

Also expressing my gratitude for the BioInterfaces International Graduate School (BIF-IGS) for being the umbrella for the cooperation projects, giving me suitable counseling and providing me with ample opportunities to appropriate technical skills.

Last but not the least, bundles of thanks to German Academic Exchange Service (DAAD) and Institute for Applied Computer Sciences (IAI) for providing me the funding to undertake this research work at KIT.

Contents

1 Motivation

1.1 Overview

Currently, the focus on using image processing techniques and methods as a powerful analysis has become quite alluring. Tools are devised in order to develop adequate comprehension of technical and natural processes and to solve challenging tasks therewith. While manual image analysis in general delivers good results, it is inadequate in understanding parametric effects on the outcome and quite hectic when dealing with large datasets containing a diversity of information. Automatic image analysis in this context is both imperative and challenging for an efficient and meaningful image analysis. Therefore, the appeal of fulfilling desired goals in image analysis incorporating user-interactive software applications with improved results is all the more rising. The aim of this doctoral thesis is to introduce a new methodology of automatic feedback based image analysis routines using feedback concepts. Consequently, developing user-interactive tools to facilitate time-efficient and improved image analysis is the aim in this research work. The incorporation of statistical data mining analysis into feedback based image processing routines is also a subject of investigation. Image pre-processing (shading correction etc.) and segmentation are integral procedures for eventual image objects classification and other similar image analysis goals. In comparison to automatic feed-forward methods, feedback based automatic image normalization, object segmentation, feature calculation and consequent object type classification have the potential to adapt involved parameters in an iterative and automatic fashion. Therefore, an optimal parameter set is obtained resulting in an optimal object classification based on given *a priori* knowledge. The automatic feed-forward methods with the manual feedback is the state-of-the-art. Each image processing step could be handled iteratively through a feedback to produce optimal parameters for that step. The evaluation criteria for the rectitude and effectiveness of each procedure as well as *a priori* knowledge are formulated using fuzzy logic to induce a range of confidence of included object features and variables.

Image processing has always been the driving technique in order to manipulate and interpret the acquired images in such a fashion that resulting information could be used in a very efficient manner. Image processing is a means for automatically extracting the image content, often used in science (e.g. biological readouts (mouse [157], fish [82], insect [100]) surveying and mapping [19], particle accelerator [12] and man-machine interaction [53, 99]) or industry (nutrition industry [14, 31], quality supervision [70] or pick-and-place applications [106, 150], an overview of image processing in industry is given in [88].).

In computer vision applications, image processing routines need to be developed for image datasets containing a set of similar images. This happens in many real-time acquisition systems (surveillance camera etc.), lab equipment (high-throughput microscopic imaging etc.) or offline analysis of big databases containing similar images (objects in satellite images, human blood cells analysis in laboratory etc.).

Image analysis is done usually by employing different image processing steps in what constitutes a pipeline a.k.a image processing/analysis pipeline. Each step has a specific function depending upon the goals of image analysis. Image data riddled with artifacts may require additional processing steps. Therefore, first step is the structural design of an image processing pipeline. For each individual image processing step, algorithms are performed with certain amount of parameters. These parameters may be chosen manually or automatically. Consequential, after the design of image processing pipeline, parameters have to be selected that produce desired outcome with respect the set targets. The main aim in this research work is to propose exemplary image processing/analysis pipelines and select the optimal parameters for the employed algorithms automatically. This is proposed to be done on challenging datasets in order to establish the usefulness of automatic parameter selection through feedback mechanism.

1.2 Related work

A typical workflow of an image processing pipeline consists of different steps and is shown in Fig. 1.1. Different step involved in an image processing pipeline are:

- Image pre-processing (image normalization, shading correction etc.)
- Image segmentation,
- Feature extraction,
- Feature selection and
- Image object classification.

The most important of the above mentioned steps are handled in detail in the following sections. As an input raw image, a grayscale image is used. In the first step, image filtering is done in order to remove background noise and illumination such that it is more convenient to discern objects from the background scene. In the next step, useful objects in the image are segmented using various techniques discussed in the following section. Using the segmented objects, features are calculated in order to perform some analysis (e.g. classifying object types etc.), based on these features. Finally, a classifier is used to find the best features that would enable it to differentiate between different object types.

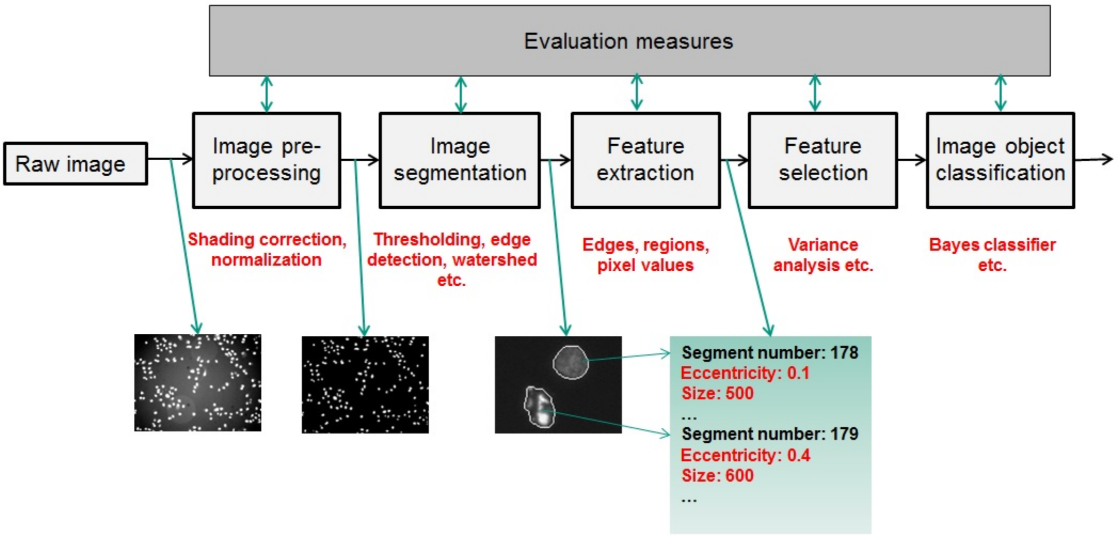


Figure 1.1: A basic image processing and analysis pipeline (adapted from [93])

1.2.1 Image pre-processing

Digital images are subject to a diversity of unwanted distortions that are inevitably linked to the acquisition conditions, pre-processing and storage of images. In the realm of image processing and image analysis, normalization alludes to retrieval of original inherent information (i.e. intensity values) in an image. This is achieved by manipulation of pixel values (i.e. corresponding image intensity values) in order to obtain a desired range of pixel values satisfying certain desired image processing goals. In case of a single image, objectives of a normalization process, for one, can be rescaling of pixel values to a desired range while disregarding outliers and noisy background pixels.

The optimal parameters of an image segmentation procedure are often affected by side effects (e.g. blurriness, noise, inconsistent background illumination etc.) [114]. These side effects cause distortions in image intensities, which in turn causes loss of targeted information present in an image. For instance, microscopic images are often corrupted by erroneous intensity variation on account of inherent shortcomings of the image formation process. This phenomenon of intensity variation in literature is described using terms such as shading, intensity inhomogeneity, intensity non uniformity, bias field and gain field [147, 83]. While manual image analysis maybe less prone to invalid interpretation of images in presence of such spurious intensity variation effects, automatic image analysis is likely to get riddled with such effects [144]. Shading is generally assumed to be a smooth spatially varying function for 2D images that corrupts intensities of the imaged objects. In absence of such a shading, image intensities of the imaged objects would be the same irrespective of the location of imaged objects in an acquired image.

Currently, there are a myriad of shading correction methods on the horizon. Shading can be roughly categorized as either object-independent or object-dependent [83]. The former is originated from certain shortcomings in the image acquisition process and is independent of the imaged object while the latter is caused by imperfectly prepared objects to the acquisition device such as staining inhomogeneity etc. [147]. Shading correction methods are basically categorized as *prospective* and *retrospective*, depending upon the degree of access to the available information related to a given image. *Prospective* methods are related to calibration and improvement of image acquisition process, whereas *retrospective* methods are based exclusively on the information present in the acquired image (an *a priori* is also available sometimes).

Various methods have been proposed in the field of *retrospective* shading correction methods using a linear image formation model, consisting of additive and multiplicative shading components in an acquired image [83, 115, 144, 147].

Retrospective methods can be simplified by using only one shading component i.e. additive. *Retrospective* methods are further divided into several approaches such as filtering, surface fitting, segmentation, histograms and others [147]. In *retrospective* method, it is assumed that imaged objects are limited in size and smaller than the scale of background variations, in addition to background being either darker or lighter than the objects everywhere in the image. Another retrospective method CIDRE (corrected intensity distributions using regularized energy minimization) was proposed in [130] that uses both multiplicative and additive noise models in order to achieve the quality of *prospective* methods.

In case of filtering, a reliable removal of a shading effect from true image data is only possible in the case where the spectra of the shading component and true image data are not overlapping. If this condition does not hold, then the true data is liable to be corrupted by the shading correction procedure. These techniques are more suited for small scale structured images [83]. On the other hand, surface fitting methods are parametric methods approximating a smoothly varying image background by parametric surfaces such as polynomials etc. Background parameters can be retrospectively estimated either by a manual or an automatic selection of control points using a least-squares fitting. The optimal parameters can be ascertained by iterative searching in parametric space and optimizing certain criterion such as squared error or other robust distance measure between the parametric background and the acquired image [83].

Besides shading correction, several other image pre-processing techniques do exist [54] such as distribution linearization, spatial digital filtering methods using Fast Fourier Transform (FFT) or recursive partial differential equations, image subtraction, contrast enhancement etc. These techniques are designed to improve the feature extraction from a given image and to eliminate irrelevant intensity values disturbing the image [54]. Image restoration normally alludes to the problem of estimating the ideal image from its blurred and noisy rendition [124]. The ultimate goal is to restore the image to a very usable form devoid of artifacts and noise inherited by various image acquisition causes. Different approaches for image restoration include criterion-based (e.g. minimum-norm least square), constraint-optimal (e.g. constraint least square) and constraint-based (e.g. set theoretic). Additionally, there are other specialized techniques too such as restoration of images degraded by space-variant blurs [124]. Moreover, there are various methods used for image denoising (i.e. noise removal) for digital images and a short survey in terms of comparison is given in [15].

Image deconvolution is the term used when reversing the effects of optical distortions present in a image due to an imaging instrument is desired. The usual assumption is a perfect optical path through an imaging instrument convolved with a point spread function (PSF), i.e., a mathematical description of the distorted pathway of a theoretical point source of light when passed through the instrument [22]. Practically, the PSF can only be approximated by theoretical calculations [95] or based on some experimental estimation. Trying different PSFs systematically in order to estimate it when PSF is not directly given is called blind deconvolution [22]. The most common examples are Richardson-Lucy deconvolution (iterative algorithm) and the Wiener deconvolution (non-iterative algorithm).

1.2.2 Image segmentation

Image segmentation is an integral part of image analysis that divides an input image into regions [43]. According to objective evaluation of the segmentation outcome, an image segmentation procedure can be called as *supervised* or *unsupervised*, based on the presence of reference *a priori* knowledge [156]. *A priori* knowledge can be explicit (i.e. based on known objects) or implicit (i.e. using certain set of rules or examples).

Manual *supervised* image segmentation performed by humans delivers, in general, good results as a human brain bridges information known about the image (e.g. noise) with the information of segments (e.g. segment size and intensity etc.) to obtain a plausible outcome with respect to the posed problem. However, human image segmentation is downright time-inefficient when dealing with huge image datasets containing a variety of information.

Automatic *supervised* image segmentation techniques with manually tuned parameters (e.g. threshold values etc.) suffer from restrictions in parameters: Basically, the manual tuning seeks to optimize features like segment size, roundness etc. but the parameters only affect parameters like brightness threshold, filter size etc. Therefore, the tuning does not directly affect the features to be optimized.

Image segmentation can be classified based on approaches such as model-based (e.g. based on spectral distribution etc.) or image-based (e.g. edge-based etc.) [28]. Core problems originating from an image segmentation procedure are the objective and quantitative evaluation and the validity of the obtained results [159]. As opposed to visual perception (i.e. *subjective* evaluation of image segmentation process), a *supervised* image segmentation technique generates results that are based on an *a priori* knowledge whereas an *unsupervised* technique yields an objective evaluation without using any reference [156].

Different techniques used for image segmentation are thresholding [103], histogram-based [128], edge detection [17], region growing [126], clustering [105], compression-based, split-and-merge [133], watershed [146], graph-partitioning [49], active contours [69], partial differential equation based (i.e fast marching methods [123], active contours [19]), model-based [89], segmentation using sparse pixel classification [25] and many more [101]. The evaluation methods are given in [62, 145]. There has been a lot of work done too in terms of image segmentation on biological structures such as cells in [161, 160, 153, 109, 61, 55, 84, 155, 154, 116, 92, 91, 85, 1, 4, 24, 64]. However, the main focus of this work will remain of robust automatic image segmentation methods based on feedback concepts.

An image segmentation procedure is also affected by side effects of noise, inconsistent background illumination etc. This effect can be elucidated by a typical example from a biological image showing numerous cells in different states shown in Fig. 1.2. The image suffers from an inconsistent background illumination since the background in corners is darker than in the middle of the image. Computer routines additionally suffer not only from restrictions in the parameters but also from the combinatorial problem due to an increased number of parameters [7]. Thus, the optimal parameter set may not be found manually, while the tuning is very time-intensive and subject to repetitions on the arrival of a new dataset.

1.2.3 Feature extraction and image object classification

In machine learning, image object classification is contained in a broader domain of pattern recognition (i.e. label assignment to a given input value). In image classification, this assignment problem is exclusively limited to a given set of classes. Precisely speaking, it is the problem of categorizing a new observation to known set of classes based on a training dataset containing observations of known class membership. A classifier refers to an algorithm or a concrete implementation of classification mapping input data to a known category.

A detailed systematic categorization of image object classification methods is given in [86]. In machine learning, two distinct categories in terms of the availability of *a priori* for classification exist [67]:

- ***supervised* :** establishing a rule for classifying a new observation into one of the existing classes given a set of observations
    1. parametric

        - Linear Discriminant Analysis (LDA)
        - Quadratic Discriminant Analysis (QDA)
        - maximum entropy classifier (a.k.a. logistic regression)
    2. nonparametric
        - decision trees, decision lists
        - kernel estimation and k-nearest-neighbor algorithms
        - maximum entropy classifier (a.k.a. logistic regression)
        - Bayes classifier and Naive Bayes classifier
        - neural networks
        - Support Vector Machines (SVM)

- ***unsupervised/clustering*** **:** establishing the existence of classes or clusters in the data given a set of observations
    - categorical mixture models
    - hierarchical clustering
    - k-means clustering
    - correlation clustering
    - kernel principal component analysis (kernel PCA)

Methods such as deep learning can be both *supervised* and *unsupervised*. Similarly, there are many other existing methods and categories. In *supervised* methods, there are certain advantages as in error detection by user and corresponding solution finding. Since expert knowledge is required to train such classifiers, an accurate result is expected. However, such methods are not useful for big datasets since a lot of input effort and time to train such classifiers is required. Moreover, it is more laborious to use them for huge variety of different data classes. Consequently, the validation process also becomes very hectic.

On the other hand, *unsupervised* methods save the time required to specify the data classes and types looked after in the whole dataset but may require a larger effort and expert skills to understand the classifier results on a particular dataset. An ad-hoc classifier in this regard is hard to obtain.

Correctly parametrized pipelines deliver good results on the image dataset they were designed for.

1.2.4 Goals and performance evaluation

Various tasks and algorithms are normally incorporated in the term ’Image processing’ such as image normalization, image segmentation, image enhancement etc. Normally, the goal of an image processing algorithm is to achieve qualitative and quantitative success in one/all of the steps described above. For instance, the goal of image segmentation algorithm when applied to an image should be to deliver targeted segments (i.e. objects containing useful information). The assumption of successfully extracting an object or region of interest from a particular background in image processing applications has to be backed up by reasonably successful quantitative evaluations. The rate of success can be assessed by different performance metrics. In order to decipher meaningfulness out of achieved results, one should define the success rate. A reliable segmentation outcome is hard to obtain without having any prior knowledge of the object or region of interest that has to be extracted from the scene under consideration. In the presence of *a priori* knowledge (i.e. knowledge about segments known or provided before hand in case of image segmentation), the success rate is the minimization of error between segmentation outcome and *a priori* knowledge reference. In the presence of such a reference knowledge, the algorithm is called to *supervised*. In case of *unsupervised* image processing, evaluation of a segmented image is based on its goodness to match a broad set of characteristics of segmented images as desired by humans. Both methods described above are quantitative and objective.

Image processing and analysis could both be done in different ways. Image analysis that is done manually by human observations without using statistical analysis or algorithmic intelligence produces in general good results. Unfortunately, human is neither sensitive to fine differences in color intensity nor effective to quantify and average brightness levels among different objects to be found in an image e.g. manual quantification of cell intensity values in fluorescence microscopic. Moreover, in modern era the amount of data to be handled and processed to extract useful information is growing exponentially. In addition to that, the information extraction inherent to an image is getting more and more challenging. Therefore, using such an image analysis is very time consuming and inadequate in assessing and quantifying human vision capabilities. The time effort required in large image datasets are not worth a shot. However, auto-quantification of cell features never refers to straight-forward problems. Thus, automated quantification techniques are needed to be devised.

Using statistical computing and its algorithmic efficiency, one can analyze images in a much shorter and intuitive way. By the use of computer algorithms and automatic extraction of information present in images, the time effort could be reduced and statistically more handy and appealing information extraction can be achieved. Therefore, a degree of autonomy (i.e. less human intervention) is induced in the information extraction process. The aim of this algorithmic computation based on image processing is to produce automatically an image interpretation close to the desired goals of information extraction from that particular image. Nonetheless, one would need certain parameters by changing which the outcome of an image processing routine is changed.

In order to evaluate the performance of new algorithms in comparison with standard methods, routines and parameter sets are validated by using benchmark datasets with a clear ground truth about objects contained in all images. Therefore, numerous benchmark datasets exist (e.g. datasets used in [44] for benchmarking and validation in biological image segmentation, [119] for cell image analysis by using simulated cell populations, [36] for semantic automatic image annotations using complex scenes, [102] contains large scale benchmark for event recognition in surveillance videos and many others). In one case, they contain highly complex objects or background as in large datasets e.g. [27, 52] such as animals and vehicles against a complex background. Benchmark databases for traffic signs [134], a street scene with pedestrians, a lion sitting on grass, visual event recognition in videos etc. also exist. They may also include complex rules of evaluation such as face or gesture recognition etc. In other cases, there are sets with incomplete ground truth [1] such as images containing cells with only the total number of cells as ground truth without any information of the type and shape of the objects to be segmented.

1.2.5 Applications to biological datasets

A lot of work has been done to date about application of image processing and analysis methods in various fields. Currently, one of the more eye-catching domains is the application of automatic image processing algorithms to the field of biology. It has been therefore nowadays, become a norm to develop sophisticated algorithms for efficient and robust cell detection, segmentation, classification algorithms for comforting the biologist to interpret different inherent information usually got from images acquired using a microscope.

The use of machine learning in the analysis of huge microscopic image data has become norm. A general pipeline used for cell image analysis is akin to one shown in Fig. 1.1. Machine learning techniques used in an image processing pipeline are now being optimized for large scale bio-image applications. The long-term target in this regard is to replace many manual analysis pipelines to help assay development while increasing the processing throughput, accuracy and efficiency [131].

Different researches related to cell image segmentation, robust feature extraction, classification of zebrafish embryos, small molecule microarray analysis etc. have already been conducted and are being furthered to increase robustness and efficiency of new and old image processing algorithms.

Robust feature extraction algorithms and classification of inhomogeneous datasets using these algorithms in fields of biological image processing have been discussed in [114]. Side effects for e.g. different experimenter or time difference etc. affect the outcome of an experiment. An example of it is a high-throughput screening of zebrafish larvae using image processing by employing gray scale values, morphological properties etc. Automated image feature extraction is robust and performed using filter approaches and wrapper approaches for segmentation or for optimizing extracted features. Wrapper approaches are used to optimize robustness w.r.t. side effects and quantifies the side effects based on data mining routines. The parameters are then optimized sequentially to minimize influence of side effects.

[1] http://www.broadinstitute.org/bbbc/

Wrapper approaches use the classifier output for adaptation of the parameters to reduce the influence of side effect through feature extraction and feature selection. These approaches, akin to feedback based automatic processing, deliver better results but are computationally intensive. The scheme is given in Fig. 1.2. The normalization of the images was performed before processing them further. Adaptation is done using different normalization procedures for e.g. normalization based on statistical parameters or quantile values [114].

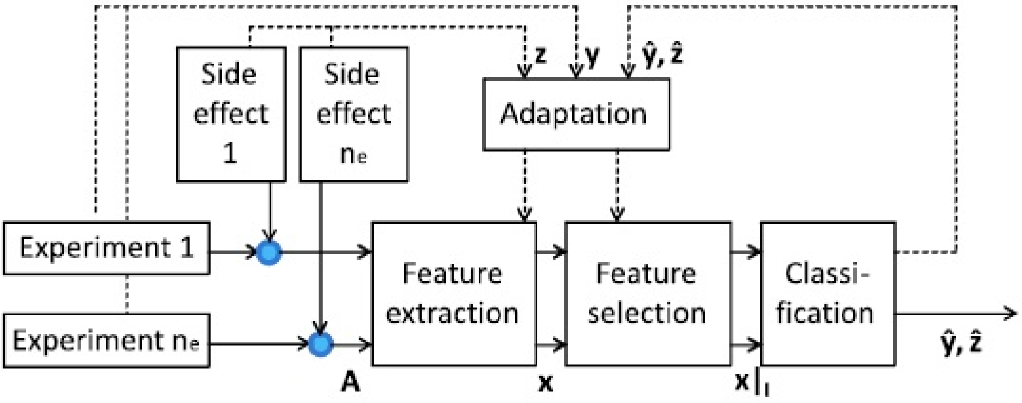


Figure 1.2: Intended and unintended properties in an image processing pipeline [114]

A similar work related to feature extraction and plausibility check was shown in [13] using small molecule micro-array images. However, it dealt with image processing challenges based on background normalization and smart feature extraction without iteratively adapting the image processing parameters (i.e. filter approach).

1.2.6 Bio-image processing tools and softwares

The use of mathematical formulation and computational power to extract useful information from image datasets is becoming necessary and challenging. This is achieved using algorithms and methods developed in diverse fields such as computer vision, image processing, computer science etc. [20]. Such methods are further developed in such a way that facilitates the development of software for acquisition, analysis and visualization of the imaging data. Consideration of each computational step that biologists face when dealing with images is done in addition to the inherent challenges and the overall status of available software for bioimage informatics [35]. A vast number of image-analysis algorithms and software packages are recently seen that have been developed for biological applications. The software packages are distinguishable based on certain factors i.e. application domains, usability, copyrights, source code and cost and detailed comparison is given in [148]. Since there are already a large number of open-source and commercial solutions present currently, the choice of applying a proper tool appropriate for a given task is always tricky [35].

Currently, there are a number of image processing and data mining softwares available in the field of cell biology. One way to broadly classify them is according to their free availability. They can be dichotomized as:

1. Commercial softwares
2. Open-source softwares

Commercial softwares

These softwares are not freely available to be used. Although, there are numerous commercial softwares available currently, the more widely used ones are given below:

- ACIS - Automated Cellular Image analysis System (ChromaVision Medical Systems, Inc., San Juan Capistrano, CA) [6]
- Definiens [26]
- Icy [20]
- Fluoview and ScanR (Olympus)
- Imaris (Bitplane)
- Volocity (PerkinElmer)
- ZEN Lite (Zeiss)
- MATLAB plus Image Processing Toolbox (MathWorks)

Open-source softwares

These softwares are freely available to be used. Some more commonly used open-source softwares are mentioned below:

- BioImage Suite [104]
- BioImageXD [68]
- CellProfiler [18]
- Gait-CAD image processing extension [94, 136]

- Interactive Learning and Segmentation Toolkit (ILASTIK) [132]
- ImageJ [120]
- Insight Segmentation and Registration Toolkit (ITK) [65]
- Konstanz Information Miner (KNIME) [8]
- Open Microsoft Environment (OME) [46] and OMERO (OME Remote Objects) [2] for metadata, image storage and basic image processing
- Vaa3D (3D visualization-assisted analysis) [108]
- VisBio [118]
- XPIWIT-An XML Pipeline Wrapper for Insight Toolkit [5]

All the above mentioned softwares have their own advantages and disadvantages. The choice of the software depends highly upon its usability, adaptability, extendability and efficiency. Combination of image acquisition and analysis is present in many commercial software packages such as Metamorph (Molecular Devices), SlideBook (3i), Image-Pro (MediaCybernetics) and Volocity (Perkin-Elmer). Most of them were commercialized later. Moreover, major microscope companies provide their own software packages, such as AxioVision (Zeiss), NIS-Elements (Nikon), cellSens and ScanR (Olympus) and ACIS (ChromaVision Medical Systems, Inc.). The commercial image-acquisition packages have an advantage of giving solution to standard image analysis strategies (acquiring individual images, taking time-lapse series, collecting three-dimensional (3D) stacks at multiple x-y positions and so on). However, it is nearly impossible for individual researchers to improve and extend these packages significantly or to make substantial custom hardware changes to the imaging system. This limits the scope of software usability to within laboratories that strictly are using hardware from same inventors [35].

Against the backdrop of frequently changing needs and equipment, researchers opt to write their own code. Software development is favorable in environments such as LabView (National Instruments) and Matlab (Mathworks). They provide interfaces to a certain set of available equipments and are viable for creating a graphical user interface (GUI). While toolkit environments provide high flexibility and high suitability to coders, they are less favorable in terms of distribution since infrastructure back-ends are costly and under-developed distribution channels.

Two open-source software projects, μManager [32] and ScanImage [113], which are developed and sustained by researchers, are thought to provide tools having more flexibility and greater usability than the toolkit environments in terms of microscope control. μManager mainly deals with camera-based imaging in addition to being used for scanning systems. The inclusion of easy-to-use interface in μManager running as an ImageJ plug-in enables researchers to design and execute common microscopy functions as well as customized image-acquisition routines.

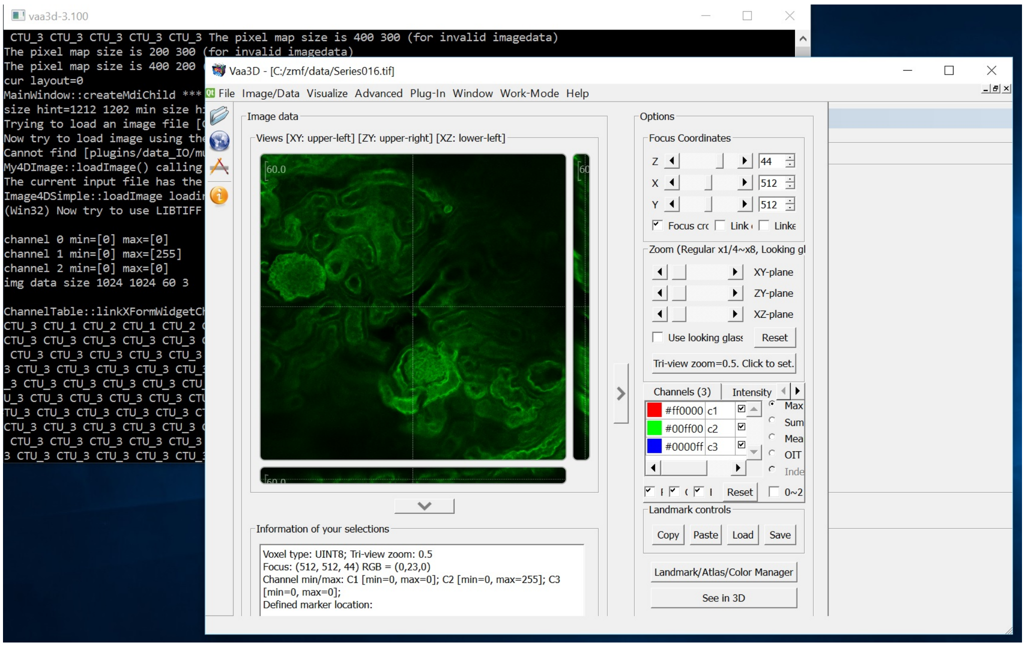
The solutions are then efficiently distributed as scripts or plug-ins. μManager's hardware abstraction layer can also be used without its ImageJ user interface in environments such as Icy, Matlab, LabView and Python, in order to include new approaches and methods for the development of the software [35].

Although there is no denying the importance of every step in the bioimaging pipeline, the heart of bioimage informatics is, of course, the images themselves and methods for their analysis and visualization. These two processes are inextricably linked, and image data derived from different imaging modalities, applications and experimental designs require a rich diversity of ever-evolving tools and techniques to extract biologically meaningful quantitative data from different types of microscopy images. A high-level overview of the tools available, related online resources and cited papers for detailed information are given in [35].

Categorically, there are two types of image analysis tools: 1) Niche image-analysis tools 2) Generalist image-analysis tools [35]. Former are image-analysis software packages that have been developed in academia and are formed to strictly accomplish problem-specific tasks e.g. the fluorescence association rules for multidimensional insight (FARSIGHT) toolkit, which grew out of the need to map the glio-vascular substrate of brain tissue surrounding neuroprosthetic devices. The latter type is those that can address more general problems. These software packages offer greater flexibility to multiple applications due to being typically modular in nature. Some examples of this category include MetaMorph, Amira (Visage Imaging), Volocity, Imaris (Bitplane Scientific Software), NISElements, SlideBook, ImagePro Plus (Media Cybernetics) and ZEN (Zeiss); usually provided by microscopy companies sold together with imaging instrumentation. There are many open-source image-analysis solutions originally developed to solve the needs of a particular community but later used or expanded to other purposes, such as BioImageXD [68], Icy [20], Fiji [120], Vaa3D (see Fig. 1.3) [108], CellProfiler [18], Reconstruct [41], FluoRender [149], ImageSurfer [40] (see Fig. 1.4(a)), OsiriX [117] and IMOD [78].

Even many generalist image-analysis tools are being developed and utilized for specific projects although they are usable for most tasks, they still have more emphasis on aspects of image analysis that are more suitable according to the description of the project for which they were designed for in the first place. For example, Fiji is currently more suitable for analysis of electron microscopy data, Icy has unique features pertaining to behavioral analysis, cell segmentation, tracking etc., and Vaa3D is strictly inclined to neurobiology applications (high dimensional image data rendering and analysis) and, together with BioimageXD, offers the best facilities for 3D visualization. ImageJ (originally called NIH Image) has a marked attribute for being used for the longest time period and being free always (see Fig. 1.4(b)). Consequently, it is most popular with hundreds of plug-ins later integrated into it.

CellProfiler is a flexible multipurpose open-source image-analysis tool that has a good track record in terms of utility and success. It contains highly curated modules that can be arranged and matched to create customized image analysis pipelines for a variety of biological systems including cells. It is designed to accommodate high-throughput analysis and is used to address

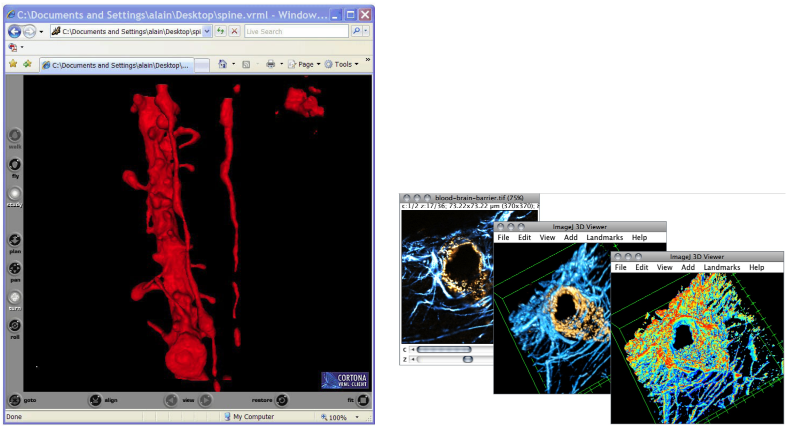


```
 CTU_3 CTU_3 CTU_3 CTU_3 CTU_3 The pixel map size is 400 300 (for invalid imagedata)
The pixel map size is 200 300 (for invalid imagedata)
The pixel map size is 400 200 (
cur layout=0
MainWindow::createMdiChild ***
size hint=1212 1202 min size hi
Trying to load an image file [C
Now try to load image using the
Cannot find [plugins/data_IO/mu
My4DImage::loadImage() calling
The current input file has the
Image4DSimple::loadImage loadin
(Win32) Now try to use LIBTIFF

channel 0 min=[0] max=[0]
channel 1 min=[0] max=[255]
channel 2 min=[0] max=[0]
img data size 1024 1024 60 3

ChannelTable::linkXFormWidgetCh
CTU_3 CTU_1 CTU_2 CTU_1 CTU_2 C
CTU_3 CTU_3 CTU_3 CTU_3 CTU_3 C
 CTU_3 CTU_3 CTU_3 CTU_3 CTU_3
3 CTU_3 CTU_3 CTU_3 CTU_3 CTU_3
_3 CTU_3 CTU_3 CTU_3 CTU_3 CTU_
U_3 CTU_3 CTU_3 CTU_3 CTU_3 CTU
TU_3 CTU_3 CTU_3 CTU_3 CTU_3 CT
CTU_3 CTU_3 CTU_3 CTU_3 CTU_3 C
 CTU_3 CTU_3 CTU_3 CTU_3 CTU_3
3 CTU_3 CTU_3 CTU_3 CTU_3 CTU_3
```

Voxel type: UINT8; Tri-view zoom: 0.5
Focus: (512, 512, 44) RGB = (0,23,0)
Channel min/max: C1 [min=0, max=0]; C2 [min=0, max=255]; C3 [min=0, max=0];
Defined marker location:

Z 44
X 512
Y 512
☑ Focus crc ☐ Link ☐ Linke
☐ Use looking glass
Tri-view zoom=0.5. Click to set.
#ff0000 c1 ☑
#00ff00 c2 ☑
#0000ff c3 ☑
● Max ○ Sum ○ Mea ○ OIT ○ Inde
☐ 0~2

Figure 1.3: An example image for the software environment of Vaa3D for reading and displaying an image stack.



(a) (b)

Figure 1.4: Example images for bio-image analysis software tools where Fig. 1.4 (a) ImageSurfer2 and (b) ImageJ.

several application areas, including intensity and morphology measurements, phenotype scoring by machine learning and object tracking.

Automated biological cell image classification

Automated and robust image classification is indispensable when it comes to dealing with biological cells. Different manual and automated methods and techniques with respective software implementation exist ([34], [39], [45], [79], [110], [121], [122], [129]). In the scope of human cancer cells detection, one needs to put effort in automatically segmenting cells from diverse and noise-polluted backgrounds in images in order to classify further different cell states as shown in the Fig. 1.5.

The knowledge of different cell states (i.e. normal, living, dividing, dying cells etc.) would enable biologists to acquire useful information about patient status and future actions in a broader picture. Therefore, the ultimate goal is to use such techniques in order to produce desirable results in challenging applications such as cell segmentation and cell state classification. Different methods and their implementations in software tools discussed above perform different tasks on cell images. Although they produce good results according to the task assigned to them, the different subclasses of cell states in such a setting is quite eye-catching and a laborious task to perform. Some commercial softwares are able to classify some cell states in samples containing human cancer cells but they still are unable to classify certain desirable cell states such as mitotic (dividing cells) and different subclasses of dying cells. To this date, it remains a challenging task to have this functionality in a user-friendly software easy to be used by biologists. In the process, potentially new cell states can also be discovered based on diverse features extracted using automatic schemes.

A lot of work has also been done on analysis of living and dying (i.e. apoptotic, necrotic) cells as given in [58, 59, 66, 71, 96, 112, 125]. The assessment of exercise-induced immune cell apoptosis using morphological image processing was done in [96] using deformable models. Automatic quantification of apoptosis in animal cell culture using grayscale images and multichannel information was done in [112]. The algorithm consisted of a number of morphological image operators with carefully selected parameters. There has also been work available in the direction of different cell types classification ([97]). This work employed a software for high-throughput time-lapse cell imaging in order to detect different cell states i.e. interphase, mitosis, apoptosis (several cell death phenotypes with supercompacted or fragmented nuclei including, but not limited to, apoptosis), shape (nuclei of abnormal shape including but not limited to, a high percentage of binucleated cell) and artifact (containing other objects). The classification was reported to be 97 % accurate compared with manual annotation. Segmentation was done using an optimized method of local adaptive thresholding and automatic classification was performed by supervised machine learning using multiclass Support Vector Machines (SVMs).

In a similar work [98], a computational pipeline based on morphology recognition by supervised machine learning was used. About 200 features were extracted from each nucleus and used for classification into one of 16 morphological classes by SVM classifier trained on a set of 3000 manually annotated cells. The classifier, thereby obtained automatically recognized morphological changes due to the cell cycle, cell death or other phenotypic changes with an overall accuracy

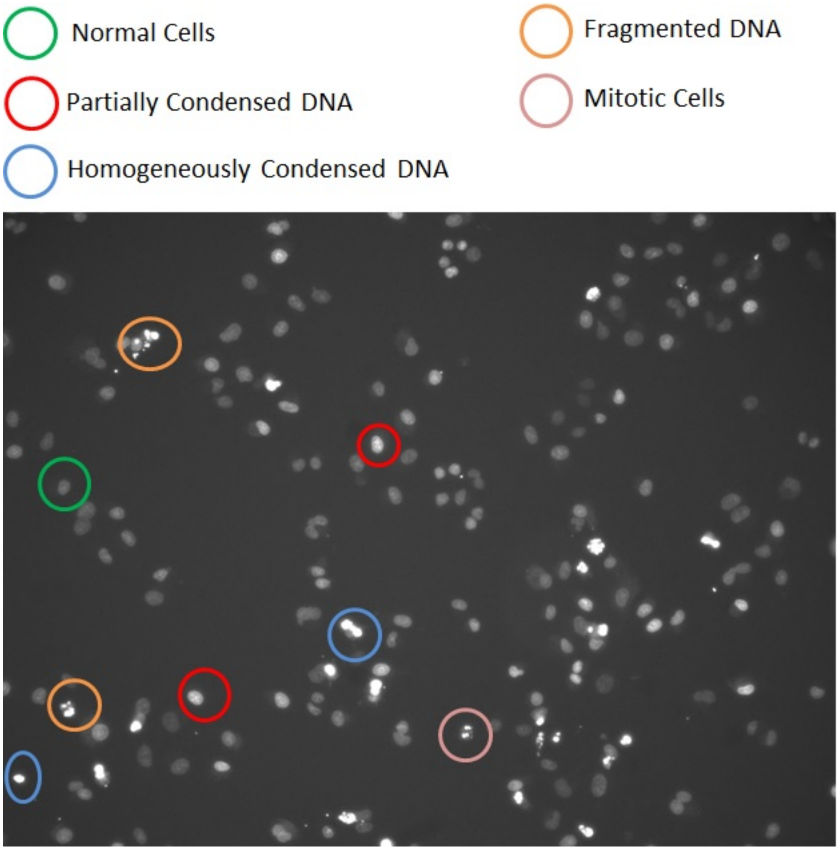


Figure 1.5: Different cell types: challenging for an accurate classification

of 87 % [98]. The classes were extended to several subclasses having different morphologies observed over time. An automated cellular analysis system was proposed in [77] where morphological changes occurring in transfected HeLa cells were detected and novel genes that induced mitotic phenotypes were found.

Another work lately [107] presented a population-based modeling approach to quantify dynamic phenotypes from time-lapse cell imaging assays. The temporal information helps to localize the timing of events such as cell death, mitotic arrest or quiescence, and to estimate the duration of processes such as mitosis. 16 nuclear morphologies were grouped into four cellular states recapitulating the cell cycle: interphase, mitotic, polynucleated and dead. An ordinary differential equation (ODE) model was used to characterize the dynamic transitions between the four populations (Fig. 1.6). Recently, large-scale tracking and classification for automatic analysis of cell migration and proliferation, and experimental optimization of high-throughput screens of neuroblastoma cells was done in [57] using four different cell classes i.e. interphase, mitosis, apoptosis and artifact.

The main focus in this research work would be on cell image analysis, since it is quite challenging against the backdrops of varying image intensities, inherent acquisition noise, shading

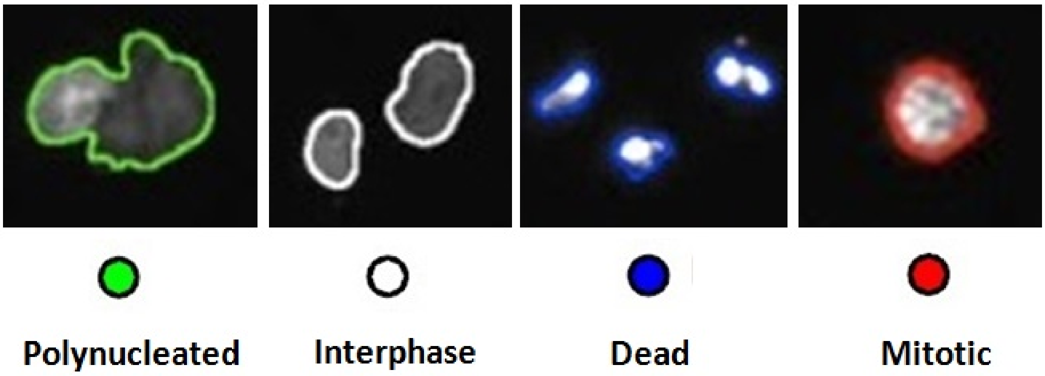


Figure 1.6: HeLa cells image having interphase cells outlined in white, mitotic cells in red caused by prometaphase arrest, polynucleated in green and dead in blue [107]

phenomena, variety of cell types containing a high range of feature variations. A recent overview on the advances of cell analysis in terms of cell morphology can be studied in [21]. In [29], analysis of infected blood cells using morphological operators in addition to watershed segmentation algorithm was used to detect these cells. They aimed for an improvement on already existing watershed segmentation seeking better cell separation. Therefore, dataset from human cancer cells was chosen. The dataset consists of images showing human lung cells (A549) treated with the anticancer drug cis-platin for 24 hours and representative images were acquired as described previously in [30]. The dataset has a variety of different cell types to be classified and discussed in later sections. Another method presented in [60] elucidates correction of segmentation errors in the automated cell image analysis in time sequence images. It employed a variety of cell tracking and segmentation algorithms. Live tracking of dividing cells by use of image processing was elaborated in [81].

1.3 Open problems

- Selection of parameters play important part in keeping the outcome of image processing routines closer to desired goals. For the sake of simplicity, assumption of gray level threshold t in a gray scale image that contains round objects in the foreground against a darker background could be considered in case of image segmentation. The segmentation outcome based on the selection of t in the presence of reference information would enable the quantitative objective evaluation of the image processing routine. However, one never knows the correct value of parameter that would produce the best results (i.e. minimal error of outcome compared to reference). This is commonly referred to as feed-forward automatic image processing technique, thereby requiring a new value of t and analyzing the ensuing results. The optimal value of t may be found out after several iterations of an image processing algorithm. This effect is shown in Fig. 1.7.

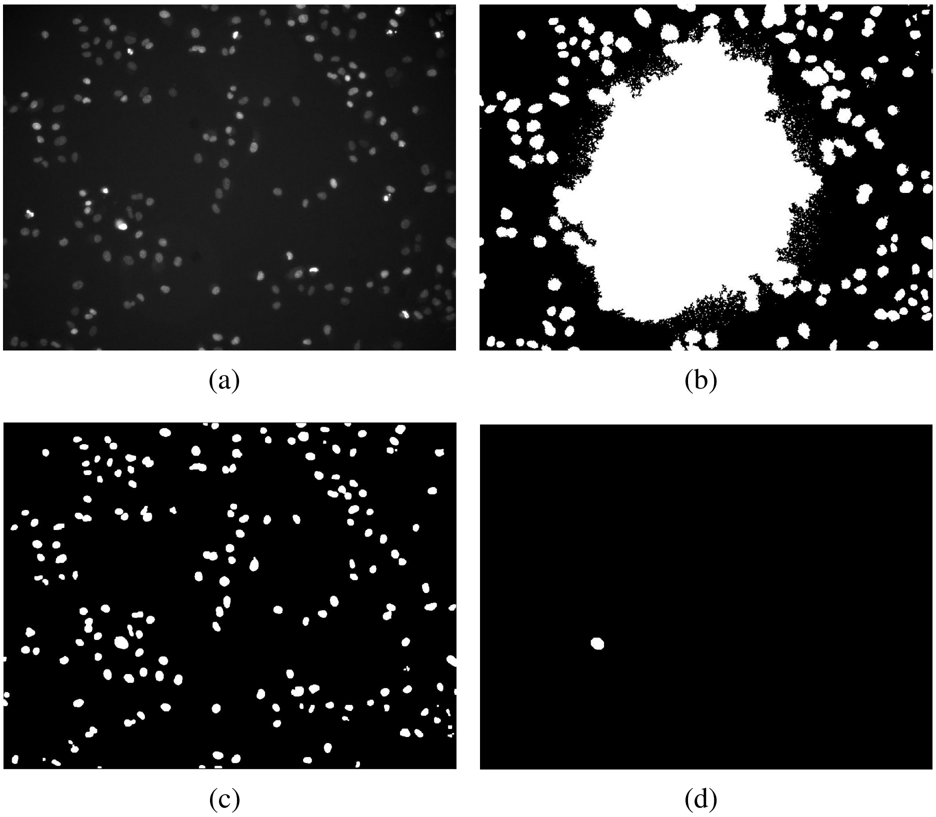


Figure 1.7: Segmentation results using manual selection of parameters involved in an image segmentation procedure, where Fig. 1.7 (a) Original grayscale image and (b), (c) and (d) show segmented binary images resulting from different parameter vectors

There are different methods on the horizon using both *supervised* and *unsupervised* image analysis techniques. The latter are however subject to interpretation which in turn is hinged on the problem type and data points class and number to be found. This can be problematic considering huge amount of data points with diverse classes in addition to thick overlap of the feature space based on feature extraction. This can produce erroneous results (e.g. unknown classes, class inseparable by the user etc.) even if the hyperplanes in the feature space are carefully designed. On the other hand, *supervised* methods tend to solve this by carefully feeding the segmentation and classification algorithms with examples of class types to be found. The *supervised* techniques that are available are mostly feed-forward. They can cause problem when introduced to diverse datasets with lot of background noise and artifacts. The available feedback-based *supervised* algorithms are more suited for applications where robustness is of chief importance. However, they are quite few (e.g. [114]) and also lack in exploiting fore-knowledge of available data points to be found in a much more intuitive and efficient way. So, adaptation of image processing parameters is of paramount importance. Mainly, the usual image processing parameters don't deliver robust results when using non-adaptive techniques. And in cases where adaptive techniques are used, parameter tuning still has the room for refinement.

- Algorithms that are robust to noisy conditions and varying artifact levels (including very high levels) should be devised. Additionally, they should deliver results automatically in different datasets i.e. biological, other non-biological datasets with different object features and classes.
- In order to evaluate the outcome produced by various algorithms using different parameter selection methods, one needs to have a good benchmark dataset. This dataset should contain ground truth in terms of both the segmentation and the class type of objects in the scene. Moreover, such a dataset should also contain artifact levels in terms of both artificial background noise and varying shading levels in order to evaluate the robustness of image processing pipelines.
- In a challenging environment of cell image analysis, especially human cancer cells, a lot of cell states are emerging based on the type of investigation which the cells are put under. In addition to normal healthy and mitotic cells, there are different dying cells (early apoptotic stage, late apoptotic stage, necrotic etc.). Moreover, the analysis is further complicated by some cells contained partially condensed DNA or fragmented DNA. To devise a software solution, that is not only user-interactive and efficient in analysis but also is able to classify diverse and challenging cell states with acceptable accuracy, is currently becoming a need of the hour in bio-cell imaging communities.
- A core problem remains to use efficient and completely automatic image processing and analysis algorithms on challenging datasets (e.g. image containing different human cell types with noisy background) using abstract *a priori* knowledge about segments to be found. Hence, to supervise the image processing algorithms with a well-formulated abstract knowledge, a comprehensive and efficient method is also required.

1.4 Proposals

The image processing algorithms and routines face restrictions in optimal parameter selection with increasing number of parameters involved and the combinatorial problem lying therein [7]. Thus, the optimal parameter set may not be found manually, while the tuning is very time-intensive and subject to repetitions on the arrival of a new dataset. In order to produce an optimal *supervised* image processing result, optimal parameters set should be found out. In case of image segmentation and simplistic description including only t, image processing routine should find optimal value of t, such that no or minimal human intervention is involved. The algorithm only requires a definition of search space to which the optimal value of t belongs and initializing the algorithm with it in such a way that the algorithm automatically adapts t which yields best segmentation outcome. This may require several iterations but no human intervention. Such image processing techniques are referred to as feedback automatic image processing technique. This consequently saves a lot of time and ensures an optimal or close to optimal results. This is a major aspect of research in this thesis, in order to improve the parameter tuning using feedback methods whereas other methods available hardly deal with such room for parameter perfection.

The aim of this thesis is to device such feedback-based techniques in order to adapt parameters related to different processing steps included in a typical image processing and analysis pipeline. The proposed algorithm which would perform ultimate image object classification task is divided into several steps. More straightforwardly, it contains a sequence of steps that such that results from each step are propagated to next step. The work done in feed-back tuning of parameters in image analysis and processing is limited and its applications to challenging tasks especially in the field of biology are scarce. Therefore, this thesis tends to improve the deficiencies in the existing methods and incorporate a well-formulated search space for optimal parameters adaptation.

The feedback technique can be used in variety of different operations in image processing pipeline. For one, shading correction and background noise removal can be done in an iterative manner. A brief overview of variety of *retrospective* methods for shading correction in term of applicability and comparison are given in [83, 87, 144, 147, 162, 63]. However, these methods are limited in terms of well-formulated search space for ascertaining optimal normalization parameters and integration of segmentation results in addition to internal consistency check for the performance of normalization routines. The parameters should be adjusted automatically in an iterative manner for improvement of subjective and quantitative results based on *a priori* knowledge. The use of different computational intelligence techniques (such as neural networks, fuzzy sets [16], genetic algorithms [23] etc.) in formulating and solving image processing tasks is getting more common nowadays. Therefore, a new method for an automatic feedback-driven design of normalization technique for tuning processing parameters using fuzzy *a priori* knowledge and segmentation results based on adopted normalization technique is proposed. In contrast to fuzzy formulations in [137, 138, 135], fuzzy sets are used here as an optimization criterion and not only as an auxiliary measure for image operators. Feedback-driven automatic approaches for segmentation have already been employed in [7, 38, 72, 114, 143] but application to normalization is yet to be explored deeply.

The implementation of automatic image segmentation and image object classification in the domain of biological cell imaging in downright challenging. With emerging number of diverse cell states, a user-interactive software that is able to differentiate different cell classes including more intricate and peculiar ones like fragmented cells, necrotic cells in addition to partially condensed and different stages of apoptosis has been proposed in this work. The proposed method would not just only be able to adapt to different cell morphologies against a drifting background noise, but would enable user to annotate the whole cell image dataset with different cell classes with desirable accuracy.

In a nutshell, following proposals have been made in this doctoral research work:

1. To use supervised feedback-based automatic parameter tuning at different stages in an image processing pipeline and to integrate the results to improve image object classification

2. To use fuzzy formulation of *a priori* knowledge in conjunction with feedback-based automatic parameter tuning with fuzzy evaluation criteria.

3. To tune parameters to make supervised feedback-driven design of normalization/segmentation techniques more robust.

4. To segment challenging new cell classes in datasets containing human cancer cell using devised feedback-based automatic image processing techniques.

5. To develop new benchmark dataset (with absolute ground truth using different object classes) with progressive shading and artificial noise levels for performance evaluation standard processing algorithms.

6. To devise a platform to compare feedback-based parameter adaptation of image processing algorithms with standard feedforward algorithms.

7. Developing a GUI or enhancing existing ones for interactive and efficient implementation of devised algorithms and comparison with standard market softwares.

2 New methods for processing and analysis of multidimensional image data using feedback concepts

Feedback-based supervised automatic image analysis is proposed using fuzzy formulation of given *a priori* knowledge in order to perform an optimal image object classification. The *a priori* knowledge about the objects could be their shape and intensity features or this may also include some information of background and foreground object distribution.

As a proof of principle and for better perception of the proposed scheme, only grayscale images having adequate noise were used as to provide enough room for the algorithm and methods to be robust. The idea is to find the optimal parameters that affect the outcome of individual image processing steps. Therefore, it is necessary in the first step to define all the parameters involved in image processing and analysis. These parameters can be selected manually through hit-and-trial, heuristics or intelligent guesses to produce the optimal results. However, this process becomes extremely time-consuming and hectic as the number of parameters increases. Automatic tuning is the solution to avoid manual interruptions, minimize time-effort and to create a robust methodology of optimal parameters search. Parameters can be improved iteratively by looking into a predefined search space and selecting the ones that increase overall quality and evaluation criteria set for judging the outcome of these image processing routines. In this regard, the iteration should continue insofar as to achieve a quality described in terms of either some internal consistency checking criteria or some predefined knowledge i.e. *a priori*. This *a priori* is the reference according to which the parameters are adapted. The image processing algorithms and techniques used in such cases are *supervised*. Once the optimal parameters are found, they are used for final image processing and analysis routines. Each step involved in an image processing chain can be used in such a way i.e. adapt parameters for one step automatically and iteratively and pass on the optimal result based on selected parameters to the next step and so on.

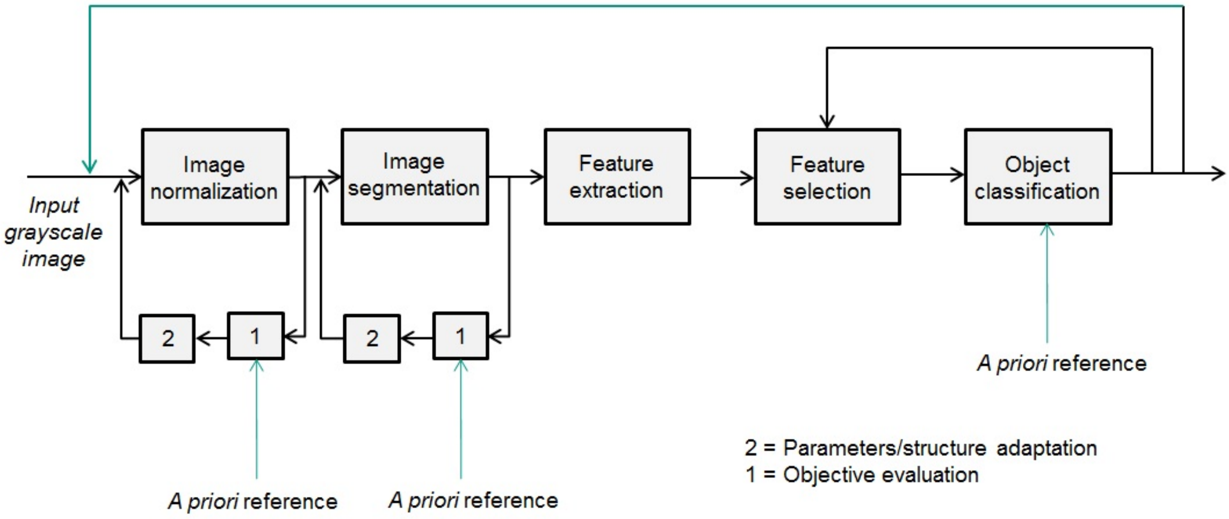


Figure 2.1: Overall feedback-based supervised automatic image object classification scheme

2.1 General concept

A *supervised* and feedback-based automatic image processing technique using a classical image processing and analysis chain has been proposed in this research work. The individual steps of such a scheme are shown in Fig. 2.1. It is clear from Fig. 2.1 that the image pre-processing and image segmentation steps are handled using feedback and parameter adaptation concepts in the presence of *a priori* knowledge. Similarly, parameters for other steps can be adapted. The scheme is proposed to be applied to real world applications e.g. human cells biological image datasets obtained from microscope, irregular shaped objects against a varying background etc. The proposed techniques are to be applied on challenging datasets such as biological cell image since it is quite attractive and challenging, not to mention an increasing trend and high commercial demand of such techniques and their translation to end-users. At the same time, it should be applicable to other similar datasets containing different object types against a noisy and shaded background. These nascent techniques are expected to produce more optimal results by simply increasing the scope of *a priori* knowledge by using fuzzy logic and interpreting the results in fuzzy sense in order to increase the intuition and scope of objective evaluation and reference formulation.

The basic concept of this work is efficiently usable even if the routines involved are *unsupervised* i.e minimizing the evaluation criteria based on how well the object classes/pixels are grouped together to adopt certain image segmentation parameters iteratively. It is also applicable to already devised techniques and algorithms in order to obtain the optimal parameter set using these techniques while integrating the quality and evaluation criteria in the same fashion.

In this research work, not only the specific task of image object segmentation dealt with but also the robust feature extraction and efficient object classification techniques. A great emphasis has been put to generate a greater versatility of the techniques and methods by simply using different image processing routines and optimizing them in terms of the parameters required for the image processing steps under consideration. The incorporation of proposed methods into a graphical user interface is targeted that not only helps end-user to create pipelines to conform a specific problem but also to handle different image processing tasks. The intent is to develop such an application using multipurpose open-source image-analysis platform such as MATLAB. An open-source toolbox named Gait-CAD [136] used for image processing and data mining has already been on horizon having different possibilities to conform the algorithms and method to a particular problem such as the usage of macros, scripts, plug-ins etc.

2.2 Benchmark data

2.2.1 Pre-requisites

In order to evaluate the performance of automatic feedback-based algorithms in comparison with standard methods, some clean benchmark images with clear ground truth are required. However, the proof of principle demands benchmark images that are both universal in properties and useable with variety of other comparable image processing algorithms. Certain desired properties in a dataset are sought after i.e.

- containing different object types (having different shapes, sizes, mean intensity values etc.)
- having distinguishable background with respect to the objects to be found
- having distinct number of total objects

However, in order to show robustness and adaptability of the algorithms, a dataset with images containing challenging properties is required:

- background shading (using variety of illumination conditions)
- background noise (such as salt and pepper, Poisson etc.)
- overlapping objects (including different types)
- unknown objects (to be rejected by the algorithm)

Different benchmark datasets were used to test algorithms (see Tab. 2.1 for comparison). One dataset (i.e. Benchmark dataset 1) was created to ensure all these effects and artifacts. The Benchmark dataset 2 [1] was based on human lung cancer cells [30] and Benchmark dataset 3 [140] was based on images of simulated cells resembling human cancer cells. All benchmark datasets are described below:

Benchmark number	**Absolute ground truth**	**Artifact type**	**Labeling effort**	**Number of images**
1	object boundary, class type	gradual shading, artificial noise	moderate	728
2*	not available	inherent (acquisition noise)	high (if done manually)	10
3	object boundary, class types (few)	artificial noise (salt and pepper etc.)	low	30
4	cell count	inherent (microscopic noise)	not available	6

Table 2.1: Comparison of 4 benchmark datasets used for evaluation of image processing and analysis algorithms. Here, asterisk sign shows that the dataset used here for Benchmark dataset 2 is a subset of a bigger dataset (mainly application dataset) described later in Chapter 4.

The labeling effort in Benchmark dataset 2 is high due to size of the dataset and number of segments with different class types involved. Therefore, only a part of dataset 2 is used as Benchmark dataset 2. The rest is used as application dataset. The task is further complicated even by an expert due to presence of different cell states having overlapping features and boundaries touching each other. From here on, the collection of 10 images would be referred as Benchmark dataset 2.

2.2.2 Benchmark dataset 1 - Irregular shaped solid hardware items

A major problem in image processing are alterations in image data sets due to shading, noise etc. Therefore, not only the absolute quality of a segmentation algorithm is important, but also robustness. To evaluate robustness, the segmentation quality in relation to the intensity of the alteration is a good measure. To derive this measure for a new routine, the benchmark must not only contain the ground truth for segments but also image variations of the mentioned effects. The strength of these effects needs to be given for each sample.

Furthermore, if the subsequent outcome of a classifier is to be evaluated, not only a ground truth in segmentation but also in object labels is required. For classifier validation, there are

[1] Benchmark dataset 2 contains only a part (10 images) of the biological dataset containing human cells. It contains 314 images in total, and the rest 304 images are used as an application dataset later described in Chapter 4.

plenty of benchmark datasets (IRIS [3] or WINE [42]), a combination of both has not yet been published. As well, there are robustness and evaluation measures for image segmentation [139], but no consistent methodology to evaluate both, the combination of image segmentation and classification.

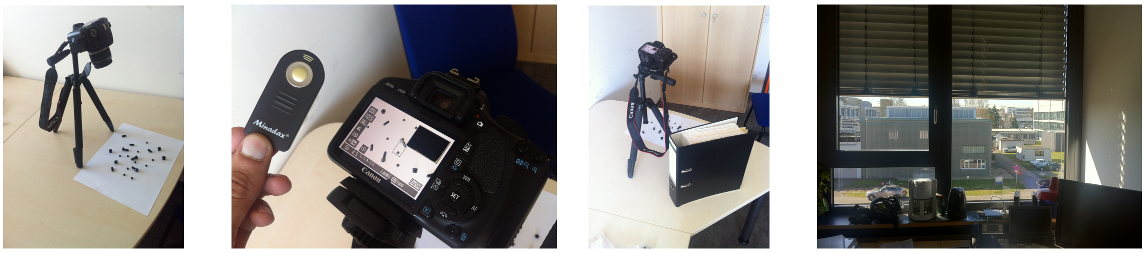
Therefore, a new benchmark data set was introduced for the validation of image processing robustness and classification. It contains easy-to-find segments of different object types (having different shapes, sizes, mean intensity values etc.) with given ground truth in segmentation and classification. Furthermore, effects like shading, noise and overlapping objects are inserted. Altogether, four datasets are introduced, each containing labeled segments of 5-9 object classes. The datasets contain a series increasing shading intensity and noise level.

Furthermore, the aim is to provide a basic image processing routine and deliver initial results on the dataset regarding accuracy and robustness. To give other developers the possibility to compete with this routine, robustness measures evaluating the success of the image segmentation and the classification are introduced. The sources for the image-processing routine, the robustness measures and the dataset as well are freely available and downloadable under [2].

This dataset is specifically designed to conform to evaluation criteria that are most suitable for the new methodology. It employs shadowing and shading created using the experimental setup. The images were taken using 9 different objects (i.e. different size, shape and intensity). These objects were used to create different datasets by introducing varying shadowing and shading using lamp lights and sunlight controlled by a rolling curtain. The experimental setup is shown in Fig. 2.2(a). Images were taken using a professional camera with a remote trigger (see Fig. 2.2(b)) in order to keep the object positions the same for all lighting conditions. Images were taken under varying lighting and shadowing conditions. The shadowing conditions were created using a paper documents folder on different sides of the experimental setup to constrict sunlight coming from the window. Three different datasets were taken using different object types and different object settings. Each dataset contained an image using full lighting conditions with camera flash light ON (see Fig. 2.3). This image was used to label the objects semi-automatically (first using Otsu segmentation and then manual correction in Windows Paint Application), both in terms of their type and object boundaries (see Fig. 2.5). As an example, different lighting conditions for Benchmark dataset 1 are shown in Fig. 2.7.

The benchmark consists of images $\mathbf{I} = ((I_{ij})) \in \mathbb{N}^{m \times g}$ with $m = 1000, g = 1500, I_{ij} \in [0, 255]$ with variations in brightness/shading and noise. There are $R_o = 4$ datasets, each combining $B = 13$ grades of shading and $N = 14$ grades of noise. Each dataset contains different individual scenes. For example, one dataset contains scene that has more classes of objects present in the foreground and are placed in such a way as to create an overlapping problem. Thus, altogether $B \cdot R_o \cdot N$ images are contained in the dataset (see Fig. 2.4). Images are saved as 8bit TIFF-Files, the naming convention is 'benchdata_r_b_n.tif', where r, b, n are numbering dataset, brightness

[2] https://sourceforge.net/projects/gait-cad/files/Benchmarks/hardware_items/



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

Figure 2.2: Experimental setup consisting of a professional camera mounted on a tripod with different objects placed on A3 white paper lying on the table (Fig. 2.2(a)), Remote camera trigger (Fig. 2.2(b)), use of paper documents folder for shadow construction (Fig. 2.2(c)) and Rolling window curtain (Fig. 2.2(d))

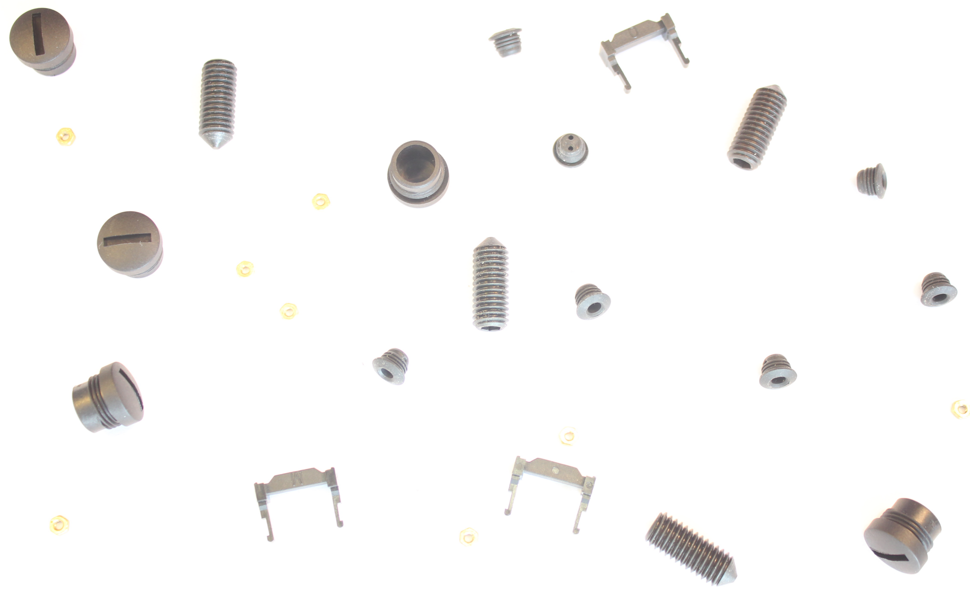


Figure 2.3: Benchmark image example with full lighting conditions and camera FLASH ON i.e. almost no shadowing

and noise level. Each number is given in two digits (e.g. 'benchdata_02_01_03.tif').

Images were obtained using different grades of shading i.e. $b = 1, ..., B = 13$ lighting conditions ($b = 1$: very bright, no shading; $b = 13$: dark, lot of shading).

To each taken image, artificial background noise in varying intensities was added. Altogether, 13 additional noise levels were generated using Gaussian-distributed random numbers with varying

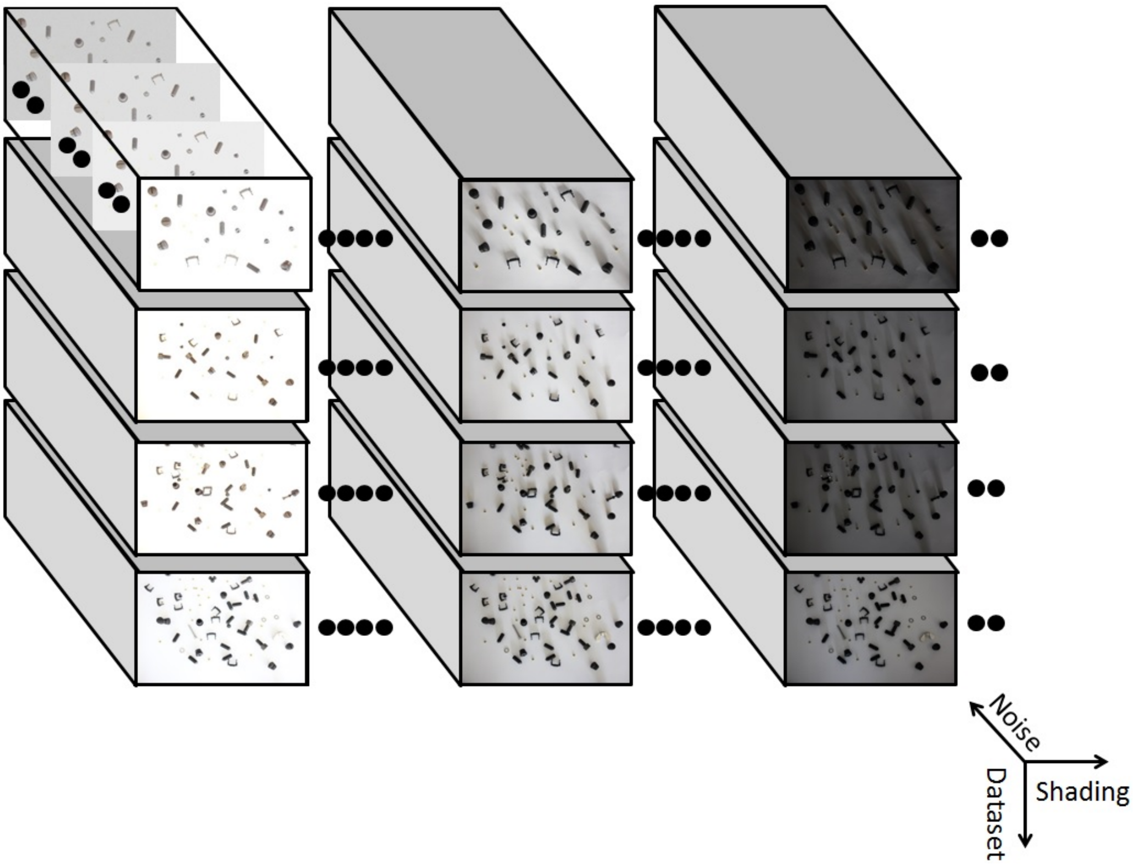


Figure 2.4: Benchmark dataset 1 - Overview of images [75]

standard deviations. Thus, the image $\mathbf{I}(r, b, n)$ denoting dataset r, brightness-level b and noise-level n is given by

$$I_{ij}(r, b, n) = f_u(I_{ij}(r, b, n) + I_{\text{rnd},ij}) \tag{2.1}$$

where $I_{\text{rnd},ij}$ is a realization of the Gaussian-distributed random variable $I_{\text{rnd},ij} \sim N(0, \sigma_n^2)$ with $\sigma_n^2 = 125(n-1)$. The function f_u restricts possible values to $[0, 255]$:

$$f_u(i) = \begin{cases} 0 & \text{if} \quad i < 0 \\ 255 & \text{if} \quad i > 255. \\ i & \text{else} \end{cases} \tag{2.2}$$

The maximum number O of objects in any dataset is 63 and maximum number of object types is $K_n = 9$. Object types differ in size, shape, intensity and reflection properties. Objects are not aligned and multiple viewing directions are possible. For example in Fig. 2.5, the object type numbered as 4 (see image on right) has different possible views (see image on left). For each dataset *r*, a ground truth in object classification and segmentation is given: $\mathbf{I}_{\text{truth}} = ((I_{ij,truth})) \in \mathbb{N}^{m \times g}$ with $I_{ij,\text{truth}} \in \{0, ...K_n\}$ (0: background, $1, ..., K_n$: objects). An exemplary image is depicted in Fig. 2.5.

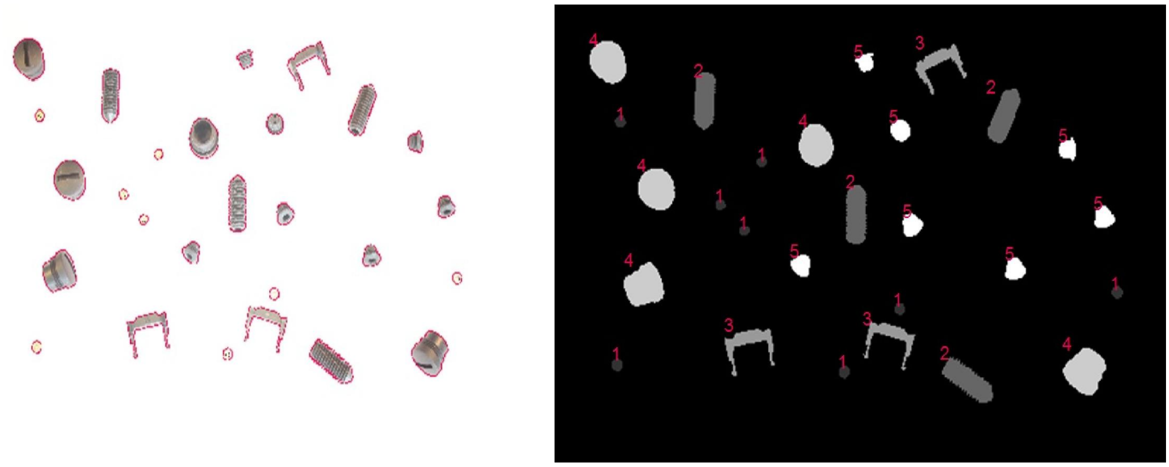


Figure 2.5: Left: Brightfield-image **I** of Benchmark dataset with marked object edges, right: Ground truth image $\mathbf{I}_{\text{truth}}$, with object types in numbers. Gray scales denote the value of I_{ij} (0: black (background), 5: white)

Artifact level: To quantify an algorithm's robustness versus shading and noise, an artifact level $A(r, b, n) \in [0, 1]$ which aggregates the shading level and the noise level is introduced.

The quantification of shading and noise is done using an artifact function $\mu \in [0, 1]$ based dependent on a quantifying parameter θ (e.g. mean brightness or deviation). The overall artifact level $A(r, b, n)$ is then calculated as:

$$A(r, b, n) = 1 - (1 - \mu_1)(1 - \mu_2). \tag{2.3}$$

The artifact function μ is monotonic with tunable higher and lower bounds α and β, to suppress a tunable percentage of higher and lower outliers which should not change the function value

$$\mu(\theta; c, \alpha, \beta) = \begin{cases} 1, & \theta \leqslant \alpha \\ 1 - 2^{c-1}(\frac{\theta-\alpha}{\beta-\alpha})^c, & \alpha < \theta \leqslant \frac{\alpha+\beta}{2} \\ 2^{c-1}(\frac{\theta-\beta}{\beta-\alpha})^c, & \frac{\alpha+\beta}{2} < \theta < \beta \\ 0, & \theta \geqslant \beta \end{cases} \tag{2.4}$$

In this benchmark, two different artifact functions i.e. μ_1 and μ_2 for background shading and Gaussian noise respectively as shown in Fig. 2.6 are included.

The parameter c defines the curvature: $c = 1$ is a trapezoidal function, $1 < c < \infty$ is a spline-based function, $c \to \infty$ delivers a step function.

To quantify the amount of shading and background present in $\mathbf{I}(r, b, n)$, the mean pixel value i_m across the whole image as a parameter $\theta = i_m$ is used: Since the benchmark contains dark objects

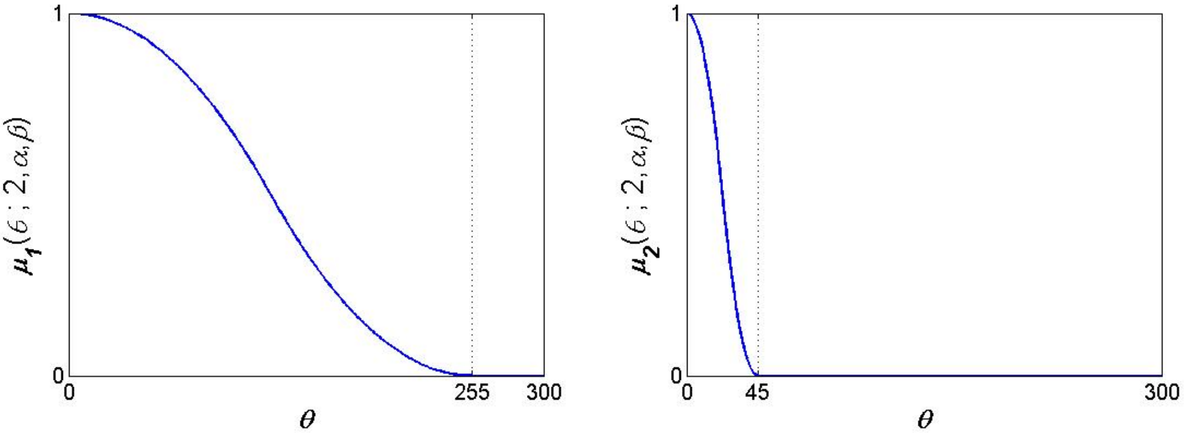


Figure 2.6: Spline based fuzzy artifact function showing the effect of parameter selection where dotted line indicates the selection of β value if α is kept at zero. Left: μ_1 with α = 0 and β = 255, right: μ_2 with α = 0 and $\beta = 45$

in front of a bright background, higher i_m values mean less shading and vice versa. Furthermore, $c = 2, \alpha$ = 0 and β = 255 is set for shading. Thus,

$$\mu_1 = \mu(i_m; 2, 0, 255). \tag{2.5}$$

On the other hand, the Gaussian variance is given as $\sigma_n^2 = (N-1) \cdot 125$. Thus, increasing values of σ_n denote a higher level of artifacts. Therefore, $c = 2, \alpha$ = 0 and β = 45 is set for noise levels. The value of β was selected so since values only in this range are obtained.

$$\mu_2 = 1 - \mu(\sigma_n; 2, 0, 45). \tag{2.6}$$

Quality measures for segmentation : Here, fuzzy evaluation criteria derived from aforementioned measures were used since it is intuitively understandable. The evaluation criteria are defined upon a segmented and classified image $\mathbf{I}_{\text{seg}} \in \mathbb{N}^{m \times g}$ with $I_{\text{seg},ij} \in \{0, ..., K_n\}$ (0: no segment, 1,...,K_u: class label) and the given ground truth image $\mathbf{I}_{\text{truth}} \in \mathbb{N}^{m \times g}$ with $I_{\text{truth},ij} \in \{0, ..., K_n\}$. $\delta_{\text{seg},ij}$ and $\delta_{\text{truth},ij}$ represent binary pixel values of $\mathbf{I}_{\text{seg}}$ and $\mathbf{I}_{\text{truth}}$ respectively.

Quality criteria in accordance with the number of detected objects (Segmentation measure 1 q_1 in Eq. (2.36) of Section 2.6.1) and their respective areas (Segmentation measure 2 q_2 in Eq. (2.38) of Section 2.6.1) and the classification accuracy (Classification measure q_3 in Eq. (2.44) of Section 2.6.2) based on the number of misclassified objects are used. It is specifically designed to incorporate further evaluation measures in addition to the necessary criteria for segmentation evaluation. These quality criteria were chosen because the absolute ground truth in object number and class in pixels exists. Since both class labels and object pixels are given as ground truth in Benchmark dataset 1, an overall criterion (2.46) was used for evaluation.

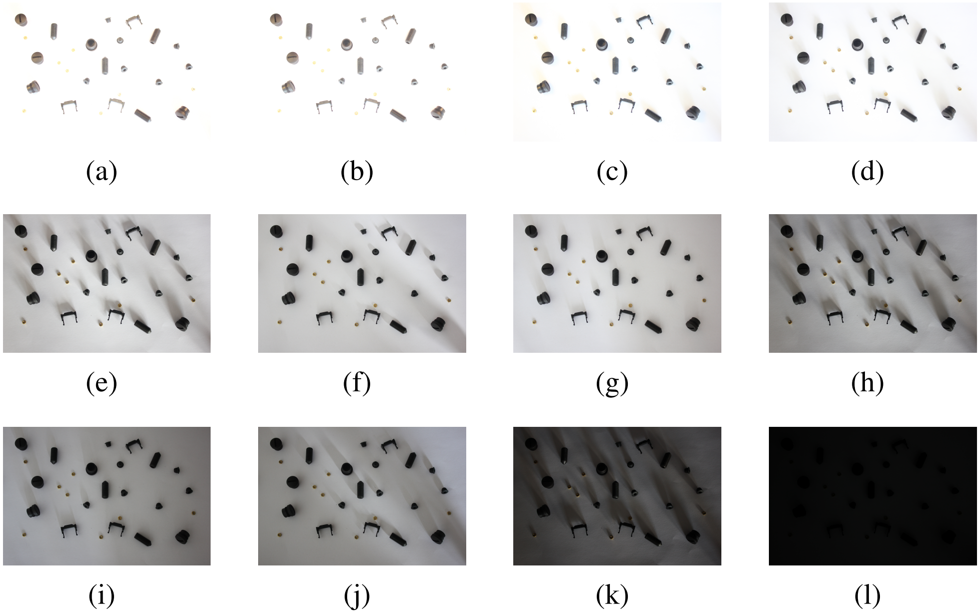
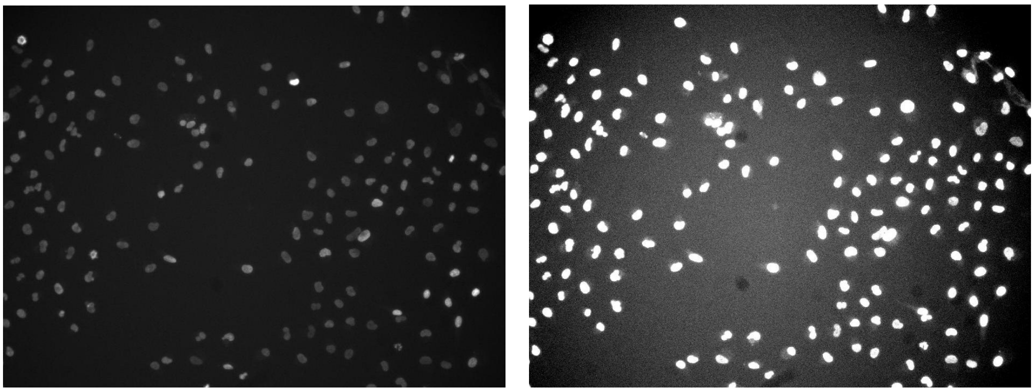


Figure 2.7: Benchmark dataset 1 using different lighting and shading conditions without using camera flash mode, where Fig. 2.7(a) shows brightest image with all lamps and ceiling light ON and curtain full UP, Fig. 2.7(b) shows image with ceiling light OFF, lamps ON and curtain full UP, Fig. 2.7(c) shows image with ceiling and right lamp OFF and curtain full UP, Fig. 2.7(d) shows image with all lights OFF and curtain full UP, Fig. 2.7(e) shows image with all lights OFF and curtain 25 % down, Fig. 2.7(f) shows image with all lights OFF, curtain 25 % down and light blocking by a folder on right, Fig. 2.7(g) shows image with all lights OFF, curtain 25 % down and light blocking by a folder on left, Fig. 2.7(h) shows image with all lights OFF and curtain 50 % down, Fig. 2.7(i) shows image with all lights OFF, curtain 50 % down and light blocking by a folder on left, Fig. 2.7(j) shows image with all lights OFF, curtain 50 % down and light blocking by a folder on right, Fig. 2.7(k) shows image with all lights OFF and curtain 75 % down and Fig. 2.7(l) shows image with all lights OFF and curtain 100 % down

2.2.3 Benchmark dataset 2 (10 nuclear staining channel images) - Human A549 lung cancer 1 cells

This is not strictly a typical benchmark dataset for performance evaluation of image processing algorithms. However, it does possess traits that makes it a very challenging dataset for object classification. This was included as benchmark dataset since a number of different classes of objects in the whole dataset were labeled manually by experts.

The aim of this dataset is to provide images of different challenging object types i.e. cells. The images are taken by microscope and contain inherent additive and multiplicative acquisition noise (see Fig. 2.8). Therefore, this dataset provides a challenging real-life application problem for image processing routines in terms of segmentation and classification. This is the only multichannel dataset that has been used to evaluate the image processing routines. Only 10 nuclear staining channel images were taken: 4 images for normalization/segmentation evaluation purposes and 6 images for correct segment assignment task. The later also included 6 images from an additional channel that shows morphology of the object boundary.



(a) Original grayscale image

(b) Contrast-stretched image using 5 % percentiles

Figure 2.8: Background noise present in microscopic images from Benchmark dataset 2

2.2.4 Benchmark dataset 3 - Artificial biological cells

This benchmark dataset is included in order to have an easier cell segmentation platform introduced in Benchmark dataset 2. In this dataset, no inherent microscopic noise is present, so only systematic noise is added uniformly throughout an image. Moreover, only a few object types are available. The aim is to have an easy benchmark to evaluate image segmentation for different object types resembling those present in Benchmark dataset 2.

This dataset includes artificially simulated human cells and details of its synthesis technique are given in [140, 141, 142]. It has been used in [135] and here a modification of that is used as a benchmark. Using this dataset, different cell states can be artificially created resembling the human cell state undergoing cancer (i.e. including various cell death states) as shown in Fig. 2.9. For simplicity, a very lower number of cells with only two different cell states were selected. The noise could be also be artificially added using different techniques such as using salt and pepper, Gaussian, speckle and Poisson. With each noise type, 10 different images were created using stepwise increments in their respective control parameters. Similarly, image blurring can also be added stepwise using different filters such as Prewitt, Gaussian, disk, average, LoG, Laplacian, Sobel etc. The idea is to muddle the image in a stepwise fashion and obtain a gradual trend by the application of different algorithms. The images have 16-bit gray value resolution with available ground truth. The simulated image has a background intensity of 0 and maximal foreground object intensity 4.2 $\times$ 10^{-3}. This dataset was created using three different noising techniques in the following order:

- Gaussian
- speckle
- salt and pepper

A different combination and number of noising techniques could also be used. The aforementioned sequence and their respective parameters ensure that an adequate amount of noise was induced not only in the background but also on the foreground information. The selected noising techniques have following control parameters and their respective values (see Tab. 2.2):

The image series contained in this dataset has increasing noising level. The images created are shown in Fig. 2.10 and the artifact level *A*(n_1,n_2,n_3) is defined qualitatively between 0 and 1 by using linearly spaced intervals defined the number of images present in the dataset.

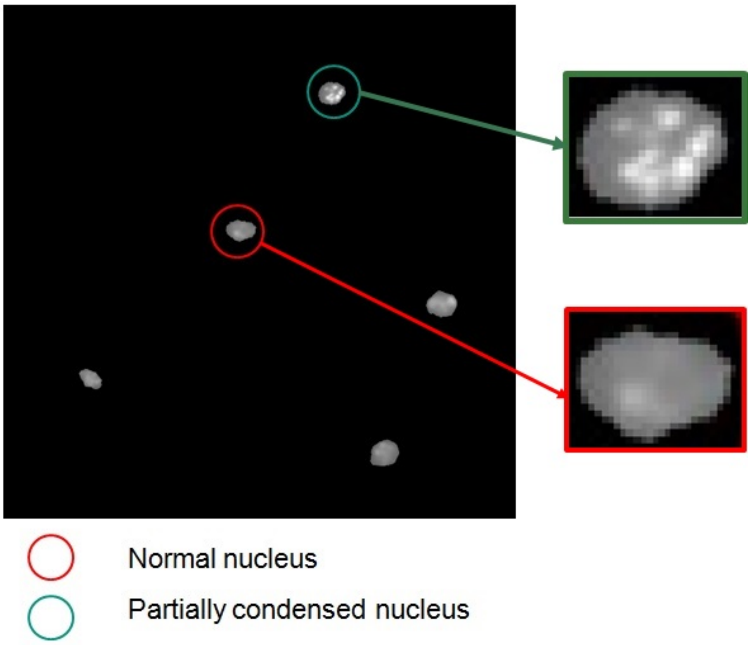


Figure 2.9: Artificially created benchmark image to match different cell states like that in Benchmark dataset 2 using a clear background. Two cell states were included in this dataset i.e. normal and partially condensed.

2.2.5 Benchmark dataset 4 - Human HT29 Colon Cancer

This dataset was published in Broad Bioimage Benchmark Collection available freely online[3]. For this dataset, only the ground truth of the total cell count was available. It contains microscopic images (showing cells) B_k where k = 1...6, shown in Fig. 3.8. The ground truth for B_k was only the average total number n_{ref} of cells based on two observers. For cell detection and counting, the benchmark has to be evaluated by:

$$\sigma_{GD} = \frac{||n_c - n_{ref}||}{n_{ref}}. \tag{2.7}$$

[3] ⋆http://www.broadinstitute.org/bbbc/

Technique used	**Control parameter**	**Minimum value**	**Maximum value**	**Interval used**
Gaussian	Variance with mean at 0 (n_1)	0.0002	0.002	0.0002
Speckle	Speckle variance (n_2)	0.02	0.2	0.02
Salt and pepper	Noise density (n_3)	0.05	0.5	0.05

Table 2.2: Noising techniques used with maximum, minimum and step sizes of their respective control parameters

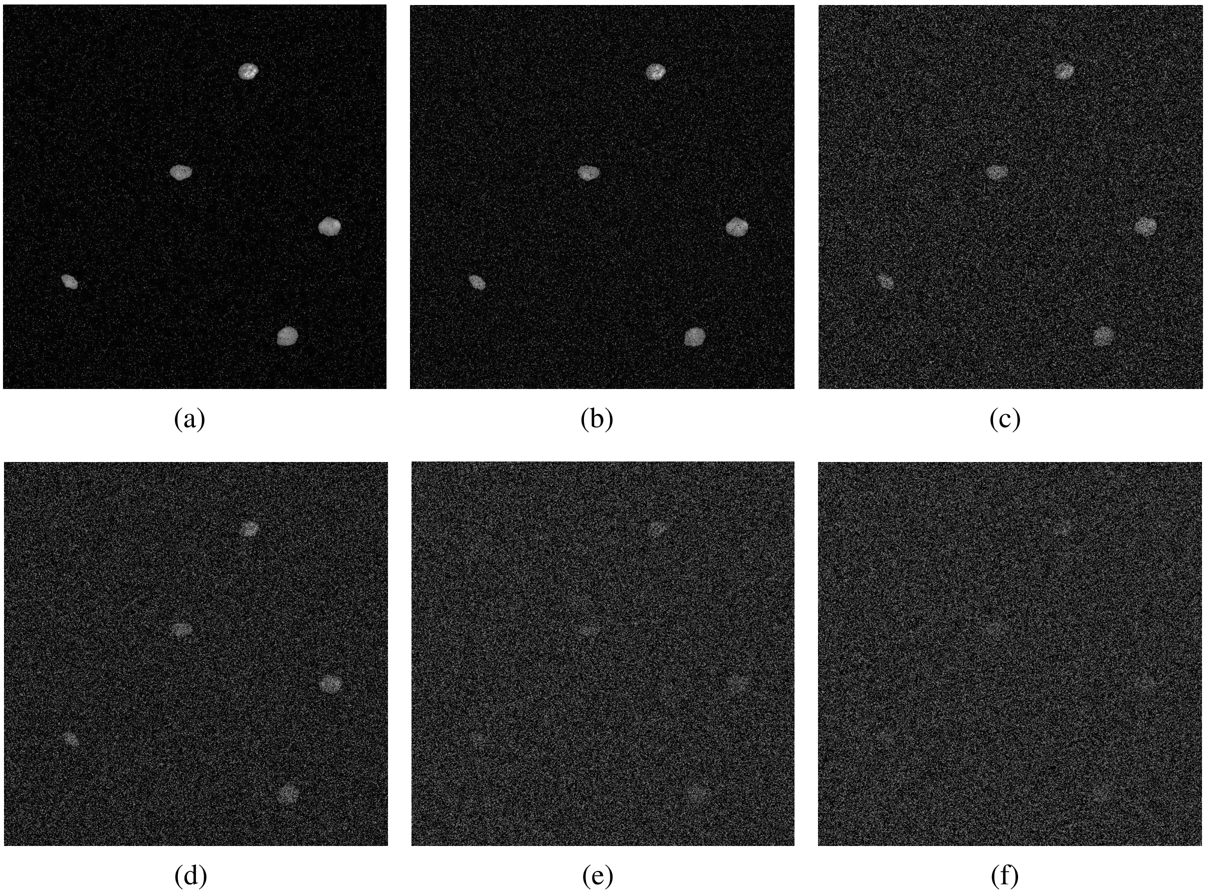


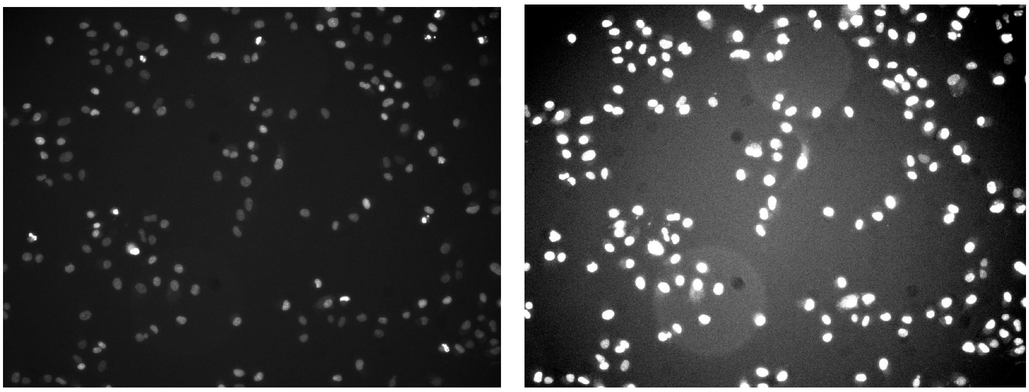
Figure 2.10: Benchmark Dataset 3 - images containing increasing noise level from Fig. 2.10(a) to Fig. 2.10(f)

2.3 Automatic feedback-based image normalization

2.3.1 Motivation

Since the real life image acquisition imparts different noise types to images, it is hard to extract the desired image objects information, especially when the noise is spatially varying (additive and multiplicative noise induced by acquisition through microscope etc.). Traditional methods are not sufficient to remove such a noise when the conditions are drifting from image to image. The aim here is to obtain noise-free image or at least a workable image with reduced background noise such that it is easier to segment the objects in the foreground. It is hereby proposed to remove background noise using an adaptive image normalization method. Feedback-based methods were used here to iteratively find optimal parameters that fit an estimated background and then subtract it from the real image background.

Since biological images acquired using a confocal microscope possess such a noise and currently quite a challenge for image analysis, Benchmark dataset 2 containing living and dying cells for the evaluation of the proposed normalization technique was chosen. The shading effect, in an image having numerous cells in different states, is illustrated in Fig. 2.11(a). The image suffers from an inconsistent background illumination since the background in corners is darker than in the middle of the image as shown in Fig. 2.11(b) using high contrast by applying 5 $\%$ percentiles following (4.2) with $I_{low,a}$ and $I_{high,b}$ equal to 5 and 95 respectively. This was done by cutting-off intensity values of 5$\%$ cells at lower and higher side of intensity histogram. Consequently, very dark and very bright pixels were ignored and an image with increased background intensity was obtained. Therefore, a shading correction is imperative in order to disentangle images from such variations concealing the original intensity information in an image.



(a) Original grayscale image

(b) Contrast-stretched image using 5 $\%$ percentiles

Figure 2.11: Shading effects present in microscopic images from Benchmark dataset 2

The main idea here is to formalize the knowledge about the desired objects in images using fuzzy sets. Image normalization is proposed to be done assuming a model for shading compensation. Parameters involved in shading compensation are tuned to find an optimal parameter set. To choose the optimal parameters, normalization evaluation measures are also described as fuzzy sets. The final shading compensation is done with a model that fulfills these evaluation criteria in the best manner. Using a normalized image, image segmentation is performed and the results are expected to be improved due to the removal of shading. The outcome is in terms of object delineation and boundaries giving an overall object count.

To quantify the outcome, fuzzy evaluation criteria were developed and an inference machine was built to obtain a scalar output fed back to manipulate the normalization routine to obtain an optimal parameter set. The employed scheme is introduced in Section 2.3.2 using some motivating examples from Benchmark dataset 2 and results are presented in Section 3.3.1. Since such a noise is only present in Benchmark dataset 2, this scheme only accounts for additive microscopic noise. For other noise types such as Gaussian noise, salt and pepper and speckle noise, another benchmark was introduced in Section 2.2.4. In both benchmarks 2 and 3, *A priori* knowledge is given in terms of number and size of objects to be found. In benchmark dataset3, also a class type of objects are given as ground truth.

2.3.2 Employed scheme

A surface fitting *retrospective* shading correction method with parametric fitting to a surface described by a parabolic polynomial was employed. The relation between an acquired image $I_A(x,y)$ and the true shading-free image $I_{SF}(x,y)$ is then described by:

$$I_A(x,y) = I_{SF}(x,y) \cdot S_m(x,y) + S_a(x,y) \tag{2.8}$$

where, $S_a(x,y)$ and $S_m(x,y)$ denote additive and multiplicative shading components respectively as a function of spatial pixel locations in $I_A(x,y)$. The components $S_m(x,y)$ and $S_a(x,y)$ account for the global contrast and brightness adjustment respectively in (2.8). In case of fluorescence microscopy, the acquired images are always distorted with additive shading noise. For the sake of simplicity, only $S_a(x,y)$ was considered, accounting only for the brightness correction in the normalization procedure, such that:

$$I_A(x,y) = I_{SF}(x,y) + S_a(x,y). \tag{2.9}$$

The proposed scheme for the feedback-based normalization technique is shown in Fig. 2.12. It includes the normalization of an input grayscale image by fitting the pixel values of $I_A(x,y)$ to $S_a(x,y)$. Image normalization is performed for different parameter combinations. Normalization results are evaluated for internal consistency of normalization procedure iteratively to adopt an optimal parameter set $\hat{\mathbf{a}}_{opt}$. Automatic segmentation based on normalized image using $\hat{\mathbf{a}}_{opt}$ is also done iteratively to adopt optimal segmentation parameter set $\mathbf{p}_{opt}$. Optimal segmentation is performed based on this $\mathbf{p}_{opt}$ obtained from optimally normalized image using $\hat{\mathbf{a}}_{opt}$.

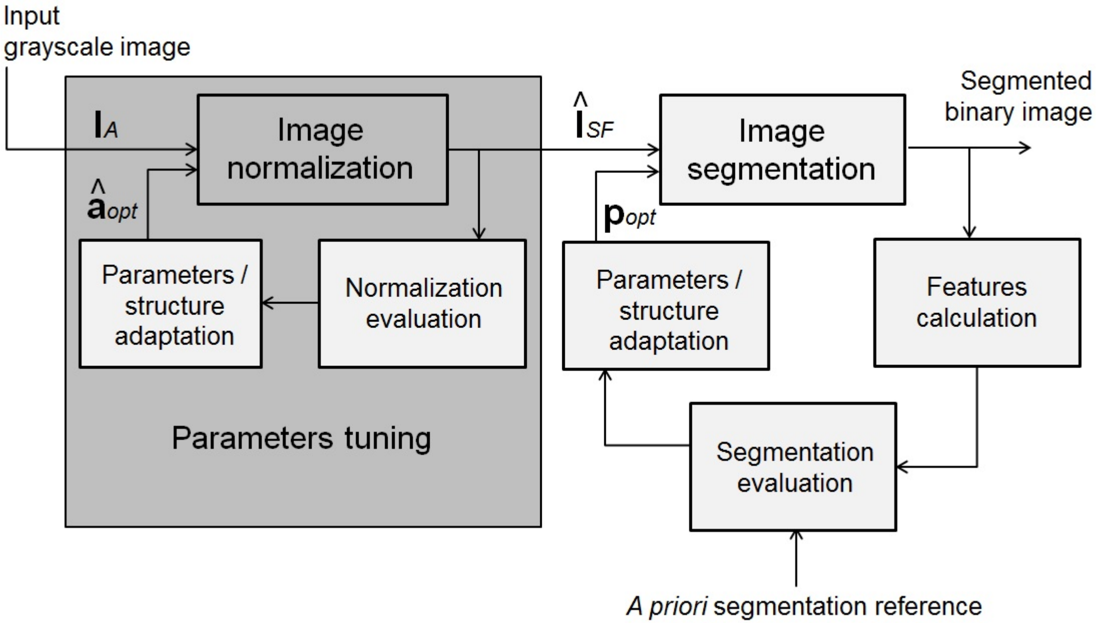


Figure 2.12: Employed feedback-based automatic normalization segmentation scheme

With reference to very large amount of shading seen in Fig. 2.11 of benchmark dataset, the aim to introduce automatic normalization was to improve the automatic feed-back based segmentation results. *A priori* knowledge included area, intensity and roundness factor (i.e. ratio of major cell axis to sum of major and minor axes) of normal cells in addition to the total number of cell count within an image. The *A priori* segmentation reference was given based on manual labeling of the cells in image dataset discussed in Section 3.3.2.

However, this is not the only noise present in images acquired with real equipments. Other types of noise like Gaussian noise, speckle, salt and pepper etc. as shown in Benchmark dataset 3 are also commonly observed. Nevertheless, such noise types could be removed using different 2-D order-statistic filtering such as median, mean, averaging, minimum filter etc. It could also be removed using morphological operations such as image opening with a structuring element having a size comparable to the objects that have to be found in an image. The employed parameter tuning scheme is given later in this Section and results are presented later in Section 3.4.3 by applying it to Benchmark dataset 3.

Image normalization for shading correction: As an example, a free exponent polynomial was used for the proposed feedback-based normalization routine. Since the intent here is just to estimate the shading-free image denoted as $\hat{\mathbf{I}}_{SF}$, the employed polynomial function denoted as $\hat{S}_a(x,y)$ used for estimation is given as:

$$\hat{S}_a(x,y) = \hat{a}_0 + \hat{a}_1 x + \hat{a}_2 y + \hat{a}_3 xy + \hat{a}_4 x^{\hat{a}_6} + \hat{a}_5 y^{\hat{a}_7}. \tag{2.10}$$

In order to induce better clarity in mathematical representation, the polynomial function and the acquired image denoted in matrix form as $\hat{\mathbf{S}}_a$ and $\mathbf{I}_A$ respectively, can be written in vectorized form as:

$$\hat{\mathbf{s}}_a^* = \mathbf{X} \cdot \hat{\mathbf{a}}, \tag{2.11}$$

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1^{\hat{a}_6} & 1^{\hat{a}_7} \\ 1 & 2 & 1 & 2 & 2^{\hat{a}_6} & 1^{\hat{a}_7} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 1 & m & 1 & m & m^{\hat{a}_6} & 1^{\hat{a}_7} \\ 1 & 1 & 2 & 2 & 1^{\hat{a}_6} & 2^{\hat{a}_7} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 1 & m & g & mg & m^{\hat{a}_6} & g^{\hat{a}_7} \end{bmatrix}, \ \hat{\mathbf{a}} = \begin{bmatrix} \hat{a}_0 \\ \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \\ \hat{a}_4 \\ \hat{a}_5 \end{bmatrix}, \ \mathbf{i}_A^* = \begin{bmatrix} I_A(x_1, y_1) \\ . \\ I_A(x_m, y_1) \\ I_A(x_1, y_2) \\ . \\ I_A(x_m, y_2) \\ . \\ . \\ I_A(x_m, y_g) \end{bmatrix}, \ \hat{\mathbf{a}}_{total} = \begin{bmatrix} \hat{a}_0 \\ \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \\ \hat{a}_4 \\ \hat{a}_5 \\ \hat{a}_6 \\ \hat{a}_7 \end{bmatrix}. \tag{2.12}$$

In (2.11), $\hat{\mathbf{s}}_a^*$ is the vectorized form of $\hat{\mathbf{S}}_a$. In (2.12), *m* and *g* refer to the number of rows and columns of $\mathbf{I}_A$ respectively. In (2.11), the dimensions of $\hat{\mathbf{s}}_a^*$ are *mg* × *1* whereas the dimensions of $\mathbf{X}$ and $\hat{\mathbf{a}}$ are *mg* × *6* and *6* × *1* respectively. The vector $\mathbf{i}_A^*$ in (2.12) also has the dimensions of *mg* × *1*. Since the analytical solution of the problem including $\hat{a}_6$ and $\hat{a}_7$ is not possible, numerical optimization in order to achieve parameters for best fit of $I_A(x, y)$ to $\hat{S}_a(x, y)$ was performed using analytical least squares minimization for $\hat{\mathbf{a}}$. The normalization routine was divided into two steps:

Step 1 - Initial estimate: Firstly, the least square regression problem is to find:

$$Q_{S1} = \min_{\hat{\mathbf{a}}} \frac{1}{2} ||\mathbf{X} \cdot \hat{\mathbf{a}} - \mathbf{i}_A^*||^2 \tag{2.13}$$

such that the optimization initially is only performed fixing the exponents $\hat{a}_6$ and $\hat{a}_7$ and finding the optimal parameter values with respect to $\hat{\mathbf{a}}$ to yield an initial estimate of the search space used in the succeeding step of proposed feedback-based normalization routine.

Step 2 - Search space exploration: The feedback-based normalization is then continued by varying $\hat{a}_6$ and $\hat{a}_7$ around initial estimates to find the optimal combination of $\hat{\mathbf{a}}$, $\hat{a}_6$ and $\hat{a}_7$. The least squares problem for automatic tuning of normalization routine was to find:

$$Q_{S2} = \min_{\hat{a_6}, \hat{a_7}} \frac{1}{2} ||Q_{S1}||^2 . \tag{2.14}$$

The calculation of criterion (2.14) was based on least square minimization in order to fit $\hat{S}_a(x, y)$ to pixel values of $I_A(x, y)$. The pseudo inverse of the system matrix $\mathbf{X}$ was used to calculate $\hat{\mathbf{a}}$ for given values of $\hat{a}_6$ and $\hat{a}_7$ as:

$$\hat{\mathbf{a}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{i}_A^* \tag{2.15}$$

to find an optimal solution for the inner criterion Q_{S1}. Consequently, $\hat{\mathbf{I}}_{SF}$ is calculated using its vectorized form $\hat{\mathbf{i}}_{SF}$ as:

$$\hat{\mathbf{i}}_{SF} = \mathbf{i}_A^* - \mathbf{X} \cdot \hat{\mathbf{a}} \tag{2.16}$$

where the dimensions of $\hat{\mathbf{i}}^*_{SF}$ are same as that of both $\mathbf{i}^*_A$ and $\mathbf{X}\cdot\hat{\mathbf{a}}$. The range for $\hat{a}_6$ and $\hat{a}_7$ was kept closely around quadratic based on the initial estimate in order to find the best exponent for optimal image normalization. Increasing the exponential powers of $\hat{a}_6$ and $\hat{a}_7$ more than cubic does not improve the results obtained. However, higher exponential powers were not used in order to avoid the suppression of useful intensity information about the objects to be found in the image. The employed algorithm also has the possibility to enhance its running time efficiency. This can be done by downscaling the resolution of $\mathbf{I}_A$ by a certain factor f such that newer $\mathbf{i}^*_A$ has lesser overall elements by a factor f. This reduces the number of calculations and consequently speeds up the automatic normalization routine.

Normalization evaluation: Generally, a normalization outcome is evaluated with respect to two different criteria i.e. check of internal consistency of spatial image intensities and improvement of image post-processing results based on the normalized image. A variety of metrics can be used as a quality measure for the normalization procedure with respect to given *a priori* knowledge. In this thesis, one criterion for the residual root mean square error $Q_{norm,1}$ of an acquired image $\mathbf{I}_A$ to the fitted function $\hat{S}_a(x,y)$ normalized by the range of given image intensities was used and is defined as:

$$Q_{norm,1} = \frac{\sqrt{\frac{1}{m \times g}\sum_{i=1}^{m}\sum_{j=1}^{g}(I_A(x_i,y_j) - \hat{S}_a(x_i,y_j))^2}}{|\max\{\mathbf{I}_A\} - \min\{\mathbf{I}_A\}|}. \tag{2.17}$$

The idea behind using criterion (2.17) is to calculate the deviation (i.e. residual shading after fitting $\mathbf{I}_A$ to $\hat{S}_a(x,y)$) of $\mathbf{I}_A$ from a uniform background which is intended to be obtained in normalized $\hat{\mathbf{I}}_{SF}$, given that the image has shading noise of the form given in Eq. (2.40). However, background estimation is not done here, instead fitting is done based on both back- and foreground intensity information. This can be dealt with by using a sliding filter to inhibit the effect of high brightness spatially in any image. Nevertheless, an ideally normalized image would consist of nearly a uniform background just containing the information about segments (e.g. cells etc.) to be found. This would be indicated by smaller values (i.e. close to zero) of criterion (2.17).

Another criterion $Q_{norm,2}$, given in Eq. (2.18), based on the normalized sum of absolute difference in median image intensity values i proposed. It is calculated over rows and columns of the estimated shading-corrected image $\hat{\mathbf{I}}_{SF}$ by applying q % percentiles on pixel median values. It is used to evaluate the decrease in spatial intensity value differences of rows and columns as corrected by the shading correction algorithm.

In ideal case, for a fairly good segment distribution and shading trends having middle section brighter than corners with no segment having area greater than 50 % of the pixels, such a criterion should yield values closer to 0. The percentile operator using q is denoted as pr_q $(\cdot)$ for lower *qth* percentile and $pr_{100\text{-}q}$ $(\cdot)$ for the upper *(100-q) th* percentile in Eq. (2.18).

$$Q_{norm,2} = \frac{|pr_{100-q}(\mathbf{x}_{med}) - pr_q(\mathbf{x}_{med})| + |pr_{100-q}(\mathbf{y}_{med}) - pr_q(\mathbf{y}_{med})|}{2 \times \max\{\,|\,t\,|\,,1\}}, \tag{2.18}$$

$$t = \max\{pr_{100-q}(\mathbf{x}_{med}), pr_{100-q}(\mathbf{y}_{med})\} - \min\{pr_q(\mathbf{x}_{med}), pr_q(\mathbf{y}_{med})\}, \tag{2.19}$$

$$\mathbf{x}_{med} = \begin{bmatrix} x_{med,1} \\ \cdot \\ \cdot \\ x_{med,m} \end{bmatrix}, \quad \mathbf{y}_{med} = \begin{bmatrix} y_{med,1} \\ \cdot \\ \cdot \\ y_{med,g} \end{bmatrix}, \tag{2.20}$$

$$x_{med,i} = \underset{j=1...g}{\mathrm{med}} \{\hat{I}_{SF}(x_i, y_j)\} \quad \forall \quad i = 1...m, \tag{2.21}$$

$$y_{med,j} = \underset{i=1...m}{\mathrm{med}} \{\hat{I}_{SF}(x_i, y_j)\} \quad \forall \quad j = 1...g, \tag{2.22}$$

where, $\mathrm{med}\ \{\cdot\}$, $\max\ \{\cdot\}$ and $\min\ \{\cdot\}$ in Eq. (2.18), (2.19), (2.20), (2.21) and (2.22) represent median, maximum and minimum operators respectively. In criterion (2.18), the 2 in the denominator is used to average the effect of variations in percentile image intensity values along columns and rows.

The criterion (2.18) will ideally yield values closer to 0 in case of minimum spatial intensity variations along the row and columns, which would indicate median values along all row and columns are closer to each other. This would tend to eliminate the high spatial variations in intensity, imparting a uniform background which is the desired goal in shading correction. This could be illustrated from a practical example. For instance, if one object in the foreground is needed to be segmented in the presence of background shading. The image with high shading and the image with a more uniform background (after the shading correction) are shown in Fig. 2.13.

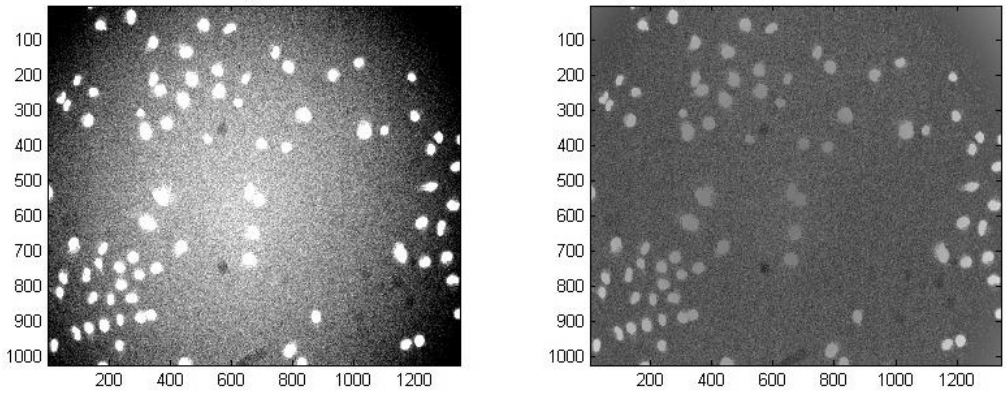


Figure 2.13: Expected output of the shading correction algorithm in a cell image with high background noise. Left: original image of cell image with high background noise. Right: image obtained after applying shading correction algorithm.

A fuzzy formulation of the error measures described in Eq. (2.17) and (2.18) to grasp the parametric effect in a more intuitive way is proposed. Spline-based (a.k.a *z-shaped*) fuzzy membership functions were employed with two parameters i.e. α and β defining the maximum and minimum x-values of criteria respectively. Fuzzy memberships for the criteria (2.17) and (2.18) are denoted as μ_1 and μ_2 respectively. It is reasonable to calculate a criterion $Q_{fuzz,norm}$ based on a product of μ_1 and μ_2 since fulfillment of both criteria (2.17) and (2.18) for each evaluation is essential and complete absence of any (i.e. μ_1 = 0, μ_2 = 0) should render $Q_{fuzz,norm}$ zero. Therefore, a criterion:

$$Q_{fuzz,norm}(\hat{\mathbf{a}}_{total}) = \mu_1(\hat{\mathbf{a}}_{total}) \cdot \mu_2(\hat{\mathbf{a}}_{total}) \qquad (2.23)$$

based on aforementioned logic is introduced to express the internal consistency of automatic image normalization.

Parameters/Structure adaptation: The criterion (2.23) needs to be maximized in order to obtain

$$\hat{\mathbf{a}}_{opt,fuzzy,norm} = \arg\max_{\hat{\mathbf{a}}_{total}} Q_{fuzz,norm}(\hat{\mathbf{a}}_{total}). \qquad (2.24)$$

In this thesis, $\hat{\mathbf{a}}_{opt,fuzzy,norm}$ was computed based on exhaustive enumeration. However, more sophisticated optimum search methods such as genetic algorithms, constraint optimization etc. could be used as well.

Image normalization for other noise types: Typical noise types present in images such as Gaussian noise and salt and pepper noise should be adequately removed before processing an image further. Any type of filter according to knowledge of noise present in image can be used. A scheme to adapt the parameters of such denoising filters are presented here. For instance, with the presence of salt and pepper and speckle noise, a median filter with a free parameter of symmetric filtering window size $w \times w$ is used as 2-D order-statistic filter. Then an image opening with disk of size s as a structuring element is applied to the resulting image. However, s is fixed and w is tuned iteratively for the sake of simplicity and for the proof of principle. Therefore, the resulting parameter set is: $\mathbf{p} = w$. For evaluation of normalization outcome, quality measures are provided in terms of:

- pixels of background noise detected in normalized image with respect to the background of ground truth image (q_1) according to Eq. (2.36)
- pixels of foreground objects detected in normalized image with respect to the objects of ground truth image (q_2) according to Eq. (2.38)

q_1 and q_2 are converted to fuzzy membership functions μ_1 and μ_2 using Eq. (2.37) and (2.39). The overall evaluation criterion is given as:

$$Q_{norm} = 1 - (1 - \mu_1)(1 - \mu_2). \qquad (2.25)$$

and the optimal parameter set $\mathbf{p}$ is adapted that maximizes the criterion (2.25)

$$\mathbf{p}_{opt} = \arg\max_{\mathbf{p}} Q_{norm}(\mathbf{p}). \qquad (2.26)$$

Normalized image $\mathbf{I}_{norm}$ is obtained based on $\mathbf{p}_{opt}$. The result of this proposed strategy when applied on Benchmark dataset 3 is presented in Section 3.4.3.

2.4 Automatic feedback-based image segmentation

2.4.1 Motivation

Image segmentation is the pivotal part in any image analysis. There are many image segmentation algorithms around that yield favorable results. Problems arise when the dataset is challenging with drifting image conditions. In such cases, the parameters of image segmentation should be adjusted automatically in an iterative manner for improvement of segmentation results based on *a priori* knowledge. There has been an adequate work on feedback-based automatic image segmentation techniques such as [7, 10, 38, 51, 114, 132]. However, these techniques are limited in terms of well-formulated reference knowledge about object characteristics and types. Therefore, a new method for an automatic feedback-driven segmentation for tuning processing parameters using fuzzy *a priori* knowledge is proposed. For quantifying the outcome, a fuzzy evaluation criterion was introduced and an inference machine was built for the segmentation routine to obtain the optimal parameter set.

The situation is even complex when the amount of information is huge and comes from more than one channel. Therefore, to do segmentation in more than one channel and aggregate their results is not trivial. In complex cases where the objects to be detected are touching each other or are overlapping, an image segmentation algorithm should be robust enough. In biological datasets, the scenario is quite like this and image processing is quite challenging. In order to evaluate techniques that perform these complex tasks, evaluation examples should be there.

In Benchmark dataset 2, images containing living and dying cells from three different channels are given for the evaluation of the proposed segmentation technique. This dataset is particularly challenging in terms of correct object delineation because correct segmentation will enable the targeted features extraction that will enable the right class annotation. Different cell areas and structures against a noisy background, therefore, are hard to segment correctly posing appropriate challenges to an image segmentation routine. Numerous class types and hard-to-segment cases like overlapping cells (see Fig. 2.14) and cell fragments belonging to each other (see Fig. 2.15) make this even harder task to perform. Image segmentation routines are proposed to perform the task of segmentation in these different settings. The scheme used for correct fragments of same nuclei is introduced in Section 2.4.3 using some motivation examples from Benchmark dataset 2 and results will be provided in Section 4.7.

Additionally, Benchmark dataset 3 was also used to validate the image segmentation routines when systematic noise addition like Gaussian, salt and pepper etc. noise were added to very few object types. It was used to make this demonstration more understandable since Benchmark dataset 3 is less complex and does not contain inherent image acquisition noise.

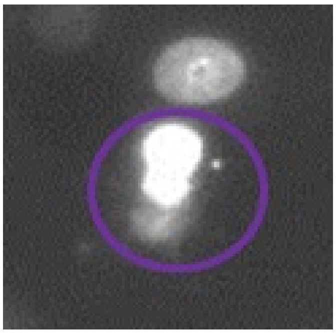
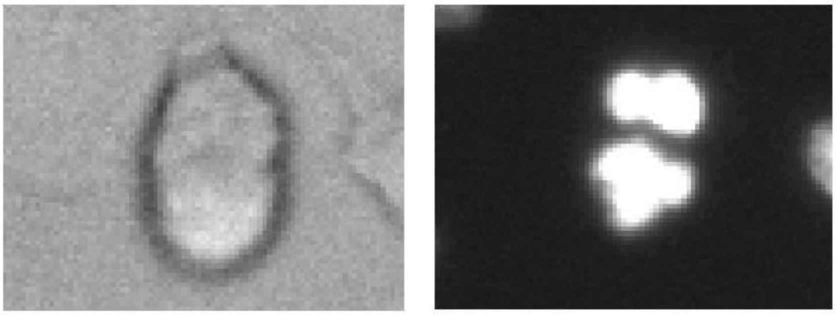


Figure 2.14: Benchmark dataset 2: Overlapping of two cells.



(a) Bright field channel image showing cellular structure (b) Nuclear staining channel showing cellular fragmentation

Figure 2.15: Fragments belonging to same objects from Benchmark dataset 2 using multichannel information

2.4.2 Employed scheme for automatic feedback-based image segmentation

The proposed scheme for feedback-based automatic image segmentation is shown in Fig. 2.16. It includes segmentation of a grayscale image by transforming it into a binary image containing so called binary large objects (BLOBs), which are the segments found by an image segmentation technique. An image pre-processing step such as image filtering using convolution etc. is included before performing binary image segmentation. The image segmentation is performed using different parameter combinations and the desired features of BLOBs are calculated. Segmentation evaluation with respect to these features is performed accordingly using given *a priori* reference features in an iterative fashion. For a user, *a priori* knowledge can be derived from number and types of objects to be found in an image. This can be done using simple counting operation manually and labeling the object types provided by experts. The size of the objects could be also included in similar fashion by counting the number of pixels of smallest and biggest objects to be found in the dataset. An optimal parameter set is adopted based on a quality criterion and optimum segmentation is performed using this optimal parameter set.

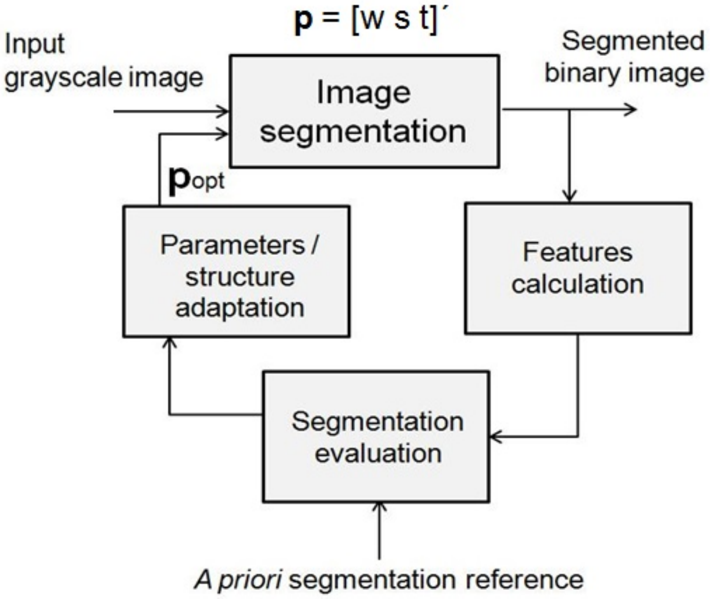


Figure 2.16: Employed feedback-based automatic image segmentation scheme

Image segmentation: To perform adaptation of parameters involved in image segmentation, a generic pipeline is required to be designed. For a user, any kind of pipeline structure and any number of processing steps can be involved. However, parameters involved in image segmentation should be carefully selected in order to affect the segmentation outcome positively. These parameters are consequently adapted to yield the optimal parameter set based on a certain quality criteria.

As an example, a basic sequential image processing routine consisting of a convolution filter, a thresholding and an opening routine followed by an image filling (see [48]) were used. The convolution is done using a symmetric $w \times w$ matrix having elements equal to $\frac{1}{w^2}$, a threshold value t is set and the opening routine uses a disc of size s as a structuring element. Therefore, the image segmentation depends upon the parameter vector $\mathbf{p} = (w, s, t)^T$.

Features calculation: Object features are needed to formalize *a priori* knowledge about the objects to be found in an image. Later, in the case of Benchmark dataset 1 object features are needed again to be fed to the classifier in order to differentiate between different object types (see Section 3.2.1). These features could be different from ones used here. With respect to **p**, the objects delivered by the segmentation process not only differ in size, extent, etc. but in the underlying pixel values as well. The setting of **p**, however, is crucial for the segmentation process. To find optimal values for **p**, a criterion needs to be calculated based on the feature vector $\mathbf{f}_i^T$ = (f_{i1}, ..., f_{ih}), considering $j = 1...h$ number of features for each segment i where $i = 1...n_c$, and the total number of segments found denoted as n_o. These features may be related to geometry (area, sphericity, etc.), intensity distribution (brightness, noise, etc.), and/or the

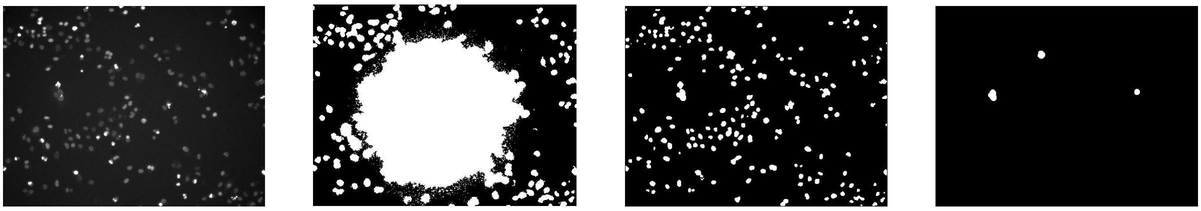


Figure 2.17: Segmentation results using manual selection of **p**, where Fig. 2.17(a) Original grayscale image Fig. 2.17(b) using $\mathbf{p} = (1, 1, 33700)^T$, Fig. 2.17(c) using $\mathbf{p} = (3, 3, 34000)^T$ and Fig. 2.17(d) using $\mathbf{p} = (5, 15, 34000)^T$ in Benchmark dataset 2

content (e.g. number of sub-fragments etc.) of each segment.

Segmentation evaluation: Generally, a segmentation outcome is evaluated with respect to two different criteria i.e. whole image and each single segment. A variety of metrics can be used as a quality measure of features with respect to given *a priori* knowledge. For Benchmark dataset 2, segmentation measures Q_{ed} and Q_{fuzz} defined by (2.41) and (2.42) given in Section 2.6.1 were used due to the presence of only abstract ground truth based on object features.

Parameters/Structure adaptation: The criterion given in (2.41) needs to be minimized with respect to the parameter vector **p** in order to obtain the optimal parameter set $\mathbf{p}_{opt,ed}$ as shown in (2.27). On the other hand, the criterion (2.42) needs to be maximized in order to obtain

$$\mathbf{p}_{opt,ed} = \arg\min_{\mathbf{p}} Q_{ed}(\mathbf{p}), \tag{2.27}$$

$$\mathbf{p}_{opt,fuzzy} = \arg\max_{\mathbf{p}} Q_{fuzz}(\mathbf{p}). \tag{2.28}$$

Here, $\mathbf{p}_{opt,fuzzy}$ was computed based on exhaustive enumeration. However, more sophisticated optimum search methods such as genetic algorithms, constraint optimization etc. could be used as well.

2.4.3 Challenging multichannel image segmentation tasks

One challenging image processing task is to correctly find the objects or segments in an image and group them properly if two or more segments belong to each other. Another challenge lies in fusing information from different image channels to employ it for correct segment assignment. As described in Section 2.4.1, object segmentation in multichannel images is quite challenging when it contains certain morphologies that are potentially detected separately while belonging to the same object. Therefore, a method to improve the segmentation process for fragmented segments using edge detector and morphological operators is proposed. It includes information from two channels namely Channel A (containing primary object information) and Channel B (containing supplementary information for objects reassignment) especially in cases where segments in Channel A cannot be infallibly assigned. The main idea is to use the segments in the Channel B in order to combine the segments of Channel A.

For performing this task and evaluating the results, Benchmark dataset 2 was used (6 images for each channel).

Employed methods: The image processing pipeline implemented for this task is shown in Fig. 2.18. Channel B segment information is extracted by applying a Sobel edge detector to the Channel B image. The extracted segments are further processed sequentially using morphological operators i.e. image closing, hole-filling and image opening respectively (Fig. 2.19).

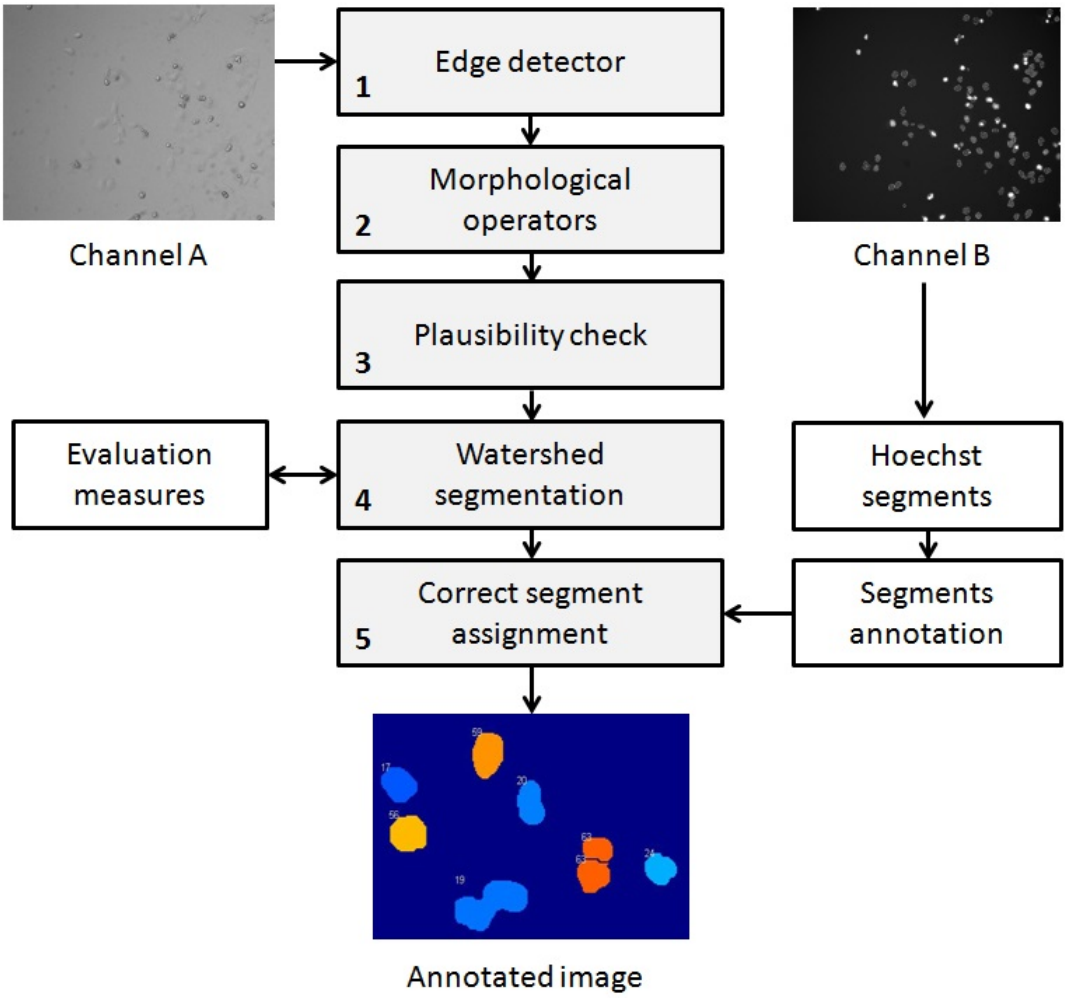


Figure 2.18: Complete image processing pipeline for assignment of Channel A segments to their Channel B counterparts. Images shown here for Channel A and B are taken from Benchmark dataset 2

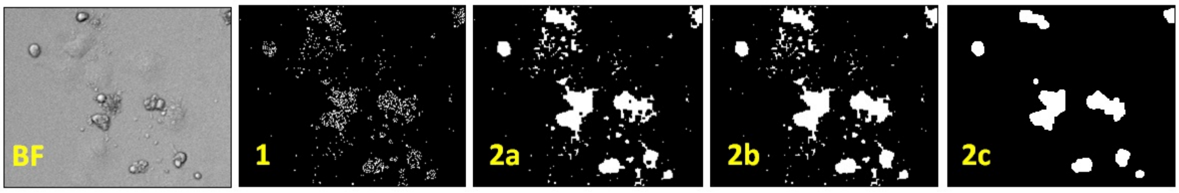


Figure 2.19: Channel B segmentation (steps 1-2 from Fig. 2.18): Channel B (section of original Channel B image taken from Benchmark dataset 2), 1 (edge detector), 2a (image closing), 2b (hole filling) and 2c (image opening).

In the next step, binary large objects (BLOBs) detected by the above mentioned operations are further checked for plausibility. Criteria based on irregular size i.e. too small/big segments are implemented, whereby removing small objects and further segmenting big objects. A watershed analysis seeking for gradients in brightness is known to perform well in cell separation [9, 111]. However, this separation cannot be achieved with desired results due to inconsistencies caused by cellular fragmentation in both channels. Therefore, the distance map of each BLOB was used. As a result all found BLOBs are within a given area-range and show convex properties (see Fig. 2.20). Channel A segments are then annotated for reassignment. A correct reassignment of the segments is done based on results obtained after step 4 in the pipeline shown in Fig. 2.18.

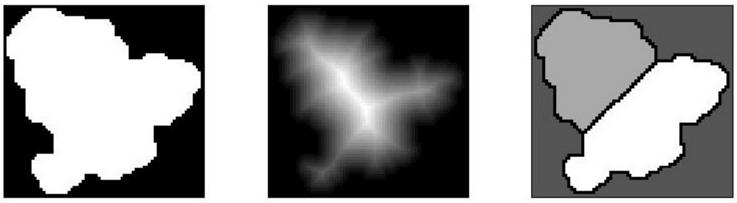


Figure 2.20: Watershed segmentation based on distance map: from left to right: BLOB, distance map and separated objects. Algorithm was applied on Channel B image from Benchmark dataset 2

For DAPI segmentation, a segmentation routine described in [73, 72] is applied. As a result, $i = 1, ..., n_b$ segments in the Channel B and $j = 1, ..., n_d$ segments in the Channel A are obtained. The pixel positions (x, y) belonging to the segments are gathered in the sets $\mathcal{B}_i$ for Channel B segments and $\mathcal{D}_j$ for Channel A segments. A set $\mathcal{D}$ defining all found segments in Channel A is defined as:

$$\mathcal{D} = \{\mathcal{D}_1, \dots, \mathcal{D}_{n_d}\}. \tag{2.29}$$

where as, a set $\mathcal{B}$ defining all found segments in Channel B is defined as:

$$\mathcal{B} = \{\mathcal{B}_1, \ldots, \mathcal{B}_{n_b}\}. \tag{2.30}$$

If Channel A segments belong to the same Channel B segment, they are annotated the same class[4]. Therefore, the portion of positive Channel B segment pixels in a Channel A segment j is calculated and the most affecting Channel B segment is denoted as n_j (see (2.31) and (2.32)).

$$p_j = \max_i \frac{\mathrm{card}(\mathcal{D}_j \bigcap \mathcal{B}_i)}{\mathrm{card}\mathcal{D}_j}, \quad n_j = \arg\max_i \frac{\mathrm{card}(\mathcal{D}_j \bigcap \mathcal{B}_i)}{\mathrm{card}\mathcal{D}_j}. \tag{2.31}$$

If $p_j \geq 0.1$, segment $\mathcal{D}_j$ is removed from the Channel A segments and transferred to a set AposB segments $\mathcal{DB}$. The set AposB contains segments in Channel A which are also segmented in the Channel B. All segment sets being affected by the same Channel B segment are merged:

$$\mathcal{D} = \{\mathcal{D}_j | p_j < 0.1\}, \tag{2.32}$$

$$\mathcal{DB} = \{\mathcal{DB}_1, ..., \mathcal{DB}_n\} \tag{2.33}$$

with

$$DB_i = \bigcup_{\substack{n_j = i \\ \mathcal{D}_j >= 0.1}} \mathcal{D}_j. \tag{2.34}$$

Empty sets in $\mathcal{DB}$ are removed. All subsets in $\mathcal{D}$ denote BF-negative segments, all subsets in $\mathcal{DB}$ denote Channel B-positive segments. In this way, segment reassignment can be done (see step 5 in Fig. 2.18). The results for the application of this scheme using Benchmark dataset 2 are presented in Section 4.7 as an application-specific extension.

2.5 Automatic feedback-based image object classification

In reference to Fig. 2.1, the concept of image object recognition and consequent identification of objects found therein, image object classification is necessary. The techniques for performing this operation are already described in Section 1.2.3. After image segmentation, the main task is to select all useful object features that would enable us to perform an automatic image object classification task.

[4] If more than one Channel B segment affects the Channel A segment, the most affecting Channel B segment will be chosen.

2.5.1 Possible tuning strategies

There are two possible tuning strategies (see Fig. 2.21):

1. Object classification can be internally tuned using a particular classifier by feeding it with a different number and types of input features (see black feedback line between two yellow blocks in Fig. 2.21).
2. Object classification can help in external tuning of image normalization and/or image segmentation using classification results from a particular classifier as feedback to the pipeline. See green feedback line to image segmentation block for tuning automatic image segmentation in Fig. 2.21

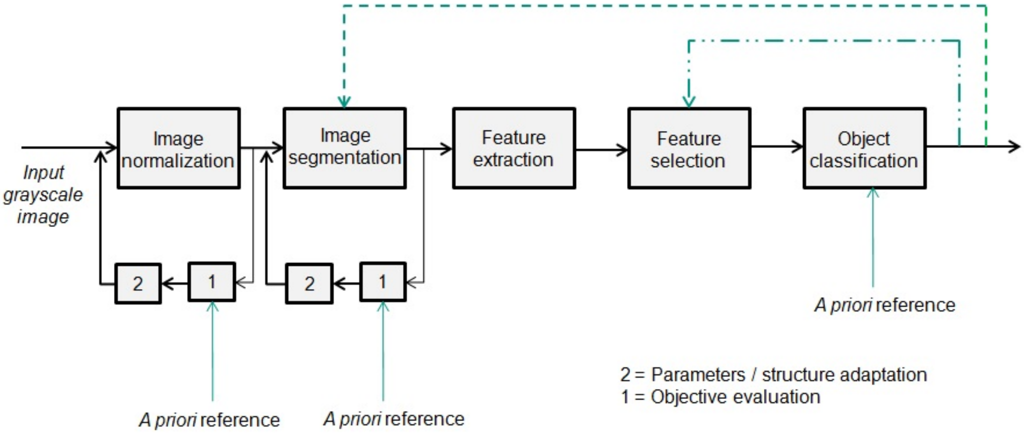


Figure 2.21: Feedback automatic image object classification scheme

However, in order to perform automatic image classification, evaluation measures are required.

2.6 Quality measures

2.6.1 Segmentation measures

In this section, the aim is to provide some standard segmentation measures. The reason behind this to give an idea about how to design and select evaluation measures based on the desired segmentation/classification goals. These measures could be specific for some datasets but could be adapted for more general use. Standard image segmentation algorithms such as Otsu thresholding, edge detection, clustering etc. are quite useful for segmenting identifiable objects against a static non-complex background. Since the given reference images denote the ground truth, supervised evaluation methods can be used. The supervised segmentation evaluation measures are based on the degree of similarity between image segmentation resulting from an application of a particular algorithm against a manually-segmented reference image. A variety of such measures exist and are mostly based on the number of pixels that are misclassified when compared to the image object pixels in the manually labeled image while penalizing pixels inversely proportional to distance from correct ones [152, 80]. Many methods (i.e. [151, 90] etc.) now use the difference in the number of objects detected with the ground truth. Other methods based on object features extraction etc. [158], edge-based image segmentation evaluation etc. [127], using Pareto front [37] also exist.

Measures like Rand Index (RI), Jaccard Index (JI), Normalized Sum of Distances (NSD) and Hausdorff Metric (HM) are quite useful when evaluating 2D image segmentation [139, 24]. However, they more or less tend to describe the same criteria described above. In RI based on true positive and negative image pixels, the image must contain useful information in its negative pixels too. In case under consideration, the background needs not to be segmented, so development of a criterion based on positive pixel values is required since there are foreground objects as ground truth against a background that contains no useful information. In such cases, RI tend to deliver high values even if no foreground object is segmented because background constitutes most of the pixels in images of the dataset. Moreover, false detections are to be penalized in such criterion.

Here, the segmentation measures are presented in three distinct manners provided that ground truth for these measures are given:

1. Based on total object count: Let *S* be the set of all the BLOBs (Binary Large Objects) found in $\mathbf{I}_{\text{seg}}$ such that, $\mathcal{S} = \{\mathcal{S}_u | u = 1, ..., U\}$ and U is the total number of BLOBs found in $\mathbf{I}_{\text{seg}}$. $\mathcal{S}_u$ is a set of C_u pixel values and is defined as: $\mathcal{S}_u = \{\{S_{ux,1}, S_{uy,1}\}, \cdots, \{S_{ux,C_u}, S_{uy,C_u}\}\}$. Here, *x* and *y* are representing image coordinates and *u* represents a given BLOB index. Altogether, $S_{ux,1}$ and $S_{uy,1}$ represent the row and column pixel indices of a given BLOB $\mathbf{S}_u$.

Similarly, let $\mathcal{T}$ be the set of all the BLOBs present in $\mathbf{I}_{\text{truth}}$ such that, $\mathcal{T} = \{\mathcal{T}_v | v = 1, ..., V\}$ and V is the total number of BLOBs present in $\mathbf{I}_{\text{truth}}$ and is given. Here, $\mathcal{T}_v$ is defined as $\mathcal{T}_v = \{\{T_{vx,1}, T_{vy,1}\}, \cdots, \{T_{vx,C_v}, T_{vy,C_v}\}\}$ where, C_v denotes the number of pixels in BLOB T_v.

For each BLOB in $\mathcal{S}_u$ the overlap with all ground truth BLOBs $\mathcal{T}_v$ is determined. If the maximum overlap exceeds $1/3$ of $\mathcal{T}_v$, assuming that the BLOB $\mathcal{S}_u$ being present in the ground truth as well, it is added to the set of correct segmented objects $\mathcal{S}_c$ and deleted from the corresponding overlapping ground truth element from the complete set of *V* ground truth elements such that for a new $\mathcal{S}_u$, *z* = 1, ... , *V* - n_v (number of $\mathcal{T}_v$ BLOBs affected by overlap) and *z* is new number of remaining T_v elements. The formula for $\mathcal{S}_c$ in each iteration is given as:

$$\mathcal{S}_c = \{\mathcal{S}_u | \max(\text{card}(\mathcal{S}_u \cap \mathcal{T}_1), \ldots, \text{card}(\mathcal{S}_u \cap \mathcal{T}_z)) > \frac{1}{3}C_u\}. \tag{2.35}$$

Where, z indicates maximum number of elements left in $\mathcal{T}_z$ after overlap. Therefore, the criterion is described as:

$$q_1 = \frac{|\text{card}(\mathcal{T}) - \text{card}(\mathcal{S}_c)|}{\text{card}(\mathcal{T})} \tag{2.36}$$

The *a priori* knowledge for this measure is handled using fuzzy membership functions in order to describe it in the range of 0 - 1 according to (2.4) such that:

$$\mu_1 = 1 - \mu(q_1; 2, 0, 1). \tag{2.37}$$

From the logical point of view, such a criterion should engulf all the cases of true positive (TF), true negative (TN), false positive (FP) and false negative (FN) detections. Since negative pixels are not the part of required information, this criterion should be able to handle all possible case of positive pixels such as:

- one $\mathcal{S}_u$ overlaps one $\mathcal{T}_v$ i.e. TP (counted according to (2.4)).
- one $\mathcal{S}_u$ overlaps no $\mathcal{T}_v$ but just the background i.e. FP. Such cases are not counted as overlap according to (2.4).
- one $\mathcal{S}_u$ overlaps two $\mathcal{T}_v$ BLOBs. As a result of overlap, both BLOBs of $\mathcal{T}_v$ will be deleted in new set $\mathcal{T}_z$ to be checked for next iteration, so it will not increase the overall count. In this way, (2.4) would not include the match twice but only once.
- two $\mathcal{S}_u$ BLOBs against one $\mathcal{T}_v$. So, for the first $\mathcal{S}_u$, the overlap will delete the corresponding $\mathcal{T}_v$ and only this count will be taken into account. In next iteration for the second $\mathcal{S}_u$, no corresponding BLOB will be present in $\mathcal{T}_z$.

2. Based on pixel misclassification: It is same as RI and is described terms of segmented and ground truth pixels as:

$$q_2 = \frac{\sum_{i,j} |\text{sign}(\delta_{\text{truth},ij} - \delta_{\text{seg},ij})|}{\sum_{i,j} \text{sign}(\delta_{\text{truth},ij} + \delta_{\text{seg},ij})}. \tag{2.38}$$

The *a priori* knowledge for this measure is handled using fuzzy membership functions in order to describe it in the range of 0 - 1 according to (2.4) such that:

$$\mu_2 = 1 - \mu(q_2; 2, 0, 1). \tag{2.39}$$

Using these criteria, the overall segmentation quality measure $Q_{\text{seg}}(r, b, n)$ for each image is then given as:

$$Q_{\text{seg}}(r, b, n) = \mu_1 \cdot \mu_2. \tag{2.40}$$

This combination is valid when absolute ground truth about objects to be found in an image is given in terms of binary pixels. This could also work, if class labels are also provided in ground truth.

3. Based on abstract object features: If *a priori* knowledge of object features is given as abstract information, then the resulting segmentation could also be evaluated based on Euclidean distance measure of segmented objects with the reference objects. Alphanumeric feature references $\mathbf{f}_{ref}{}^{T}$ = ($f_{1,ref}$, ..., $f_{h,ref}$) are given by a human expert. As features may be in different scales or order of preference with respect to the segmentation task, a weighting vector $\hat{\mathbf{w}}^{T}$ = ($\hat{w}_1$, ..., $\hat{w}_h$) is introduced. A quality criterion Q_{ed} using each parameter set for segmentation evaluation is then given in Eq. (2.41) by

$$Q_{ed} = \frac{1}{n_c} \sum_{i=1}^{n_c} \sum_{j=1}^{h} \hat{w}_i (f_{ij} - f_{j,ref})^2. \tag{2.41}$$

In this regard [72], fuzzy *a priori* formulation of reference features that are described by a set of membership functions to encompass a considerable level of feature variations in the reference set can be used. Trapezoidal membership functions $\mu_j(f_j)$ with four parameters (i.e. a_j, b_j, c_j, d_j defining f_j-values of edges of a trapezoid) were used to formulate reference features. Similarly, other forms to describe *a priori* appropriately could also be used according to the information available. Fuzzy membership μ of each segment *i* for each feature *j* is denoted as μ_{ij}. It is reasonable to calculate a product of fuzzy membership μ_{ij} of all *h* features when it is desirable to classify each single segment separately. This refers to a single class problem where a special object type is targeted in an image based on the presence of each feature (i.e. $\mu_{ij} > 0 \; \forall \, j$) necessary in its overall classification. Moreover, the total number of expected segments n_c in an image was also formulated as a feature of a single image segmentation process in such cases to facilitate evaluation process using a trapezoidal fuzzy membership function denoted as μ_c. Therefore, a criterion

$$Q_{fuzz}(\mathbf{p}) = \mu_c(\mathbf{p}) \cdot \frac{1}{n_c(\mathbf{p})} \sum_{i=1}^{n_c(\mathbf{p})} (\prod_{j=1}^{h} \mu_{ij}(\mathbf{p})), \tag{2.42}$$

based on aforementioned logic, is introduced to express the quality of automatic image segmentation when single class is to be handled based on its feature vector. A classifier is needed to be designed for multiclass problem if object labels are given as absolute ground truth.

2.6.2 Classification measure

For each image $\mathbf{I}_{\text{seg}}$, BLOBS are compared for class types using $\mathbf{I}_{\text{truth}}$[5]. As BLOBS do not perfectly match the reference, class assignment is needed to be synthesized. For each BLOB $\mathcal{S}_u$ in

[5] This measure can only be used if ground truth image in terms of class labels is available

$\mathbf{I}_{\text{seg}}$, a class is assigned based on a certain feature set $\mathcal{F}$ and then compared to class type K_v of corresponding BLOB $\mathcal{T}_v$ in $\mathbf{I}_{\text{truth}}$.

Let K_u represent the class assignment for each $\mathcal{S}_u$ and $\hat{K}_u$ represent the class assignment by a classifier (given by $\mathbf{I}_{\text{seg}}$). Then,

$$K_u = \underset{k=0,\cdots,K_n}{\text{argmax}}(\text{card}\left\{i \in \{1,\cdots,C_u\}|I_{\text{truth},S_{ux,i},S_{uy,i}} = k\right\}) \tag{2.43}$$

such that, the quality criterion could be written as:

$$q_3 = \frac{1}{U}\sum_{u=1}^{U}|\text{sign}(K_u - \hat{K}_u)| \tag{2.44}$$

This criterion (2.44) was then accordingly converted to a fuzzy criterion using (2.4) such that:

$$\mu_3 = \mu(q_3; 2, 0, 1). \tag{2.45}$$

2.6.3 Total quality measure

In terms of providing feedback to segmentation block, the total quality measure is given as:

$$Q(r,b,n) = \mu_1 \cdot \mu_2 \cdot \mu_3. \tag{2.46}$$

where, μ_1, μ_2 and μ_3 were described earlier in Eq. (2.37), (2.39) and (2.45). For a certain parameter set $\mathbf{p}$ that has to adapted in order to improve $Q(r,b,n)$, quality measures for BLOBS dependent on $\mathbf{p}$ can be represented as $\mu_1(\mathbf{p})$, $\mu_2(\mathbf{p})$ and $\mu_3(\mathbf{p})$ and total quality measure is given as:

$$Q(r,b,n,\mathbf{p}) = \mu_1(\mathbf{p}) \cdot \mu_2(\mathbf{p}) \cdot \mu_3(\mathbf{p}) \tag{2.47}$$

The criterion (2.46) needs to be maximized with respect to the parameter vector $\mathbf{p}$ in order to obtain the optimal parameter set $\mathbf{p}_{opt}$ as given below:

$$\mathbf{p}_{opt} = \arg\max_{\mathbf{p}} Q(r,b,n,\mathbf{p}) \tag{2.48}$$

The optimal segmentation results are obtained using individual $\mathbf{p}_{opt}$ for each method.

For a quality measure dependent upon different types of noise such as n_1,n_2 and n_3 based on artifact function $A(n_1,n_2,n_3)$ mentioned in Section 2.2.4, a parameter set $\mathbf{p}$ can be tuned based on quality measures for BLOBS dependent on **p** represented as $\mu_1(\mathbf{p})$, $\mu_2(\mathbf{p})$ and $\mu_3(\mathbf{p})$. Consequently, and total quality measure $Q(n_1, n_2, n_3, \mathbf{p})$ is given as:

$$Q(n_1, n_2, n_3, \mathbf{p}) = \mu_1(\mathbf{p}) \cdot \mu_2(\mathbf{p}) \cdot \mu_3(\mathbf{p}) \tag{2.49}$$

The criterion (2.49) is maximized with respect to the parameter vector **p** in order to obtain the optimal parameter set $\mathbf{p}_{opt}$:

$$\mathbf{p}_{opt} = \arg\max_{\mathbf{p}} Q(n_1, n_2, n_3, \mathbf{p}) \tag{2.50}$$

$\mathbf{p}_{opt}$ is then used for final segmentation of image for each method.

2.6.4 Robustness measure

In order to evaluate the effect of selecting a specific segmentation/classification method selection in presence of described artifact levels, a quality measure is required to be designed for robustness. Such quality measures are usually based on calculation of area under the curve, adapting a tansig-regression and finding second derivative etc. The images selected for performance evaluation constituted an image series $\mathbf{I}(1, k, k+1)$ where $k = 1,...,B$. The resulting series has images in such a way that each succeeding image has both increased shading and noise level in comparison to the one preceding it. Here, a robustness measure for an algorithm based on increasing artifact levels for a particular dataset r is described as:

$$R = \frac{\sum_{b=1}^{B} Q(r, b, b+1)}{B}. \tag{2.51}$$

The performance of different algorithms will be compared based the robustness measure given in (2.51) when classification results are fed back to tune automatic parameter adaptation in image segmentation.

3 Results

3.1 Overview

In this chapter, the results of new feedback-based methods for automatic parameter tuning are discussed. The benchmark datasets described in Tab. 2.1 would be used in the following sections for the application of new feedback-based methods. The comparison would be made between feedforward and feedback implementations of standard image processing algorithms for each benchmark dataset (excluding the natural biological cell Benchmark datasets 2 and 4).

The results for each benchmark dataset are summarized and discussed in the summary and discussion sections following each benchmark.

3.2 Benchmark dataset 1

This section describes the results of standard image segmentation algorithms when applied to Benchmark dataset 1 in comparison with automatic feedback parametric tuning of the same algorithms using fuzzy *a priori* knowledge. The *a priori* knowledge was based on the difference between number of object counts relative to ground truth and the misclassification of those objects based on ground truth objects. Benchmark dataset 1 contained roughly from 20 to 70 objects in different scenes. So, based on quality measures given in (2.36), (2.38) and (2.44), respective membership functions for *a priori* were defined according to (2.37), (2.39) and (2.45).

Exemplary processing pipeline: An exemplary pipeline for standard algorithms given in Fig. 3.1 is proposed to apply the aforementioned methods and thus to compare two or more subroutines in an image processing pipeline. This pipeline was used for testing standard algorithms by evaluation of their performance on the Benchmark dataset 1 (see solid lines in Fig. 3.1). This pipeline is used to adapt parameters in the feedback fashion as shown by the dotted lines in Fig. 3.1.

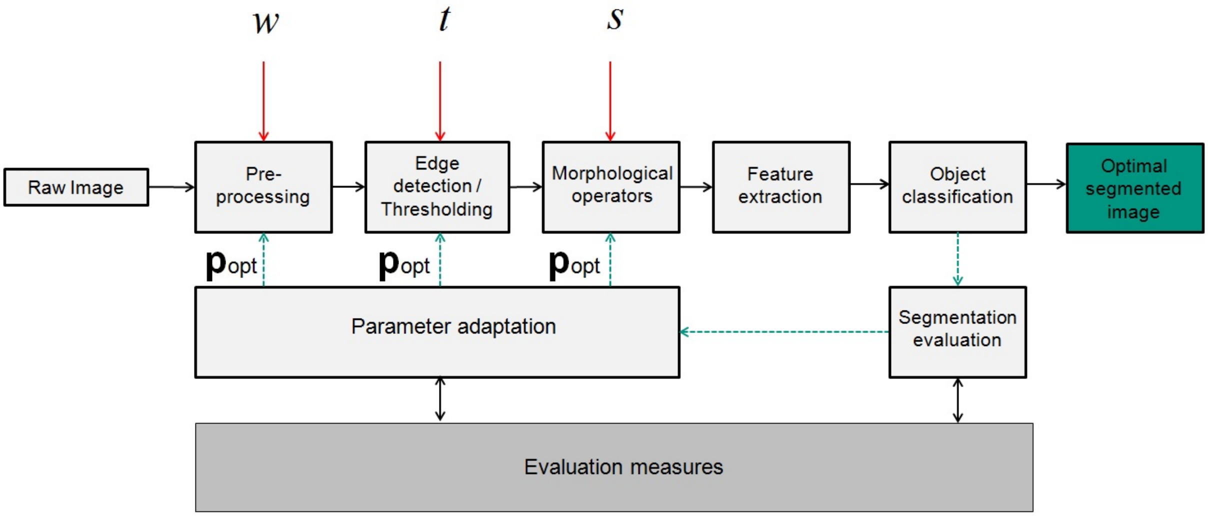


Figure 3.1: Exemplary pipeline for the segmentation of images in Benchmark dataset 1. w is the convolution filter size defined as symmetric $w \times w$ square matrix having elements equal to $\frac{1}{w^2}$. t is the image intensity threshold and s is the size of structuring element to be used for image opening operation.

3.2.1 Standard image segmentation/classification routines

Two standard and most famous segmentation methods i.e. Otsu thresholding and edge detection were chosen to show the effect of increasing artifact level on segmentation outcome. Using Benchmark dataset 1, as described earlier in Section 2.2.2, the results of segmentation from the above mentioned non-adaptive algorithms were compared with their corresponding parameter adaptation and the results were compared in terms of evaluation criteria developed based on *a priori* knowledge about the objects to be found in the Benchmark dataset 1. This knowledge, for simplicity, was based on error in terms of number of objects detected and difference in pixels of the objects detected in comparison to the ground truth objects.

In the pipeline structure shown by solid lines (see Fig. 3.1), a grayscale image is first normalized using 2% and 98% quantiles in the pre-processing step. The normalized image is then segmented using standard image segmentation techniques. Here, the aim is to use the benchmark to compare a standard Otsu operator to a Sobel edge detector using the threshold parameter t. This parameter t is selected automatically by each individual algorithm for each artifact level based on the image intensity map. In Otsu's method application, the resulting binary image is then passed through morphological operations such as image hole filling, image dilation (using 7× 7 pixels window as a structuring element), image opening (using a structuring element of disk with radius 3) and image border clearing in order to get reasonable objects. A large size of image dilation parameter was selected in order to detect very small objects that may get occluded due to presence of artifacts. In Sobel edge detection, the resulting binary image was passed through the same sequence of operators but the structuring element of image dilation was selected to be of size 3×3 pixels. Each object is described by 8 features to design a Bayes classifier. Features are area, eccentricity, mean intensity, maximum intensity, minimum intensity, solidity, median intensity and standard deviation of intensity values. This pipeline has been implemented as a macro in the Gait-CAD software ([136]) developed in MATLAB and is provided with the implementation code.

The image processing pipeline given in Fig. 3.1 is used to compare the segmentation steps i.e. Otsu operator versus a Sobel edge detector. For application, scene $r = 1$ of Benchmark dataset 1 was used. As a native edge detection does not directly seek for objects[1], the edge detector is expected to deliver worse results. The robustness of each method in a pipeline is quantified using (2.51) and scene $r = 1$[2].

The performance of the pipeline, once using the Otsu operator and once using the Sobel edge detector with respect to increasing artifact levels is given by dotted lines in Fig. 3.2. The segmentation evaluation measures are integrated in the total quality measure $Q(r, b, n)$.

Fig. 3.2 shows a decreasing performance (dashed lines) of the segmentation algorithms indicated by $Q(r, b, n)$ with increasing artifact levels $A(r, b, n)$. To quantify the performance degradation, the robustness measures R_{otsu} and R_{edge} for Otsu thresholding and Sobel edge detection, respectively are given according to (2.51). R is given on a scale between 0 and 1 where larger values show high robustness. Both segmentation methods are found to be comparatively closer (see R values in Fig. 3.2) to each other. At higher artifact levels, edge detection fails to find the adequate number of objects as compared to Otsu thresholding but classification results are seen to be better than those obtained with Otsu segmentation.

3.2.2 New feedback-based parametric tuning for standard image segmentation/classification routines

In this section, both segmentation methods i.e. Otsu thresholding and edge detection were used in such a way that parameters involved in the outcome of image segmentation were iteratively adapted using a quality criterion to be fulfilled. The automatic feedback-based image segmentation tuning is given with results in this section. Not only methods are compared with each other, but the tendency for improvement in both methods by using the new feedback parameter adaptation method was also shown. The deviation in quality at each artifact level for different scenes are given in [76].

[1] However, the subsequent hole filling delivers objects in undistorted cases.

[2] Scene $r = 1$ was selected due to the lower number of foreground objects and less complex cases to make this demonstration more understandable.

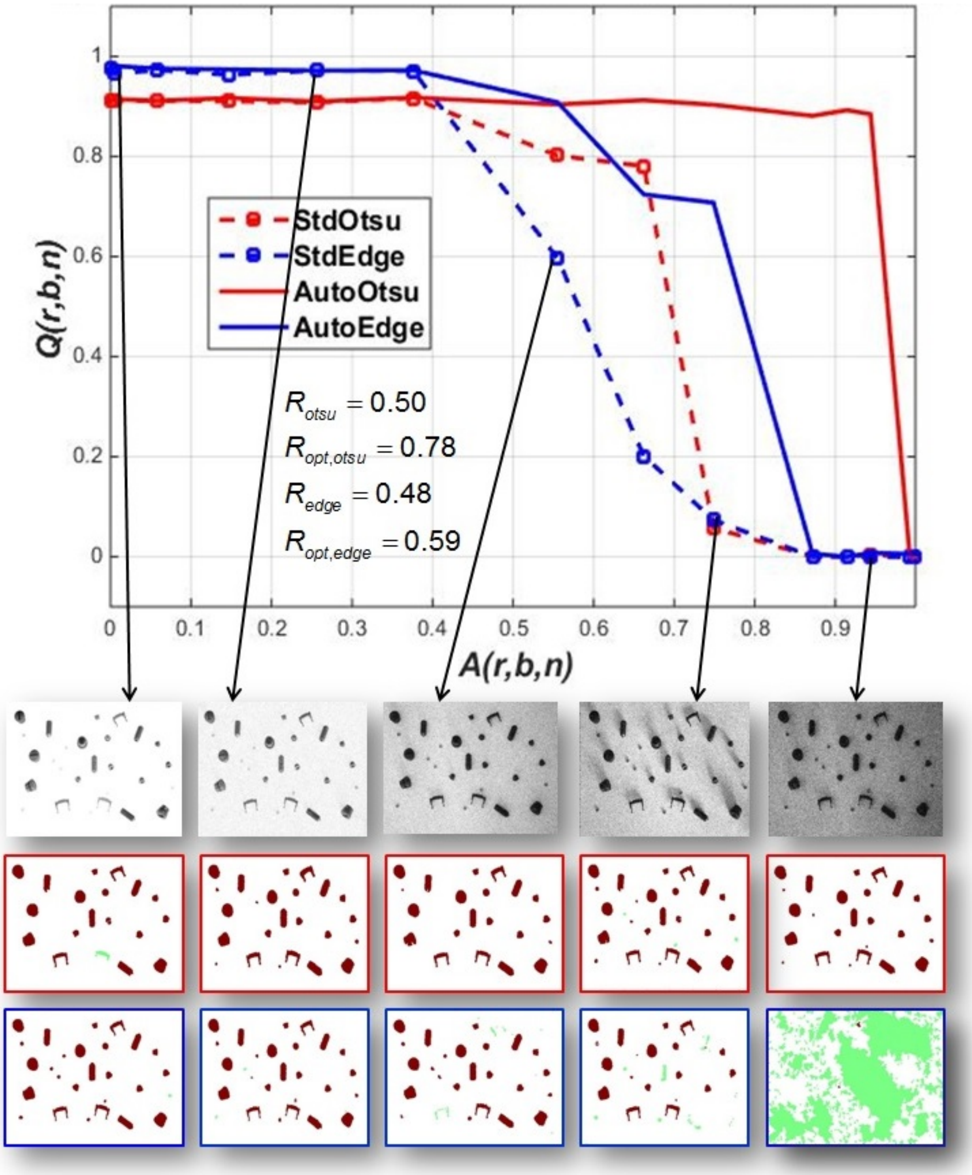


Figure 3.2: Results of the adaptation of image segmentation parameter on Benchmark dataset 1, *r* =1. $Q(r, b, n)$ vs. $A(r, b, n)$. In the first row, original images from dataset *r* = 1 are given. The second row shows corresponding segmentation and classification results using parametric feedback tuning of Otsu's method. The third row shows corresponding segmentation and classification results using similar tuning of edge detection method. Red and green colors represent correct and wrong classification of the segmented BLOB respectively compared to ground truth. Robustness values for Otsu's method and Sobel edge detection are R_{otsu} = 0.50 and R_{edge} = 0.48 respectively. Where as, robustness values for the new automatic feedback parametric tuning of aforementioned methods are $R_{opt,otsu}$ = 0.78 and $R_{opt,edge}$ = 0.59 respectively.

The optimization is done independently for each artifact level by adapting parameters [3]. In case of datasets containing images with unknown artifact levels, one can ascertain artifact level by estimating the background of the image. If ground truth is available, then it becomes relatively easy to estimate the background using the intensity values of pixels.

Parametric adaptation can be done using different pipeline structures and more parameters could also be involved. Hereby, the parameter vector **p** was reduced just to one parameter for both segmentation methods such that **p** = t_{otsu} for Otsu's method and **p** = t_{sobel} for Sobel edge detection. For parameter tuning of **p**, (2.48) is adapted based on criterion (2.47).

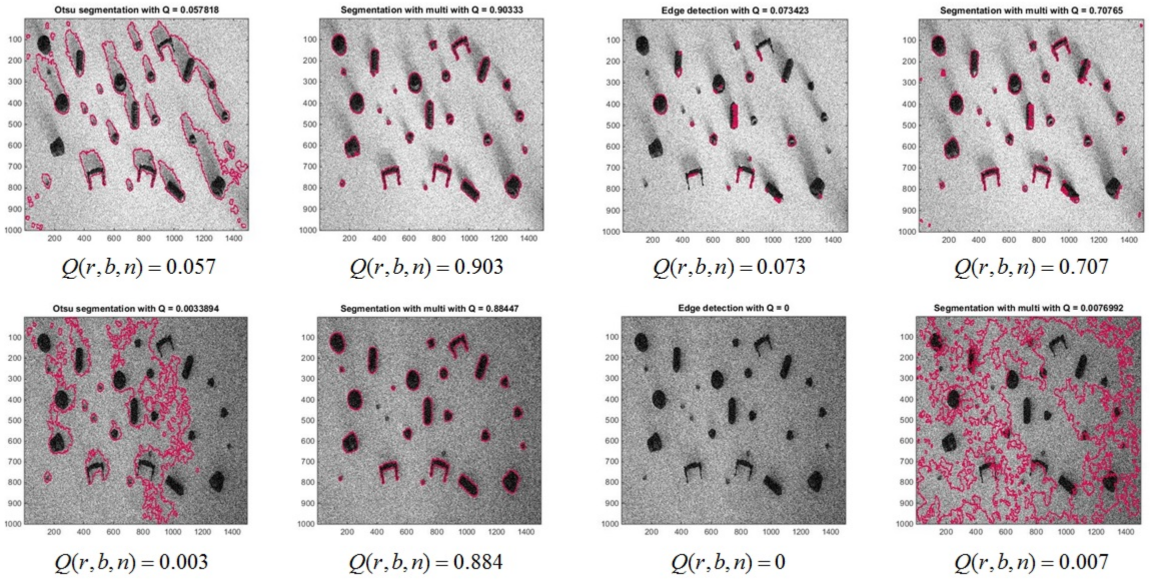
Segmentation outcome for Benchmark dataset 1 was evaluated for $r = 1$. Using image processing pipeline given by dotted line in Fig. 3.1, parameter adaptation of both Otsu's method and Sobel edge detection to scene $r = 1$ is applied. This was done to show the effect of image processing parameter adaptation on the outcome of an image processing pipeline in order to compare the segmentation steps i.e. Otsu operator versus a Sobel edge detector. For standard feedforward application of both segmentation methods, the performance was indicated by robustness measure R (i.e. values for Otsu thresholding and Sobel edge detection are R_{otsu} = 0.50 and R_{edge} = 0.48 respectively). In order to show a better performance for feedback parameter adaptation, robustness values should eclipse the former ones using (2.51).

The performance for parameter adaptation of the two methods, one using the Otsu operator and the other using the Sobel edge detector with respect to increasing artifact levels in comparison to their standard feedforward application is shown in Fig. 3.2.

From Fig. 3.2, it can be seen that using automatic feedback parameter adaptation method, the performance of both methods can be improved especially at higher artifact levels. This is evident by robustness values of both segmentation methods i.e. $R_{opt,otsu}$ = 0.78 and $R_{opt,edge}$ = 0.59. Moreover, it is also observed that, using parameter adaptation, the value of $Q(r, b, n)$ at any artifact level for both methods cannot be worse than that of standard feedforward methods.

It is also clear that at higher artifact levels (i.e.greater than 0.6), Otsu's method outperforms Sobel edge detection. At lower artifact levels, Sobel edge detection has marginally better outcome than Otsu's method. This could be understood better by seeing individual segmentation and corresponding classification outcomes at certain artifact levels. For instance, at $A(r, b, n) \approx$ 0.75 the segmentation outcome of feedforward/feedback-based Otsu's method and edge detection is shown in the first row of Fig. 3.3.

[3] For Otsu's method, the intensity threshold t was adapted over a parameter variation from 0 to 0.8 with the resolution of 0.01. The limits of t_{otsu} can be chosen based on t_{otsu} values from the standard application of Otsu shown in previous section. In case of edge detection, again only one parameter for Sobel edge detection was adapted to provide a fair comparison with the non-feedback method. This parameter was again the threshold t_{sobel} for thresholding the calculated gradient magnitude of image intensity. This parameter is adapted between 0 and 0.2 using knowledge from the standard application of Sobel edge detection described in previous section. The step size for iterative search was selected to be 0.005. For more information on fixed parameters and parameters to be optimized, see Tab. 3.1 and 3.2.



$Q(r,b,n) = 0.057$ $Q(r,b,n) = 0.903$ $Q(r,b,n) = 0.073$ $Q(r,b,n) = 0.707$

$Q(r,b,n) = 0.003$ $Q(r,b,n) = 0.884$ $Q(r,b,n) = 0$ $Q(r,b,n) = 0.007$

Figure 3.3: Segmentation outcome (shown by red outlines) for Otsu segmentation and Sobel edge detection: feedforward vs. new feedback parameter adaptation method at artifact level $A(r, b, n) \approx$ 0.75 in the first row and $A(r, b, n) \approx$ 0.94 in the second row. First column: segmentation result for feedforward Otsu segmentation. Second column: segmentation result for Otsu segmentation using feedback parameter adaptation. Third column: segmentation result for Sobel edge detection using standard feedforward application. Fourth column: segmentation result for Sobel edge detection using feedback parameter adaptation.

It can be seen in third image of first row in Fig. 3.3, that due to the presence of shadows, edges are not fully detected and predefined image dilation parameter is not able to form all BLOBs resulting in a low $Q(r, b, n)$ value using standard feedforward application. Conversely, when threshold to edge detection is adapted iteratively, comparatively better segmentation quality is obtained at high artifact level (see fourth image in row 1 of Fig. 3.3).

Wrong classification assignments occur due to incorrectly segmented objects only at extremely high artifact levels for Sobel edge detection (see third and fourth image in row 2 of Fig. 3.3), whereas Otsu's method is more efficient at higher artifact levels in comparison to Sobel edge detector (see first and second image of row 2 in Fig. 3.3).

Pipeline structure	Segmentation method	Fixed parameters	Optimized parameters	Robustness values R	Result links
Feedforward (solid lines) Fig. 3.1	Otsu	w,s,t_{otsu}	–	0.50	Red dashed line in Fig. 3.2
	Edge	w,s,t_{sobel}	–	0.48	Blue dashed line in Fig. 3.2
New feedback (dashed lines) Fig. 3.1	Otsu	w,s	t_{otsu}	0.78	Red solid line in Fig. 3.2
	Edge	w,s	t_{sobel}	0.59	Red solid line in Fig. 3.2

Table 3.1: Results showing robustness of feedforward vs. feedback pipeline structures (segmentation/-classification) for two different image segmentation methods in Benchmark dataset 1.

3.2.3 Automatic image segmentation/classification using abstract ground truth

To find objects based on user abstract information is also possible. The user has to define objects to be found, e.g. he may want to find out an object class e.g. a set screw (encircled green in Fig. 3.4) among all other objects in images of $r = 1$. By looking at an image, one can see that such objects are medium-sized, greater in eccentricity and have higher solidity. The features selected are area f_1 (in pixels), eccentricity f_2 and solidity f_3. Based on trapezoidal fuzzy membership functions, these features are represented as $\mu_1(f_1)$ with (a_1, b_1, c_1, d_1) = (0, 5000, 7000, 8000), $\mu_2(f_2)$ with (a_2, b_2, c_2, d_2) = (0.6, 0.85, 0.95, 0.96) and $\theta_3(f_3)$ with (a_3, b_3, c_3, d_3) = (0.85, 0.95, 1, 1). Moreover, $n_c = 4$ for set screws in $r = 1$. This can be formulated as well in the evaluation criterion using $\mu_c(n_c)$ with (σ, o) = (2, 4).

$\mathbf{p} = t$ was selected as shown in Fig. 3.1. The pipeline is first operated without classification block. For each **p**, feasible objects are selected that maximize (2.28). For Otsu's method and Sobel edge detection, t_{otsu} and t_{sobel} were adapted respectively. It is object based since each detected object is compared against the reference features and objects that maximize (2.28) are selected.

The results for detecting set screws in $r = 1$ are obtained throughout the graded dataset with increasing artifact levels. An example of image segmentation/classification at a medium artifact level ($A(r, b, n)$ = 0.55) is given in Fig. 3.5. Firstly, both methods are applied in feedforward fashion (see results in first column of Fig. 3.5). All fixed parameters remained same as in the case of using absolute ground truth (see Tab. 3.1). Then, feedback adaptation of both methods is done by adapting a single parameter i.e. t in both cases and the improvement in results can be seen in the second column of Fig. 3.5.

Later, the classification block is introduced into the pipeline, and the results (feedforward vs.

Algorithm type	Parameter/Operator name	Parameter setting	Processing step	Range of tuned parameters (min : step size : max)
1,2,3,4	image filling	'holes'	morphological operations	
1,3	image dilation	SE: 7x7 ones	morphological operations	
2,4	image dilation	SE: 3x3 ones	morphological operations	
1,2,3,4	image opening	SE: $s = 3$ 'disk'	morphological operations	
1,3	t_{otsu}	fixed for 1	thresholding	0 : 0.01 : 0.8
2,4	t_{sobel}	fixed for 2	edge detection	0 : 0.005 : 0.2
1,2,3,4	number of features	8	feature extraction	

Table 3.2: Fixed and optimized parameters for each processing step in Benchmark dataset 1. Here, 1: StdOtsu, 2: StdEdge, 3: AutoOtsu, 4: AutoEdge

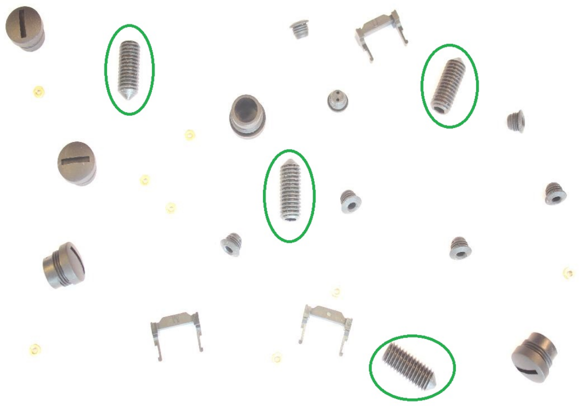


Figure 3.4: Using abstract ground truth information to find one targeted object in the scene. Set screws encircled in green are to be found in $r = 1$

feedback) are shown in Fig. 3.6. As the artifact level is increased, it is hard to detect the targeted objects based on user-defined knowledge. However, one can use the overall evaluation measure (2.48) (since it incorporates the classification outcome evaluation too) to adapt an optimal parameter set $\mathbf{p}_{opt}$ based on objects selected by maximizing (2.28) at each $\mathbf{p}$.

Feedback parameter adaptation results are shown in solid lines for both the methods in Fig. 3.6. Marginal or mostly no improvement was observed in the case of Otsu segmentation even by

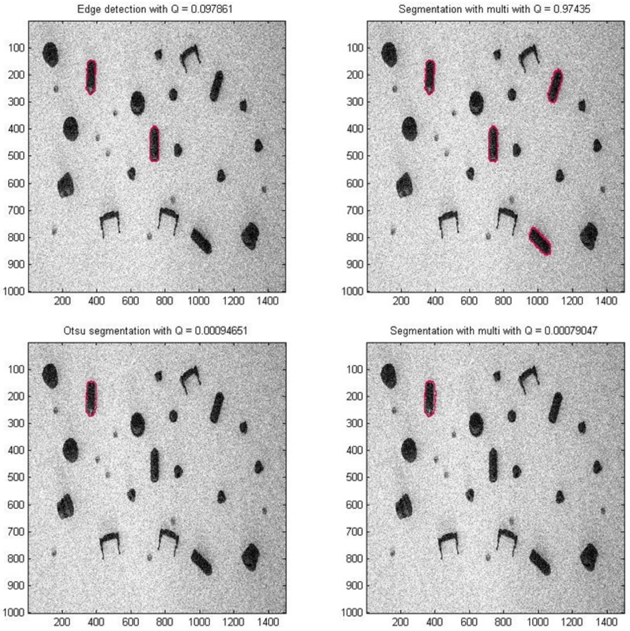


Figure 3.5: Otsu segmentation and Sobel edge detection for abstract ground truth defined by a user using only one object type at $A(r,b,n)$ = 0.5553. First column: feedforward application of both methods. Second column: feedback application of both methods. First row: segmentation result using Sobel edge detection (using red colored delineation). Second row: result from Otsu segmentation results using red colored delineation.

adapting t_{otsu}. For Sobel edge detection, there was a considerable improvement in results by adapting t_{sobel}. At $A(r,b,n) \geq 0.6$, both methods perform poorly both in feedforward and feedback structures. This is due to the fact that feature extraction is greatly disturbed as noise is detected because optimization is clearly looking for certain objects with user-defined input reference features.

In Fig. 3.7, the effect of segmented objects on classification accuracy is shown at different artifact levels in case of both segmentation methods using feedback parameter adaptation. It is seen in Fig. 3.7, that as $A(r,b,n)$ increases, number of correctly classified objects decreases. The robustness results according to each pipeline structure are given in Tab. 3.3.

3.2.4 Summary and discussion

The presented framework contains images of the same scenes under varying illumination conditions and noise levels as well as the ground truth for segment detection and object type classification. Furthermore, measures to evaluate the artifact level, segmentation and classification quality as described in Section 2.2.2 were used to evaluate illumination robustness of Benchmark dataset 1. Thus, robustness evaluation of image processing and classification algorithms becomes

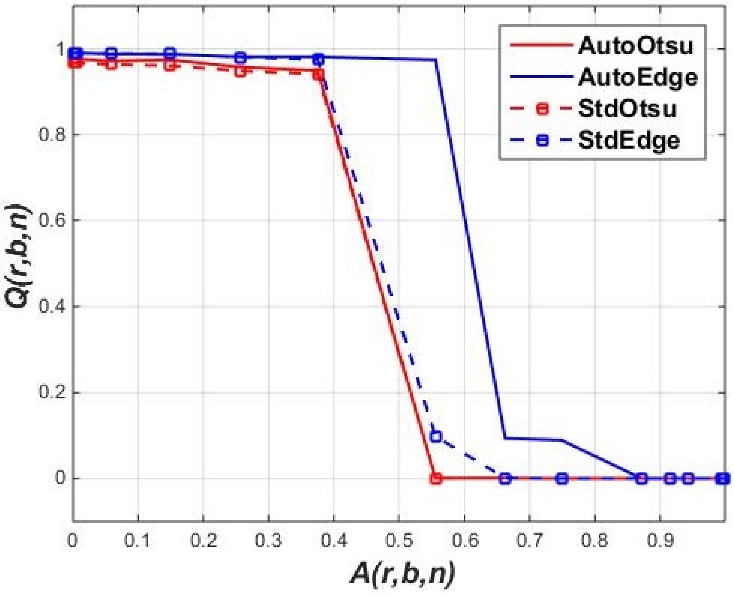


Figure 3.6: Segmentation outcome for Otsu segmentation and Sobel edge detection for abstract ground truth defined by end user using only one object type with increasing $A(r,b,n)$. Dotted red and blue lines show outcome of Otsu segmentation and Sobel edge detection respectively with fixed parameter values respectively whereas solid red and blue lines show parameter adaptation using t.

possible and enables developers to compare image processing algorithms with respect to the robustness. To this end, the development of algorithms with a focus on robustness in distorted data is accelerated. Parameters and structures can easily be evaluated and optimized.

Furthermore, not only the robustness of an image-processing pipeline can be evaluated but also the segmentation quality. Parameters can be optimized with respect to the given data quality. For example, if the quality of images is known to be bad, algorithms do not need to provide the optimal results on good quality images if the parameter adaptation of the algorithms used is not employed.

In addition, the outcome of an image processing pipeline can be fed back to optimize its parameters. Therefore, users have a means to tune the parameters in a pipeline not only to fit a special set of images but could also be applied to a more general class of problems.

The possibility to describe an image set in a rather abstract fashion was also shown with results. A user may define targeted objects in terms of estimated a priori knowledge, e.g. estimated size and number of objects. Parameters are optimized to find these described properties. Furthermore, in combination with a classifier, the method can also be used to extract a certain kind of object out of all segmented objects and therefore build an image processing pipeline being capable of picking object classes out of an image. This was also shown with pipeline feedback using parameter adaptation against a pipeline using feedforward structure for standard segmentation methods.

Pipeline structure	Segmentation method	Fixed parameters	Optimized parameters	Robustness values R	Result links
Feedforward (solid lines) Fig. 3.1	Otsu	w,s,t_{otsu}		0.41	Red dashed line in Fig. 3.6
	Edge	w,s,t_{sobel}		0.43	Blue dashed line in Fig. 3.6
Feedback (dashed lines) Fig. 3.1	Otsu	w,s	t_{otsu}	0.41	Red solid line in Fig. 3.6
	Edge	w,s	t_{sobel}	0.51	Red solid line in Fig. 3.6

Table 3.3: Results using abstract ground truth for Benchmark dataset 1: Parameters and robustness of feedforward vs. feedback pipeline structures (segmentation/classification) for two different image segmentation methods.

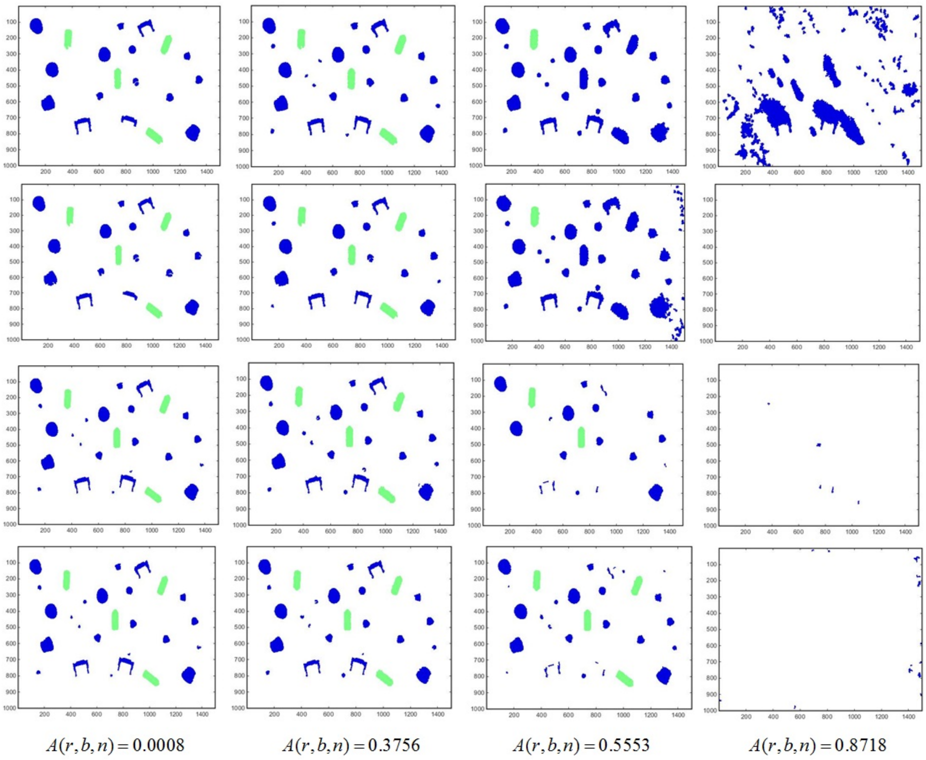


Figure 3.7: Segmentation/classification outcome for Otsu segmentation and Sobel edge detection of one object type using feedback method against increasing artifact levels along the columns from left to right. First row: Otsu segmentation using feedforward application. Second row: Otsu segmentation with feedback adaptation of parameter **t**. Third row: Sobel edge detection using feedforward application. Fourth row: Sobel edge detection with feedback adaptation of parameter t_{sobel}. Green color shows right classification of segments and blue color shows the objects that are not classified as set screws.

Basically each part of the framework (images, artifact level calculation, quality calculation, robustness) can individually be replaced, depending on the preferences of the developer. Even the images may be replaced by arbitrary images as long as the calculation of the artifact level (depending on the parameter to be used, here: mean pixel value) delivers reasonable results. Using the proposed benchmark, the effect of increasing artifact levels on the image segmentation outcome using standard algorithms is shown. The proposed quality measures may be used for other object classification benchmark without a special robustness focus as well.

The presented new parameter adaptation method of image processing parameters was also shown using benchmark dataset containing illumination and artificial Gaussian noise artifacts. A comparison was presented in terms of how much could a segmentation outcome be improved if the targeted parameter/parameters are searched iteratively in a search space. These results were shown to be robust at higher artifact levels where standard feedforward application fails to produce adequately desirable results.

The quality of the optimal image segmentation depends upon how well a certain targeted parameter is found. This, indirectly, refers to the step size or the resolution of search space contained within its bounds. These results can be further improved by increasing the resolution or further decreasing the step size of the parameter set to be optimized within the bounds. However, careful bounds and step size should be chosen in order to avoid useless drifting in unwanted parameter space. A nonlinear optimization could also be performed to search the optimal parameters more efficiently.

Furthermore, the new feedback-based parameter adaptation method can be further fine-tuned by making structural changes to an image processing pipeline. Image normalization has not been done in this dataset since even without the background noise removal, image segmentation quality has shown to be improved. Similarly, induced Gaussian noise and shading effects do not impart too big a challenge for segmentation as in case microscopic images containing inherent microscopic noise. Therefore, with this dataset, only parameter adaptation involved directly in image segmentation is adapted.

3.3 Benchmark datasets 2 and 4

Thee benchmark datasets described in Sections 2.2.3 and 2.2.5 were used for both image normalization and segmentation using feedback parameter adaptation. Section 3.3.1 shows normalization using feedback parameter adaptation to discuss the improvement in automatic feedback image segmentation. However, feedforward standard method for image segmentation is compared to feedback parameter adaptation of image segmentation parameters in Section 3.3.2. The overall improvement using both feedback-based image normalization/segmentation is discussed in Section 3.3.3.

3.3.1 Automatic feedback-based image normalization

Benchmark dataset 4: The details and reference evaluation of this dataset is given in Section 2.2.5.

Since only a partial ground truth (only in terms of total number of cells to be found in an image) is available, the mean size ($x_{1,mean}$) of cells in pixels obtained from image segmentation step should also be considered as a valid segmentation criterion. Due to background noise present in images, image segmentation does not produce desired segments. Therefore, automatic image normalization was performed and the degree of success is judged by seeing the improvement in image segmentation results based on the evaluation criteria. The segmentation results obtained from new feedback-based normalization can be seen from values of σ_{GD} and $x_{1,mean}$ in Tab. 3.4 later in Section 3.3.2.

$\mathbf{p}_{opt}$ was adopted by (2.7) based on (2.41). The images B_k, where k = 1...6, were used from Benchmark dataset 4. The plausibility of visual results with respect to human observation was slightly improved when using the new method. This is indicated in B_2 by a fine delineation of cells lying very close to each other. Since the shading effect around cells has been minimized in the normalization technique, more cells were detected in the whole dataset B_k. This effect is demonstrated in Fig. 3.8 (image on the right, section on middle top), where the outer boundaries of detected cells were seen to be hardly touching each other. So, much clearer and separated cell boundaries are found. It can be observed from Tab. 3.4 that by using automatic image normalization, the results in terms of σ_{GD} and $x_{1,mean}$ were improved which are aggregated using mean value μ in Tab. 3.4.

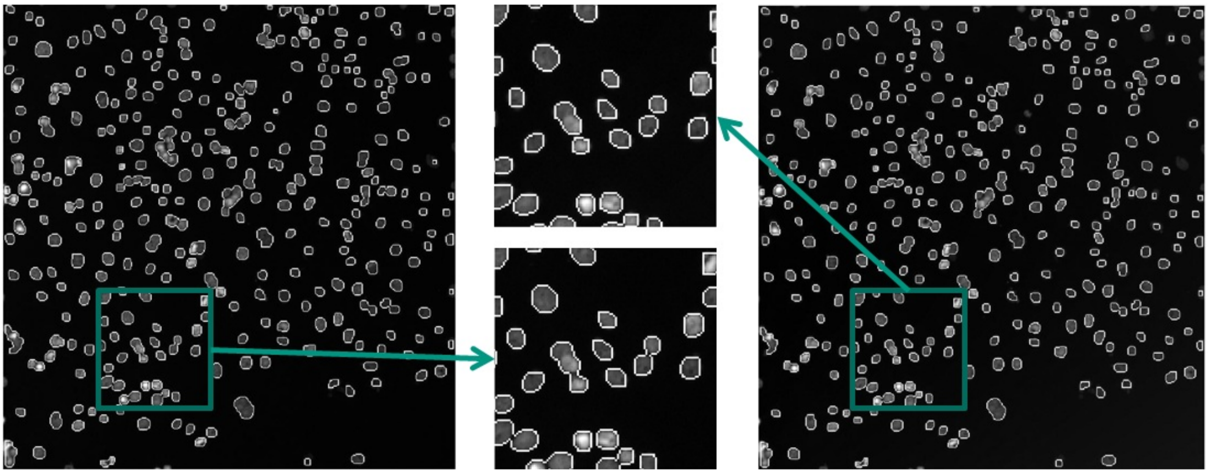
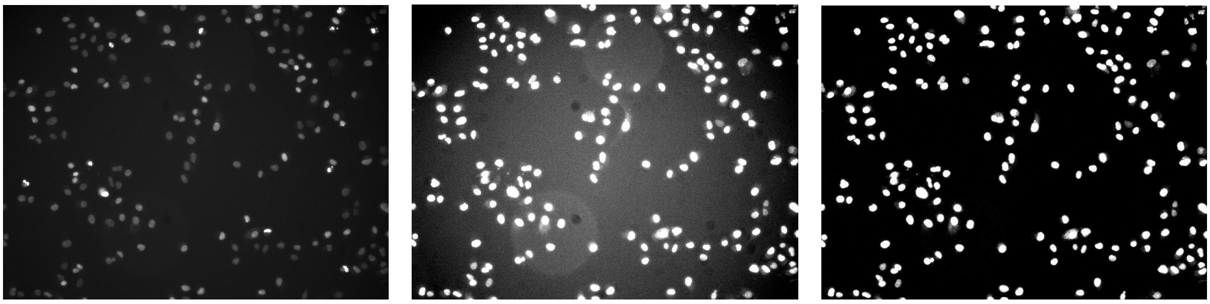


Figure 3.8: Comparison of visual results between image segmentation without normalization (left) with one using normalization (right) with sectioned images in the middle

Benchmark dataset 2: Here, images P_l where l = 1...4 were used from Benchmark dataset 2 as shown in Fig. 3.9(a). The aim is to apply automatic image segmentation on this dataset.

However, the shading effect has first to be removed to perform automatic image segmentation. As an example of shading, P_1 having numerous cells in different states, is illustrated in Fig. 3.9(a). The image suffers from an inconsistent background illumination since the background in corners is darker than in the middle of the image as shown in Fig. 3.9(b) using high contrast by applying 5 % percentiles using (4.2). Therefore, shading correction is imperative in order to disentangle image from such variations concealing original intensity information in an image. This is done to improve the image segmentation outcome.



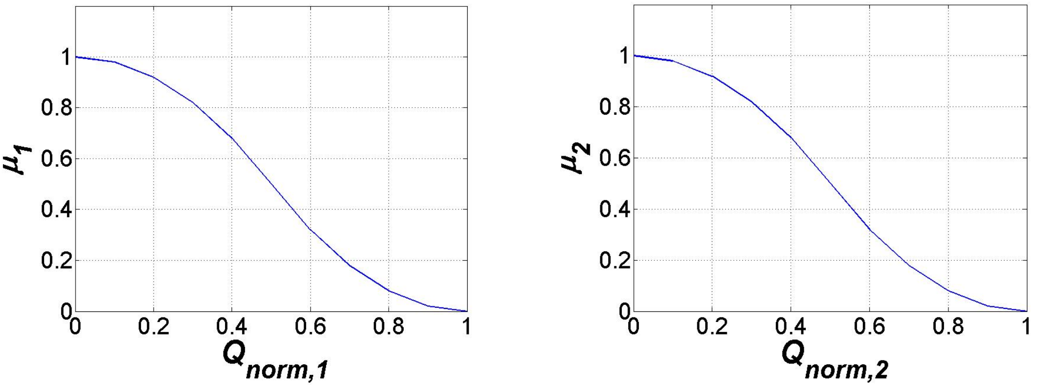
(a) Original grayscale image (b) Contrast-stretched image using 5 % percentiles (c) Contrast-stretched normalized image using 5 % percentiles

Figure 3.9: Shading effect in original grayscale image P_1 and its automatic correction using (2.41)

Normalization of P_1 using $\hat{\mathbf{a}}_{opt,fuzzy,norm}$ is done and resulting image is shown in Fig. 3.9(c). It is clear that shading effects are highly reduced. The spline-shaped fuzzy membership functions were used (see Fig. 3.10) for the evaluation of the normalization criteria (2.17) and (2.18). The results are presented in Tab. 3.5. It can be seen from Tab. 3.5 that segmentation criterion $Q_{fuzz,seg}$ was improved in case of using normalization before image segmentation. It demonstrates the beneficial effect of using a shading correction procedure. Uniform backgrounds were achieved (see $Q_{fuzz,norm}$ in Tab. 3.5) using the new automatic normalization technique, whereas in original images of P_l, criterion (2.18) yielded undesirably low values. All algorithms are implemented in MATLAB using the Image Processing Toolbox and the open source Gait-CAD Toolbox [94] for data mining.

3.3.2 Automatic feedback-based image segmentation

Benchmark dataset 4: Using the criterion (2.7), $\mathbf{p}_{opt}$ was adopted by (2.42). The results for all images B_k are given in Tab. 3.4 in terms of deviation from ground truth. In addition, a feed-forward automatic segmentation technique proposed by Otsu [103] was applied resulting in an optimal threshold t. The results, however, are not directly comparable since n_c could be indecisive especially when Otsu's method deliver such segments having size considerably smaller than the normal segments. To solve this problem, an image opening was applied in case of Otsu's



(a) $Q_{norm,1}$ vs. μ_1 with (a,b) = (0 1)

(b) $Q_{norm,2}$ vs. μ_2 with (a,b) = (0 1)

Figure 3.10: Fuzzy spline-shaped membership functions μ_1 and μ_2 for criteria (2.17) and (2.18).

method in order to remove erroneous small segments (opening filter size s = 3 and s = 5). With the addition of an image opening operation, the parameter vector for Otsu's method is described as $\mathbf{p}_{Otsu} = (s, t)^T$.

	New method (w/o norm.)		**New method (with norm.)**		**Otsu's method (s = 3)**	
Images	σ_{GD} (%)	$f_{1,mean}$	σ_{GD} (%)	$f_{1,mean}$	σ_{GD} (%)	$f_{1,mean}$
B_1	12	110	13	114	8	101
B_2	15	129	12	108	16	97
B_3	18	113	14	105	18	99
B_4	15	99	15	112	12	94
B_5	20	120	18	117	23	96
B_6	11	121	11	128	14	96
μ	**15**	**115**	**14**	**114**	**15**	**97**

Table 3.4: Reference cells detection for all B_k images by applying feedback-based image segmentation, first without automatic shading correction and then using it, in comparison to feedforward standard image segmentation using Otsu's method with s = 3

The results obtained from new feedback-based parameter adaptation technique were comparable to original Otsu's feed-forward method as can be seen from values of σ_{GD} and mean cell area $f_{1,mean}$ in Tab. 3.4. The aggregated results using mean value μ in Tab. 3.4 show that the new scheme was able to detect cell numbers closer to those detected by Otsu's method but with larger mean cell area, the direct relevance or comparison of which is not stated in *a priori* reference of B_k. The μ of σ_{GD} was equal to 15 in case of the new method and Otsu's method with s = 3. However, visual results with respect to human observation seem much more plausible when using the new method. This is indicated by larger value of μ of $f_{1,mean}$ equal to 115 in case of the new method as opposed to Otsu's method with s = 3 having μ of $f_{1,mean}$ equal to 97. This effect is demonstrated in Fig. 3.11, where the demarcation of detected cells were seen to be inside the cell boundaries yielding smaller cell areas in case of Otsu's method with s = 3.

Therefore, it can be inferred that the new method not only detects and counts the cells comparable in numbers to Otsu's method that uses an image opening filter manually, but is also able to select image opening filter size automatically. Moreover, it yields much better results from subjective point of view.

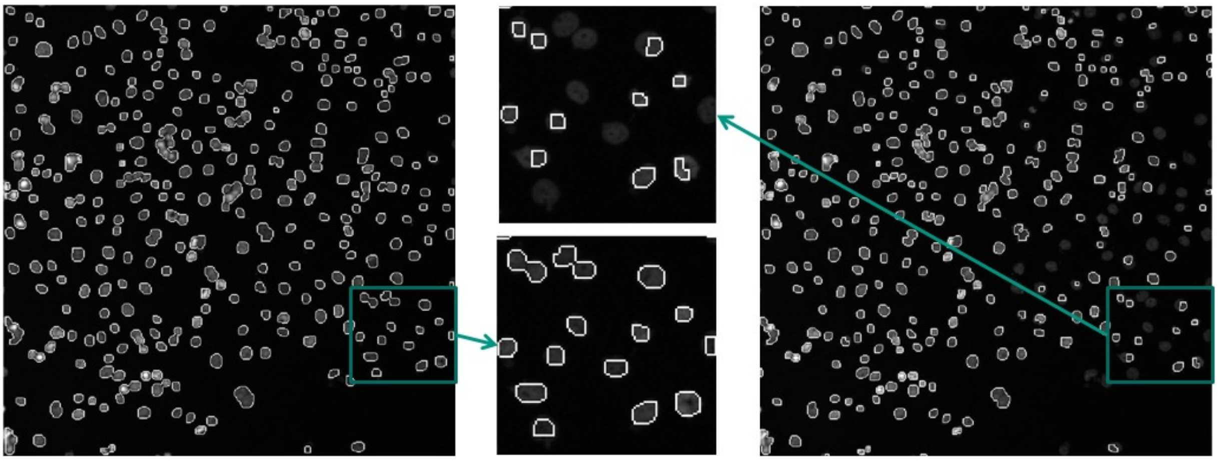
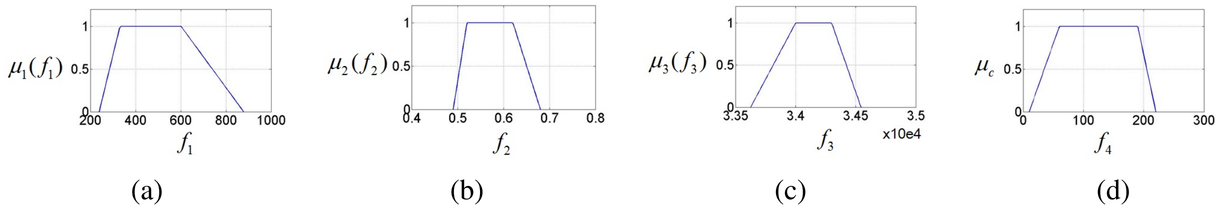


Figure 3.11: Cell detection of B_2 using the new feedback technique (left) in comparison to Otsu's method (s = 3) (right) with zoomed sections in the middle. It is visible through the sectioned images in the middle that my using new normalization method, better cell boundary demarcations are obtained.

Benchmark dataset 2: The parameter vector to be optimized in this case is $\mathbf{p} = (w, s, t)^T$, where w, s and t represent convolution square matrix size, image opening disc size and brightness threshold respectively using (2.42) with h = 3 and (2.28). Three features namely, the area f_1, eccentricity (ratio between major axis and sum of major and minor axes) f_2 and the mean of the brightness f_3 for each segment were selected. Since the intent is to find specific cell states within an image, it is recommended to define a range of confidence over which the selected features can vary. The segments for normal looking cells were labeled manually in image P_1. *A priori* knowledge was described using $\mu_j(f_j)$ for $\mathbf{f}_{ref}^T$ as shown in the Fig. 3.12. It can be seen from Fig. 3.12(a), that for a normal cell, the area lies roughly between 330 to 600 pixels. Its roundness factor (0.5 for perfectly round segments) is between 0.52 to 0.62 as shown in Fig. 3.12(b) and its brightness can vary from 34000 to 34300 as shown in Fig. 3.12(c). Moreover, the number of cells, that can be found in each image of the given dataset, can be between 60 to 190 as represented by a fuzzy function in Fig. 3.12(d).

Furthermore, labeling of normal cells of all P_l images by adapting $\mathbf{p}_{opt,fuzzy}$ given in (2.28) using fuzzy feature evaluation was also performed. The labeling of normal cells were obtained based on abstract reference fuzzy features using (2.28) for optimal parameter set adaptation. Only P_1 was labeled for normal looking cells (n_{ref} = 76) as a reference and additional normal cells n_d were sought after in the rest of images of P_l. The cells (apart from n_{ref}) which fulfill the fuzzy



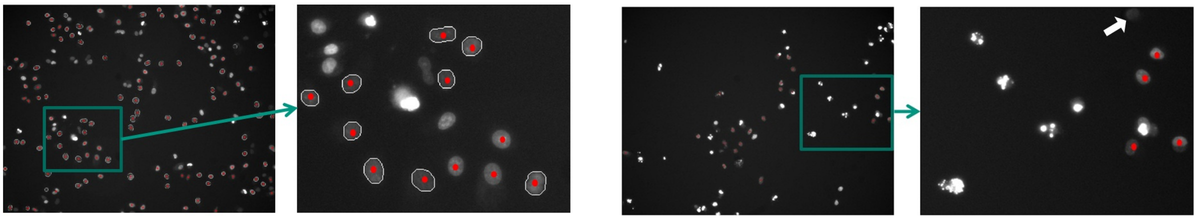
(a) (b) (c) (d)

Figure 3.12: $\mu_j(f_j)$ and μ_c for reference normal cells, where Fig. 3.12(a) $\mu_1(f_1)$ with (a_1, b_1, c_1, d_1) = (239, 330, 600, 878) represents area such that perfect normal cells are defined to be between 330 to 600 pixels, Fig. 3.12(b) $\mu_2(f_2)$ with (a_2, b_2, c_2, d_2) = (0.49, 0.52, 0.62, 0.68) represent eccentricity of cells such that perfect normal cells have eccentricity between 0.52 and 0.62, Fig. 3.12(c) $\mu_3(f_3)$ with (a_3, b_3, c_3, d_3) = (33626, 34000, 34300, 34544) represents that intensity of perfect normal cells lies between 34000 and 34300 and Fig. 3.12(d) μ_c with (a_4, b_4, c_4, d_4) = (10, 60, 190, 220) represents the membership function of number of normal cells to be found in an image. Mostly, it is assumed that 60 to 190 normal cells are present in any image.

criteria Q_{fuzz} of normal cells defined using (2.42) are n_d. The results for all P_l images are given in Tab. 3.5 in terms of n_c (sum of both n_{ref} and n_d), σ_{GD} and criterion value Q_{fuzz} using (2.42). The value of σ_{GD} was given only for P_1 since only P_1 was labeled for reference cells. The results in Tab. 3.5 show that the new method was not only able to detect n_{ref} with good accuracy but n_d as well in the whole P_l. P_4 with n_d, given as an example in Fig. 3.13(b), shows that the new algorithm was able to label n_c automatically based on *a priori* knowledge. P_1 in Fig. 3.13(a) shows demarcation of n_{ref} by an outline and n_d by colored dots. Among these n_c, 73 n_{ref} were found and are shown in Fig. 3.13(a) by segments having both an outline and a colored dot. On the right side of both images in Fig. 3.13(a) and Fig. 3.13(b), a zoomed image was shown for a selected section. It can be seen from Fig. 3.13(b) that n_d are only normal cells while the dying cell states with high mean brightness were not detected. Moreover, a very dull cell near the top right corner of the zoomed image, indicated by an arrow in Fig. 3.13(b), was also not detected. Similar trend can also be observed in Fig. 3.13(a).

	Original images				**Normalized images**	
Images	$Q_{fuzz,norm}$	$Q_{fuzz,seg}$	n_c	σ_{GD}	$Q_{fuzz,norm}$	$Q_{fuzz,seg}$
P_1	0.15	0.73	99	3.95	1	0.75
P_2	0.07	0.76	104	-	1	0.76
P_3	0.08	0.77	80	-	1	0.78
P_4	0.11	0.69	19	-	1	0.73

Table 3.5: Comparison of segmentation results with/without automatic feedback normalization of all P_l images. The results are shown for original images using automatic feedback image segmentation showing $Q_{fuzz,seg}$ values for each image. Moreover, normal cell detection is shown using n_c values for each image. Only n_{ref} for P_1 is given, therefore, its deviation from ground truth is given in terms of σ_{GD}. In the last main column of normalized images, the results are shown to be improved marginally using $Q_{fuzz,seg}$ values after applying automatic shading correction. Moreover, overall normalization criterion values $Q_{fuzz,norm}$ are given for both in presence and absence of automatic shading correction.



(a) P_1 with n_c = 99 (b) P_4 with n_c = 19

Figure 3.13: An example of automatic cell labeling using P_1. Fig. 3.13(a) shows P_1 that includes detection of both reference ground truth normal cells n_{ref} with σ_{GD} = 3.95. White demarcation shows the presence of reference normal cells and central red spot shows the detection of normal cell according to (2.42). Fig. 3.13(b) shows normal cells detection in P_4. White arrow points to a certain cell that did not fulfill the normal cell criteria and was rejected.

3.3.3 Summary and discussion

A framework was presented using partial ground truth to find good algorithmic parameters using feedback structure for image processing steps such as image normalization and image segmentation. When, only total number of cells is given as ground truth, parameters for image segmentation can be adapted to reduce deviation from ground truth in case of Benchmark dataset 4. Moreover, a user can define specific segments to be found in an image abstractly. These abstract reference features were formulated using fuzzy functions to find normal cells in Benchmark dataset 2.

Automatic image normalization was done to remove noise and shading from background. This enables us to perform image segmentation step with the chance to improve results. It was shown from results that feedback-oriented normalization algorithms using fuzzy criteria have the capability to improve image segmentation using a human reference. The results were not remarkably improved as in terms of improvement in image segmentation quality in case of Benchmark dataset 4 but adequately big cells were obtained as required at least qualitatively and cells that were obscured by shading were detected. In Benchmark dataset 2, improvement in segmentation quality was observed as there was more additive microscopic noise.

Moreover, the proposed technique could be efficiently used for the fault detection in large datasets. Based on the automatic feedback-based normalization evaluation criteria proposed, a user can easily discern faulty images in a given image dataset after normalization procedure.

It was shown from results that feedback-oriented algorithms using a fuzzy criterion have the capability to fulfill the goals of segment classification using a human reference. Moreover, the automatic labeling of the whole dataset was performed. This saves a lot of time compared to manual labeling.

3.4 Benchmark dataset 3

This section describes the results of standard image segmentation algorithms when applied to Benchmark dataset 3 in comparison with automatic feedback parametric tuning of same algorithms using fuzzy *a priori* knowledge. Here, *a priori* comprised of same foreknowledge about number and size of the objects to be found in the image as described in Section 3.2. However, this time around the number of objects (i.e. cells) to be found in an image was smaller and fixed. For this benchmark, three different processing steps were used for parametric adaptation namely image normalization, image processing and image classification. Since, the new methodology is based on the judging the segmentation outcome, the aim is first to show the results of image segmentation/classification step using feedforward method in Section 3.4.1 followed by the parameter adaptation for image segmentation step shown in Section 3.4.2. Afterwards, it has been claimed that results from automatic parametric tuning of image segmentation/classification step can further be improved by introducing image normalization. Therefore, first the results from feedforward normalization are shown in Section 3.4.3 in terms of improvement in the automatic image segmentation. Later, the automatic feedback parameter tuning of image normalization parameters is shown in Section 3.4.4.

3.4.1 Standard image segmentation/classification routines

Same segmentation methods (i.e. Otsu thresholding and edge detection used in Section 3.2.1) were chosen to show the effect of increasing artifact level on segmentation outcome. Using Benchmark dataset 3, the results of simulated human cell segmentation from above mentioned non-adaptive algorithms were compared with their corresponding parameter adaptation. Evaluation criteria were developed based on *a priori* knowledge about the objects (i.e. simulated human cells) to be found in the Benchmark dataset 3. This knowledge, for simplicity, was based on error in terms of total number of cells and types of cell states detected and all error metrics were same as described in Section 3.2.1.

Exemplary processing pipeline: The processing pipeline given in Fig. 3.1 was used to apply the aforementioned methods and thus to compare two or more standard subroutines in an image processing pipeline. There were some changes made in the morphological operators used in the pipeline. Although, the number of operators and their sequence remained unchanged, the parameters of some operators were changed using both segmentation methods. In case of Otsu's method an image dilation with structuring element having a window size of 2×2 pixels was used whereas the radius of disk used as a structuring element for image opening remained same with the value of 3. For Sobel edge detection, image dilation was carried with same parameter setting but for image opening disk of radius 9 was used. High disk radius was selected in order to attempt to remove false BLOBs created due to dilation salt and pepper noise. Moreover, at the classifier end, only the number of input features was changed. The only difference for this dataset was that only two features (i.e. maximum intensity and standard deviation of the detected cell nuclei) were used to distinguish between two cell states. These features are selected qualitatively based on expert knowledge without any quantitative abstract reference. The aforementioned parameters are also shown in Tab. 3.6.

Segmentation outcome: The performance of the two pipelines (i.e. using Otsu's method vs. using Sobel edge detector) with respect to increasing artifact levels $A(n_1,n_2,n_3)$ are given in Fig. 3.14. The segmentation evaluation measures are integrated in total quality measure $Q(n_1,n_2,n_3)$ given in (2.49). As shown in Fig. 3.14, a gradual decrease in the quality of segmentation outcome indicated by $Q(n_1,n_2,n_3)$ with increasing artifact levels ($A(n_1,n_2,n_3)$) is observed for both segmentation methods. To quantify the performance outcome, the robustness measures R_{otsu} and R_{edge} for Otsu thresholding and Sobel edge detection, respectively are given. Otsu's method outperforms Sobel edge detection in such a scenario. At higher $A(n_1,n_2,n_3)$, edge detection fails to produce any meaningful results whereas Otsu's thresholding method shows a gradually decreasing trend in $Q(n_1,n_2,n_3)$ with the increase in $A(n_1,n_2,n_3)$. The robustness results of segmentation methods for this pipeline and the parameters involved are shown later in Tab. 3.7.

3.4.2 Feedback-based parametric tuning for standard image segmentation/classification routines

In this section, same segmentation methods i.e. Otsu thresholding and edge detection that were used in Section 3.4.1 were used. The parameters involved in both methods were adapted iteratively in a feedback fashion affecting the outcome of segmentation. The best parameters were selected based on a set quality criterion (2.49) and segmentation based on these optimal parameters was selected for final segmentation. Here, not only both selected methods were compared with each other using parameter adaptation, but also a comparison with their standard usage when no feedback or parameter adaptation is given.

The exemplary pipeline for the application of automatic feedback parametric tuning is shown before in Fig. 3.1. Parameters for morphological operators and pre- and post-processing of images remains nearly identical in both cases (i.e. with and without feedback). As for the parameter adaptation of segmentation methods in Benchmark dataset 1, only one parameter was adapted here in Benchmark dataset 3 for each method to show the proof of principle.

For Otsu's method, intensity threshold t_{otsu} was adapted over a parameter variation from 0.2 to 0.3 with the resolution of 0.01. These limits of t_{otsu} can be chosen based on t_{otsu} values from standard application of Otsu shown given in Section 3.4.1. Similarly, again only one parameter (i.e. threshold t_{sobel} for thresholding the calculated gradient magnitude of image intensity) for Sobel edge detection was adapted for a fair comparison with non-feedback methods. t_{sobel} was adapted between 0 and 0.2 using knowledge from standard application of Sobel edge detection and step size for search was selected to be 0.01.

This procedure for parameter adaptation can be implemented using different pipeline structures and more parameters could also be involved. Reduction of the parameter vector **p** just to one parameter was done for both segmentation methods such that **p** = t_{otsu} for Otsu's method and **p** = t_{sobel} for Sobel edge detection. Parameter adaptation was done according to (2.50) based on (2.49). The difference in parameters involved and pipeline structure compared to feedforward implementation is shown in Tab. 3.6. The routine for this has been developed as macro in the Gait-CAD software ([136]).

Algorithm type	Parameter/Operator name	Parameter setting	Processing step	Range of tuned parameters (min : step size : max)
1,2,3,4	image filling	'holes'	Morphological operations	
1,2,3,4	image dilation	SE: 2x2 ones	Morphological operations	
1,3	image opening	SE: s = 3 'disk'	Morphological operations	
2,4	image opening	SE: s = 9 'disk'	Morphological operations	
1,3	t_{otsu}	fixed for 1	Thresholding	0.15 : 0.01 : 0.35
2,4	t_{sobel}	fixed for 2	Edge detection	0 : 0.01 : 0.2
1,2,3,4	number of features	2	Feature extraction	

Table 3.6: Fixed and optimized parameters for each processing step in Benchmark dataset 3 without using normalization step. Here, 1: StdOtsu, 2: StdEdge, 3: AutoOtsu, 4: AutoEdge. 3 and 4 are the new feedback-based automatic parameter tuning methods.

Segmentation outcome: Using image processing pipeline given in Fig. 3.1, selected parameter for each method was adapted iteratively using selected parameter space. This search for optimal parameters can be further refined by decreasing the step size of parameter iteration. The standard feedforward application of both segmentation methods delivered robustness values of R_{otsu} = 0.80 for Otsu's method and R_{edge} = 0.19 for Sobel edge detection respectively. This performance can be improved by using feedback parameter adaptation and is evident if the robustness values are better than aforementioned values of standard application. The performance using parameter adaptation for both methods with respect to increasing artifact levels in comparison to their standard feedforward application is shown in Fig. 3.14.

Fig. 3.14 shows a gradually decreasing trend for performances of the segmentation algorithms indicated by $Q(n_1,n_2,n_3)$ with increasing artifact levels $A(n_1,n_2,n_3)$ in case of feedforward application of both segmentation methods. The performance for algorithms using feedback parameter adaptation is shown by robustness measures $R_{opt,otsu}$ and $R_{opt,edge}$ for Otsu thresholding and So-

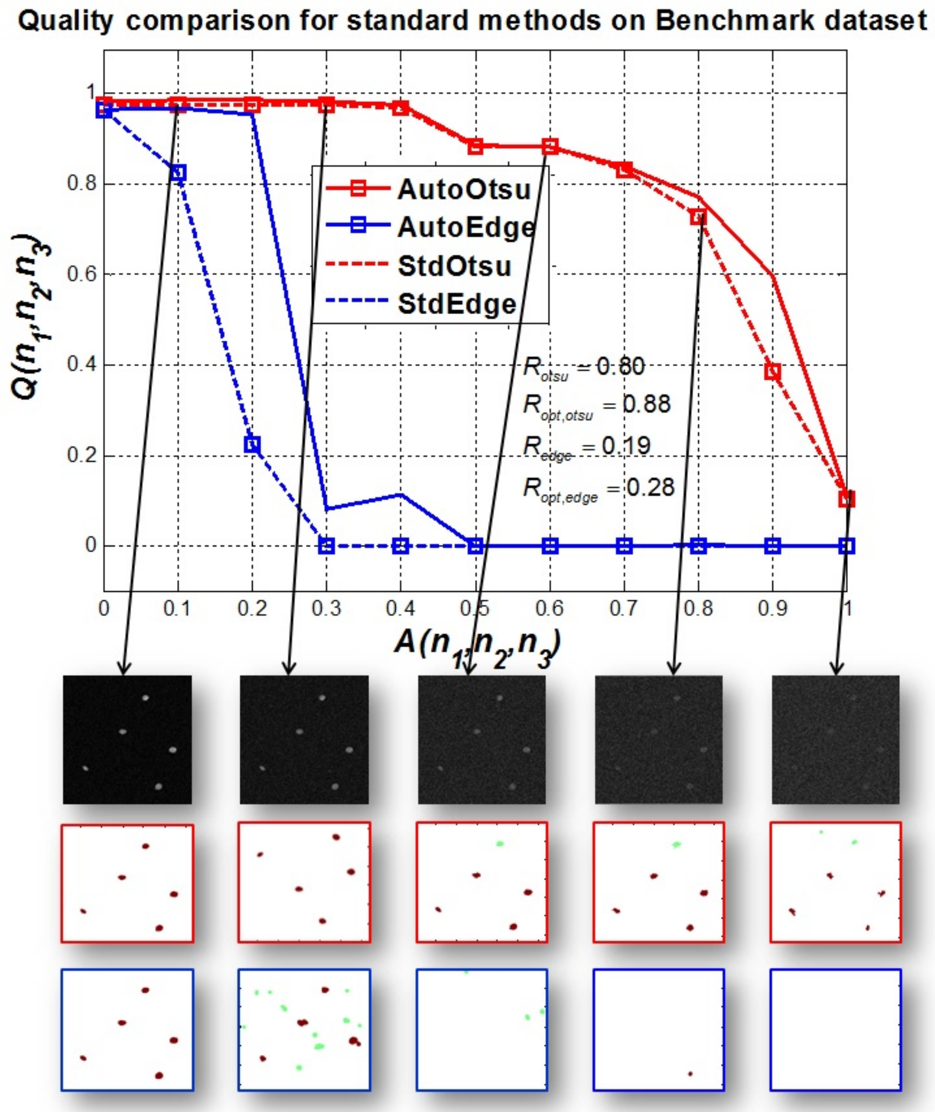


Figure 3.14: Results of parameter adaptation for image segmentation on Benchmark dataset 3. $Q(n_1,n_2,n_3)$ vs. $A(n_1,n_2,n_3)$. The first row indicates original images from Benchmark dataset 3. The second row shows segmentation and classification results using parametric feedback tuning of Otsu's method. The third row shows segmentation and classification results using similar tuning of edge detection method. Colors represent classification success where red is for correct outcome and green for incorrect. Robustness values for Otsu thresholding and Sobel edge detection are R_{otsu} = 0.80 and R_{edge} = 0.19 respectively. Where as, robustness values for automatic feedback parametric tuning of aforementioned methods are $R_{opt,otsu}$ = 0.88 and $R_{opt,edge}$ = 0.28 respectively.

bel edge detection respectively. Larger *R* values correspond to high robustness of given methods. It is observed that performances of both methods are improved as the area under the curve for feedback application is greater than that of standard application of algorithms.

Otsu's method, which performed quite well in its standard application, can further be improved using adaptation of its threshold parameter t_{otsu}. This is evident by its robustness value i.e. $R_{opt,otsu}$ = 0.88 showing a 10% improvement from its standard usage. Especially, at higher noise levels, it still delivers many cells that are correctly segmented and classified. It may be noted that noise level can further be increased to make even the parameter adaptation fail. The concern here is not to show the limitations of feedback parameter adaptation but to emphasize the fact that at artifact levels where standard Otsu's method fails, threshold adaptation can deliver very good results. Moreover, it is also observed that, using parameter adaptation, the result at any noise level for Otsu's method cannot be worse than that of its standard feedforward usage. At low noise levels, both implementations are close to each other, whereby the new parameter adaptation method edges slightly in terms of quality values as shown in first two images in first row of Fig. 3.15. No drastic change can be observed. The difference in quality values are only due to a slightly better delineation of cells in the case of feedback parameter adaptation. As noise level is increased to about 0.8, a more identifiable improvement is observed as shown in first two images in second row of Fig. 3.15. There is one misclassified cell in both cases as shown by the decrease in $Q(n_1,n_2,n_3)$ values at $A(n_1,n_2,n_3)$ = 0.8. However, in terms of segmentation, the cell in the middle of first two image in second row of Fig. 3.15 was delineated well in case of feedback implementation of Otsu's method.

Sobel edge detection, that performed poorly in its standard feedforward usage, can also be improved using parameter adaptation of t_{sobel}. The improvement in segmentation outcome was increased by 47% indicated by robustness value $R_{opt,edge}$ = 0.28. Due to certain restrictions in parameters as described in Section 3.4.1 (results of edge detection cannot be constructively improved when image dilation and hole filling is involved with high salt and pepper noise), the feedback adaptation also fails to deliver any interpretable results at higher noise levels. Still, for no image present in the whole dataset could the segmentation quality using new feedback-based methods could be worse than what is achievable in the standard feedforward application. The performance of edge detection in feedback parameter adaptation in comparison to its feedforward usage can be seen in last two images of first row in Fig. 3.15.

It is clear from last two images of first row in Fig. 3.15, that two cells were not segmented that could have been obtained at a different threshold value. However, both implementations of Sobel edge detection fail at high noise levels since no meaningful objects are detected as shown in last two images of second row in Fig. 3.15. Only a part of cell present at the bottom of last two images in second row of Fig. 3.15 was detected due to which a certain value of $Q(n_1,n_2,n_3)$ was obtained in feedback parameter adaptation. Wrong classification assignments occur only at extremely high artifact level for Otsu's method and is more efficient at higher artifact levels in comparison to Sobel edge detector. High noise levels of different types make it difficult for Sobel edge detection to produce any meaningful result because background noise is translated to such a level by the usage of morphological operators that noise is also seen as BLOBs or more accurately cells. The overall results for different pipeline structure for each segmentation method is accumulated in Tab. 3.7.

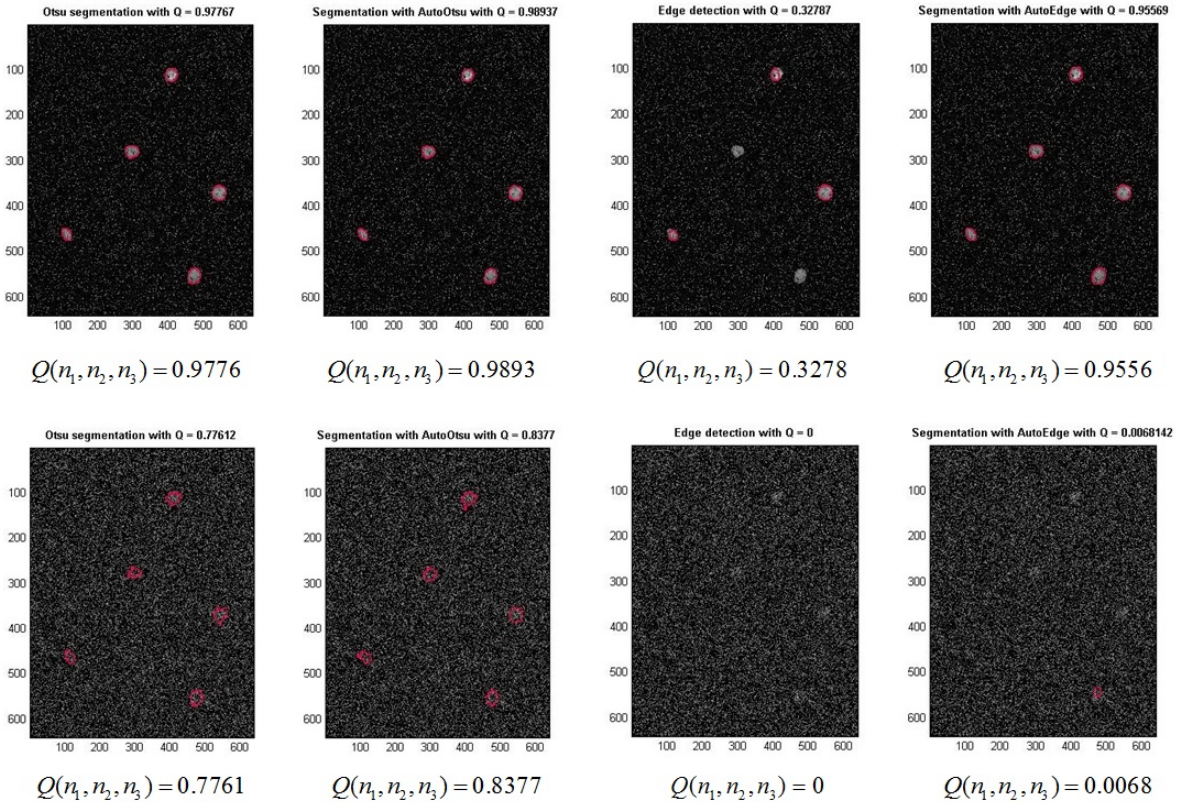


Figure 3.15: Segmentation outcome: *Q*(n_1,n_2,n_3) vs *A*(n_1,n_2,n_3) at 0.2 and 0.8. First row: segmentation results at *A*(n_1,n_2,n_3) = 0.2. Second row: segmentation results at *A*(n_1,n_2,n_3) = 0.8. First and second column: Otsu's method using standard feedforward and feedback application respectively. Third and fourth column: Sobel edge detection method using standard feedforward and feedback application respectively. Red outlines in all images show the segmentation outcome.

3.4.3 Standard image normalization/segmentation/classification routines

The results given in Sections 3.4.1 and 3.4.2 for Benchmark dataset 3 were only based on the automatic tuning of image segmentation parameters. Since the images contain increasing systematic noise, poor segmentation results are obtained at high artifact levels without tuning the parameters. By tuning the parameters for image segmentation, improved results were observed (see Fig. 3.14). Still, at higher artifact levels edge detection does not deliver meaningful results.

This can be overcome by removing noise from images. This was done (apart from normalization with quantiles method described in Section 3.2.1) by applying image normalization method described in Section 2.3.2. Here, the *apriori* knowledge was just based on the binary ground truth image. This was done to improve the image segmentation/classification result. Therefore, the resulting normalized image would be applied to feedback pipeline structure of image segmentation/classification.

Pipeline structure	Segmentation method	Fixed parameters	Optimized parameters	Robustness values R	Result links
Feedforward (solid lines) Fig. 3.1	Otsu	w,s,t_{otsu}	-	0.80	Red dashed line in Fig. 3.14
	Edge	w,s,t_{sobel}	-	0.19	Blue dashed line in Fig. 3.14
Feedback (dashed lines) Fig. 3.1	Otsu	w,s	t_{otsu}	0.88	Red solid line in Fig. 3.14
	Edge	w,s	t_{sobel}	0.28	Red solid line in Fig. 3.14

Table 3.7: Results showing robustness of feedforward vs. feedback pipeline structures (segmentation/-classification) for two different image segmentation methods in Benchmark dataset 3.

Exemplary processing pipeline: An exemplary pipeline for image normalization given in Fig. 3.16 was implemented to apply the methods given in Section 2.3.2. This pipeline was first used in feedforward structure shown by solid black lines in Fig. 3.16.

In this pipeline, a grayscale raw image is fed to a median filter with a symmetric window size of $w \times w$. The resulting image is treated with image opening with a fixed size s of the disk used as a structuring element. Parameter s is fixed at the value of 3 and w is fixed at 3. The effect of normalization on the image is shown by passing this normalized image to feedback structure of image segmentation pipeline given in Fig. 3.1 and looking at the results. The parameters[4] used in pipeline are same as given in Tab. 3.6.

Segmentation outcome: Using feedforward image processing pipeline given in Fig. 3.16, image normalization was performed. The results are given in terms of improvements found in image segmentation outcome using normalized image for Sobel edge detection. This is shown by robustness values (see Fig. 3.17) of Sobel edge detection method ($R_{edge,norm}$ = 0.82) which is a remarkable increase from robustness ($R_{edge,norm}$ = 0.28) obtained from using feedback segmentation only. Robustness value of $R_{otsu,norm}$ = 0.80 for Otsu's method was seen to be a bit lower than the value ($R_{opt,otsu}$ = 0.88) obtained using only feedback segmentation. This is due to the fact that at medium artifact levels median filter tends to produce some dull back ground objects that are detected as objects using Otsu thresholding. However, the aim here is to show the difference between results produced by using fixed normalization parameters and parameter tuning. The collective improvement by using image normalization step in the pipeline is shown later in Section 3.4.4 for both methods.

[4] Only one parameter for this feedback-based pipeline was changed i.e. $s = 3$ instead of $s = 2$. Therefore, a 3×3 SE was used for image dilation.

3.4.4 Feedback-based parametric tuning for image normalization/segmentation/classification routines

The image noise removal done in Section 3.4.3 was based on fixed parameters. This can still be improved by selecting parameters involved in image normalization which may deliver good results with a different value when high noise is present. Therefore, main idea was to target these parameters carefully and select an efficient range over which these parameters can be varied. The best parameter set can then be selected based on the quality criteria. Criteria were defined based on the difference of pixel values between the background and objects of ground truth image and normalized image at same pixel positions.

Exemplary processing pipeline: A general example of a pipeline used for image normalization in a feedback fashion shown by dashed lines in Fig. 3.16 was developed to show the application of methods given in Section 2.3.2.

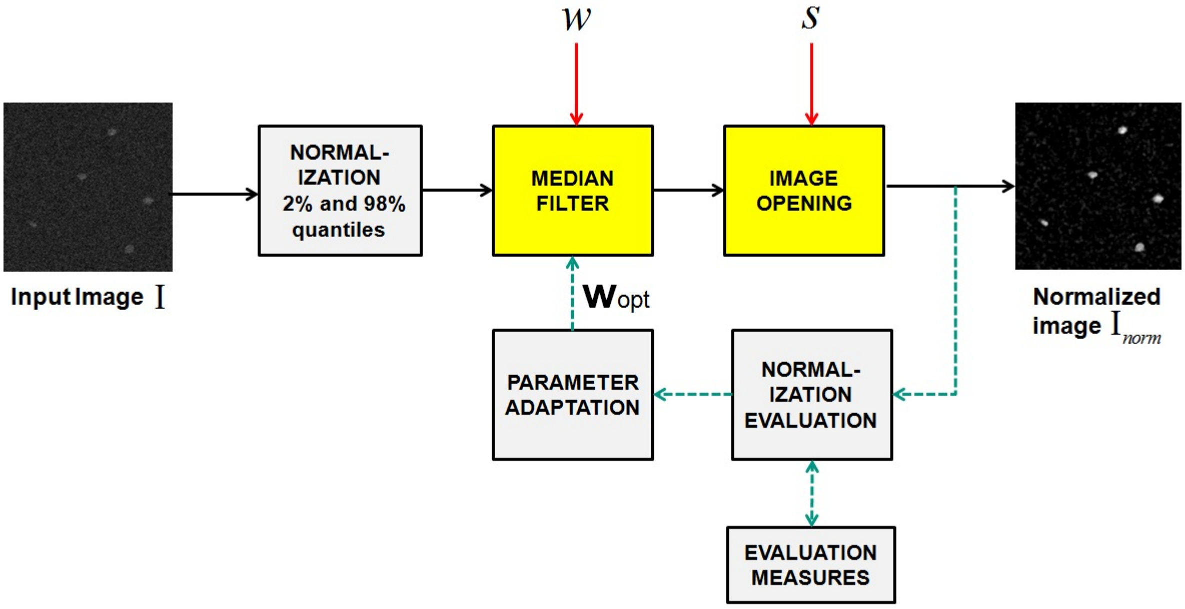


Figure 3.16: Exemplary pipeline for the parameter adaptation for normalization step for images of Benchmark dataset 3.

In this feedback pipeline structure, a parameter set can be selected to be adapted iteratively based on evaluation criterion of normalization. To keep it more simple and comprehensible, only one parameter (i.e. w) was adapted to show the effect of parameter adaptation on normalization outcome. Parameter s was fixed at the value of 3 as in feedforward case and w (which is now $\mathbf{p}$) is varied from 3 to 7 with a step of 1[5].

[5] w defines a filter window such that $m \times g = w \times w$. If w is even integer then the image pixel overlapping the top-left pixel of filter window is affected. Conversely, if w is odd integer then the image pixel that overlaps the

Consequently, $\mathbf{p}_{opt}$ is selected based on the evaluation measure given in (2.25). The optimal normalized image $\mathbf{I}_{norm}$ is calculated based on $\mathbf{p}_{opt}$. The normalization effect in the pipeline is studied by using the resulting image in feedback structure of pipeline structures given in Fig. 3.1 and comparing the results.

Segmentation outcome: The search for optimal parameters can be performed using a different or extended search space but careful selection of parameter range is required for user. Normalized image is fed to feedback pipeline structure given in Fig. 3.1. The results are shown in terms of improvements found in image segmentation outcome using normalized image. This is shown by improvement in robustness values (see Fig. 3.17) using feedback normalization step for both segmentation methods. Performances are seen to be improved by using feedback parameter adaptation in Fig. 3.17. The automatic selection of **p** is illustrated by few examples in Fig. 3.18.

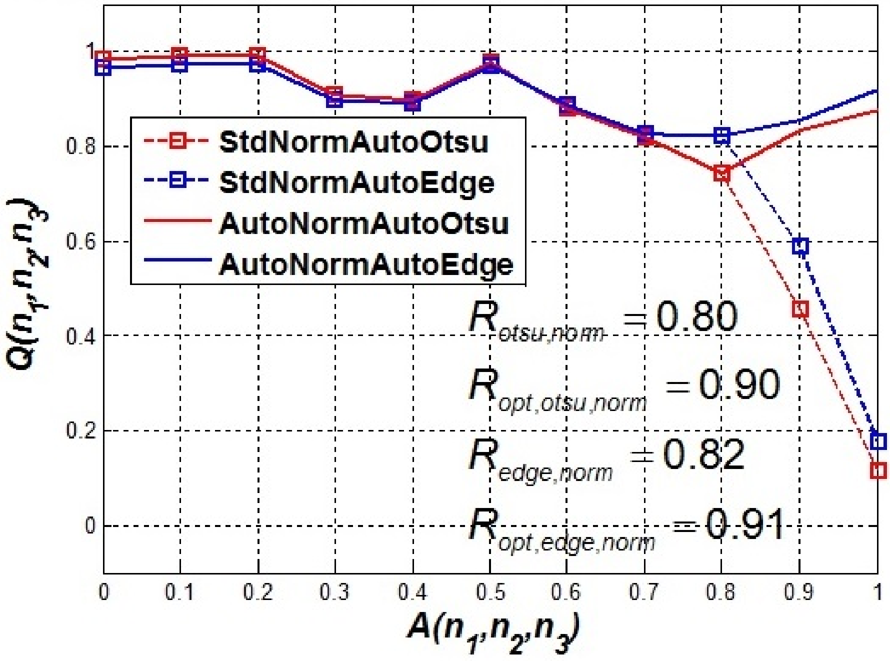


Figure 3.17: Results of parameter adaptation of image normalization for Benchmark dataset 3. *Q*(n_1,n_2,n_3) vs. *A*(n_1,n_2,n_3). Robustness values for Otsu thresholding and Sobel edge detection are $R_{otsu,norm}$ = 0.80 and R_{edge} = 0.82 respectively when feedforward image normalization is used shown but dashed line in image. Where as, robustness values for segmentation methods using parametric tuning of image normalization step are $R_{opt,otsu}$ = 0.90 and $R_{opt,edge}$ = 0.91. Feedback-based quality for both methods are shown by solid lines.

Fig. 3.17 shows a notable increase in the performance of both new methods at very high artifact levels (i.e. *A*(n_1,n_2,n_3) > 0.8). This is also indicated by their corresponding robustness values $R_{opt,otsu,norm}$ and $R_{opt,edge,norm}$ for Otsu thresholding and Sobel edge detection respectively using the new feedback-based method. It is observed that outcomes of both methods are generally

middle pixel of filter window is affected.

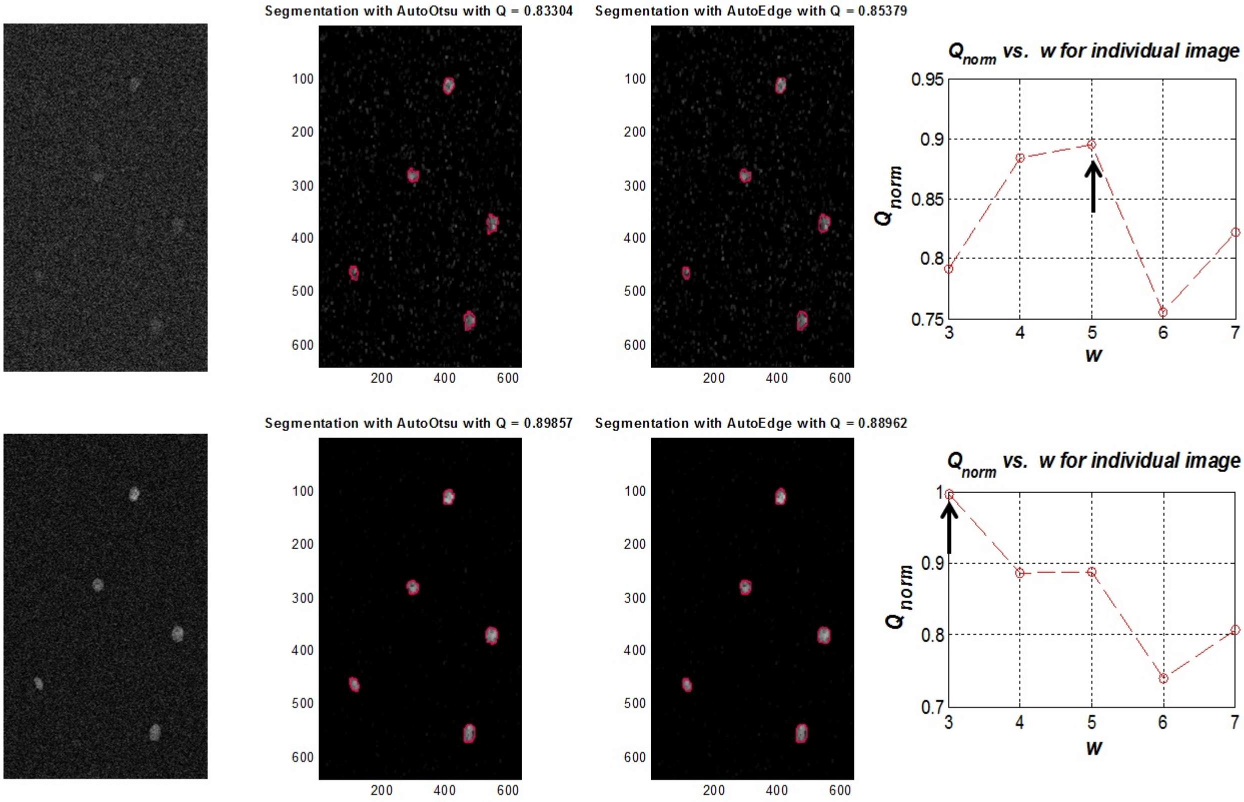


Figure 3.18: Examples for parameter adaptation for normalization step for images of Benchmark dataset 3 at different artifact levels. First row: Results at $A(n_1,n_2,n_3)$ = 0.9. Second row: Results at $A(n_1,n_2,n_3)$ = 0.4. First column: Original Benchmark image. Second column: Otsu segmentation result after automatic normalization. Third column: Sobel edge detection result after automatic normalization. Fourth column: Plot showing selection of $\mathbf{p}_{opt} = w_{opt}$ against Q_{norm} indicated by upright black arrow.

improved in comparison to the absence of image normalization. It can be noted that with new feedback application of normalization using only one parameter adaptation was better in terms of $Q(n_1,n_2,n_3)$ for both segmentation methods compared to cases where only fixed values of normalization parameters were used.

Fig. 3.18 shows that the new feedback technique was able to select w automatically at different artifact levels. This helped in improving the $Q(n_1,n_2,n_3)$ at individual $A(n_1,n_2,n_3)$ levels.

3.4.5 Summary and discussion

The presented parameter adaptation of image processing algorithms was shown using a benchmark dataset containing systematic artificial noise and only two object types for simplicity and the proof of principle. Results were presented for different image processing steps by using two

Pipeline structure	Segmentation method	Fixed normalization parameters	Optimized parameters	Robustness values R	Result links
Feedforward (solid lines) Fig. 3.16	AutoOtsu	$w = 3, s = 3$	-	0.80	Red dashed line in Fig. 3.17
	AutoEdge	$w = 3, s = 3$	-	0.82	Blue dashed line in Fig. 3.17
Feedback (dashed lines) Fig. 3.16	AutoOtsu	$s = 3$	$w = 3{:}1{:}7$	0.89	Red solid line in Fig. 3.17
	AutoEdge	$s = 3$	$w = 3{:}1{:}7$	0.91	Red solid line in Fig. 3.17

Table 3.8: Table showing normalization parameters and robustness of feedforward vs. feedback pipeline structures of normalization step for Benchmark dataset 3. The pipelines used for application are: a) feedforward normalization + feedback segmentation/classification b) feedback normalization + feedback segmentation/classification using two separate image segmentation methods at the segmentation step. Here, parameters are given just for the normalization step. Optimized normalization parameter is written in such a way that $s = \min : stepsize : \max$. Parameter for segmentation methods remain same as previous pipeline given in Fig. 3.1.

standard image segmentation algorithms for comparison.

At first, image segmentation step was used to show the difference between standard application of algorithms and the new parameter adaptation method. It was shown that improvements can be made if involved parameters for each step are adapted iteratively. These results were shown to be robust at higher artifact levels where standard feedforward application fails to produce adequately desirable results.

Secondly, a new method using parameter adaptation of image normalization was shown using Benchmark dataset 3 containing high systematic noise such as Gaussian, salt and pepper and speckle. A comparison was presented in terms of how much could a segmentation outcome be improved if image normalization is used with fixed parameters and later adapting those automatically. An improvement could be seen in segmentation results when feedforward normalization was used. This was further improved by adapting a single parameter of image normalization. The feedforward normalization may not improve the results obtained without using normalization step. This is due to that fact that different pre- and post-processing parameters could deteriorate the result at a certain artifact level. To cope with this, local parameters for each artifact level can be extracted.

Furthermore, the proposed method of feedback parameter adaptation for image normalization can be further fine-tuned by involving more parameter or selection between different filter types. Noise removal techniques can be made to adapt parameters in the presence of sequentially increasing noise. The results could be improved if careful parameter selection and their range is selected by the user. The quality criteria were only given as an example here. More sophisticated and tailor-made criteria can be defined by a user according to his needs.

4 Applications - cell state analysis in human cancer cells

4.1 Introduction

Image processing is a powerful analysis tool encompassing diverse application domains (e.g., in industrial quality inspection, automobile industry, mobile robots, biomedical, toxicology etc.). Nowadays, the focus of using image processing techniques and methods in order to develop adequate comprehension of biological processes has indeed become quite alluring, both to cell biologists and toxicologists. The acquired image datasets are challenging for analysis due to varying background noise, amount of information and drifting acquisition conditions. Adaptive image processing and feedback methods are useful concepts in tackling such kind of datasets.

In this chapter, the new feedback-based methods for image normalization and segmentation described in Chapter 2 are applied to a real biological cells dataset (i.e. Application dataset[1]). This is to show that the applied algorithms are able to perform on non-synthetic real image datasets.

This is essentially a heterogeneous biological dataset consisting of images showing human lung cells (A549) treated with the anticancer drug cis-platin for 24 hours and representative images were acquired as described in [30]. The image data is taken at a single time point whereas there are four images for each of the well that contains the cells exhibiting these processes and there are 96 wells in total. There are three channels for obtaining this data i.e., bright field (BF) which allows the detection of morphological changes, Hoechst 33342 (that gives information about nuclei or better to say DNA as it is a fluorescent dye binding to DNA) and Propidium Iodide (PI) that gives subtle indication of whether a late apoptotic or necrotic cell has been detected.

Therefore, different object types are available that can be distinguished from each other using diverse features from different channels. It also presents a dataset where information fusion and robust feature selection is of great significance.

[1] The main part of the biological dataset - 304 images in one channel and consisting of 3 channels. The whole dataset contains 314 images in one channel, out of which 10 images/channel are used as Benchmark dataset 2 and the rest of images are used for application purposes as Application dataset.

4.2 Biological background

4.2.1 Apoptosis

One such process that occurs in multicellular organisms is apoptosis and is currently a field of enormous attention in biomedical research [50]. Apoptosis is described as a genetically regulated form of cell death characterized by biochemical events leading to characteristic cell changes such as blebbing, cell shrinkage, nuclear fragmentation, chromatin condensation and DNA fragmentation [33]. It has its share in biological processes such as embryogenesis, physiological involution, atrophy of various tissue and many other processes pertaining to diseases. Evasion of apoptosis is the hallmark of cancer. Virtually all cancer cells contain mutations that enable evasion of apoptosis through a variety of mechanisms [56]. In cell culture, apoptosis limits the yield of economically and medically important products, and can result in synthesis of imperfect molecules [112].

Mutation is one trigger for the apoptosis process. Nevertheless, the apoptosis is a normal component of the health and development of multicellular organisms whereby cell death occurs in a controlled and regulated fashion. It occurs during the normal development of multicellular organisms and is continued in later stages of life. In combination with cell proliferation, apoptosis contributes for shaping tissues and organs in developing embryos. Apoptosis is also an important part of regulation of the immune system [11].

4.2.2 Necrosis

In contrast, another type of cell death used in conjunction with the apoptosis is necrosis that is considered almost always detrimental. It is a premature cell death having an inflammatory response in an unplanned fashion. It is usually caused by the external circumstances such as exposure to the toxic chemicals [47]. At the morphological level, necrosis is characterized by cell swelling rather than cell shrinkage as seen in apoptosis [112].

4.2.3 Mitosis

Moreover, another noteworthy process here is mitosis, which is the cell division resulting into two daughter cells that are genetically identical to their parent and to each other. It is of high significance for a living organism since it enables cells to reproduce and regenerate tissues in the body.

4.3 Biological *a priori* knowledge and desired features for image analysis

The aim of the project is to quantify the number of cells for each cell state in the whole dataset. From a viewpoint of image processing and analysis, the image data collected from these observed processes has an implicit potential as in the recovery of desired features enabling an efficient quantitative analysis of these captured processes. However, in order to facilitate such an analysis sufficient amount of valid data and pre-knowledge of judgment criteria for classification and feature extraction is necessary.

The three processes described above have both explicit and subtle differences. In the case under consideration, the image data is taken at a single time point whereas there are four images for each of the wells that contains the cells exhibiting these processes and there are 96 wells in total. There are three channels for obtaining this data i.e., bright field (BF) which allows the detection of morphological changes, Hoechst 33342 (that gives information about nuclei or better to say DNA as it is a fluorescent dye binding to DNA) and Propidium Iodide (PI) that gives subtle indication of whether an apoptotic (later stage) or necrotic cell state has been detected. PI is used to stain cells since it binds to nucleic acids giving them fluorescence. This is illustrated in Fig. 4.1.

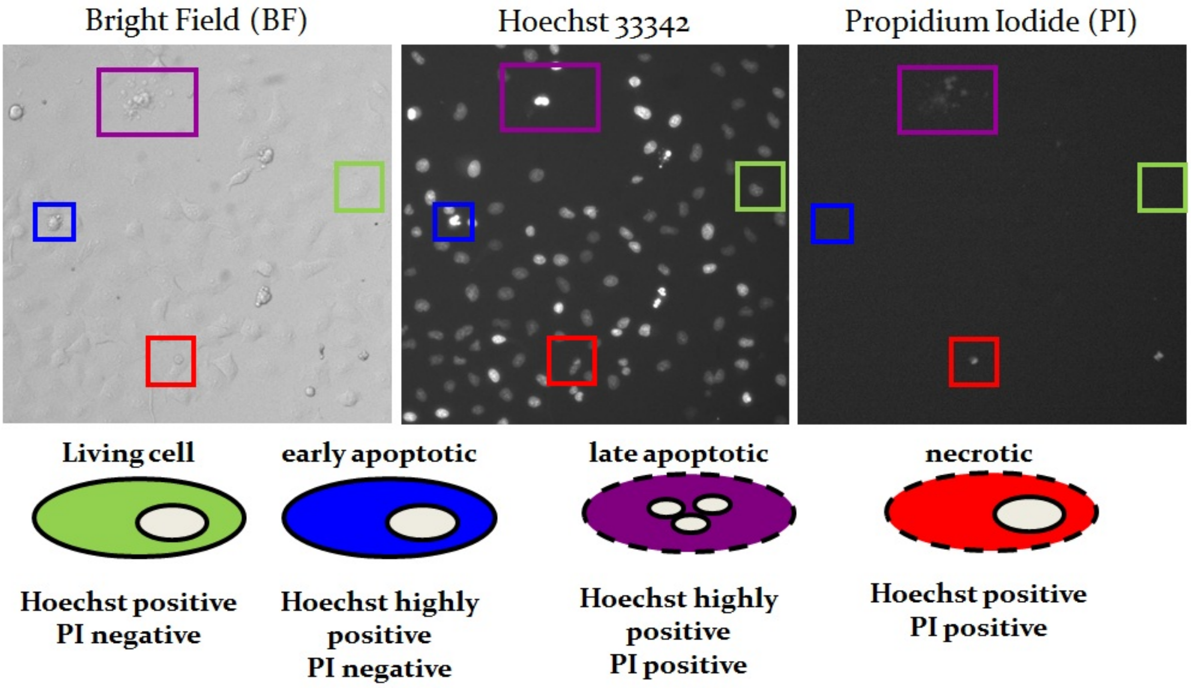


Figure 4.1: Description of different cell phenomena using different channels.

The morphological process of cell nuclei in the apoptosis can be described stepwise in a chronological order as:

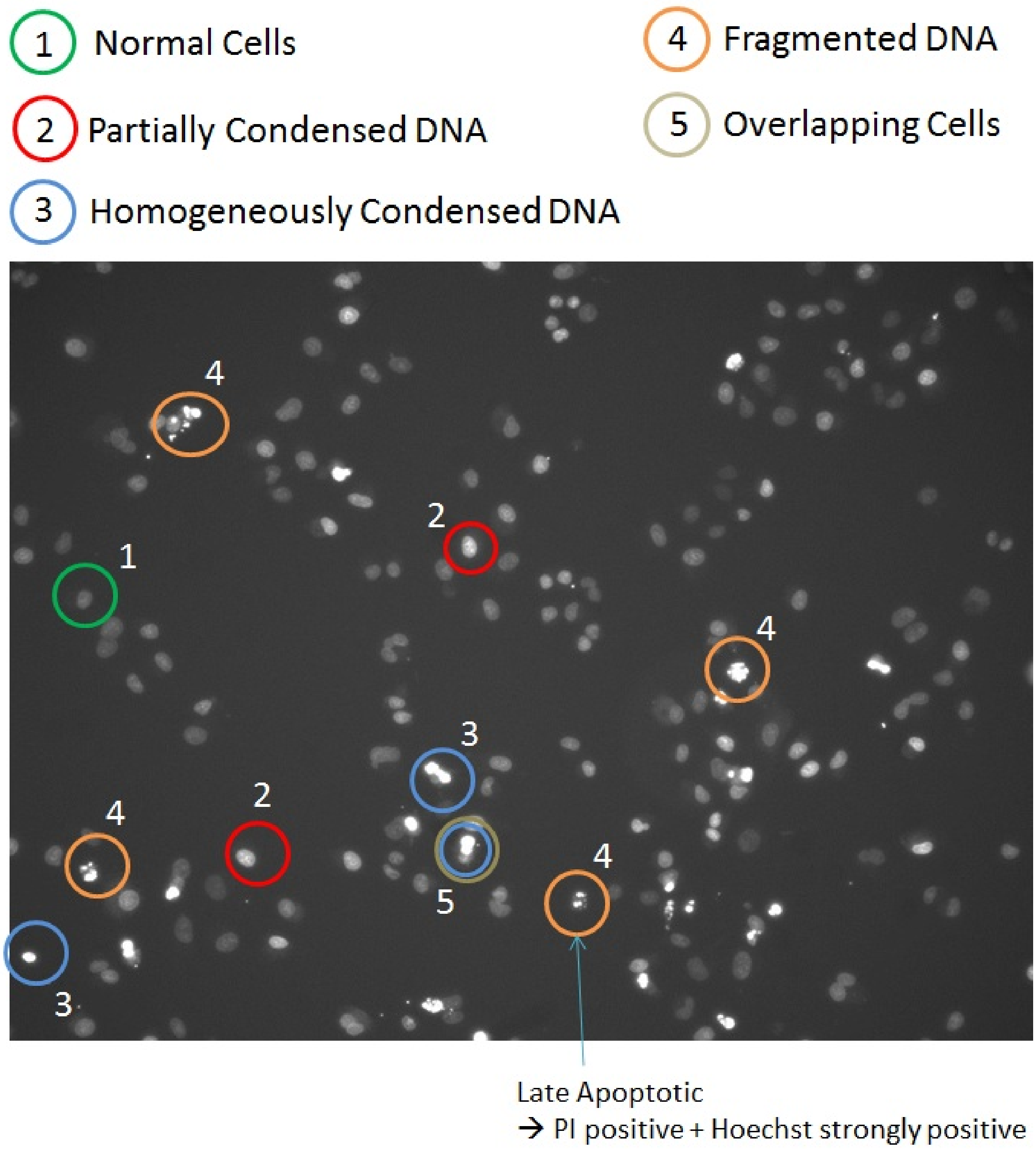

1 Normal Cells
2 Partially Condensed DNA
3 Homogeneously Condensed DNA
4 Fragmented DNA
5 Overlapping Cells
4
2
1
4
3
4
2
5
4
3
Late Apoptotic
→ PI positive + Hoechst strongly positive

Figure 4.2: Different cell states using Hoechst channel.

- On the nuclear level in the first step at the beginning of the apoptosis, there occurs a partial condensation (shown as 2 in Fig. 4.2) of the DNA in the nucleus (i.e. packaging of DNA) that is followed by a homogeneous condensation (shown as 3 in Fig. 4.2) in the subsequent step as shown in Fig. 4.2. The nucleus with partially condensed DNA is slightly bigger and more oval and has bright distinguishable spots in contrast to the one with homogeneous condensation which is round and show a homogeneous bright Hoechst signal. A partially condensed DNA is seen as high fluorescence distributed unevenly inside a normal nucleus in the Hoechst channel as shown by number 2 in Fig. 4.2. A normal nucleus is almost homogeneous and dull in terms of intensity with very low fluorescence in the Hoechst channel (see number 1 in Fig. 4.2). It is difficult to recognize in the BF channel and has no signal in PI channel.

- Next step shows early apoptosis indicated by homogeneous condensation of DNA as nucleus is more bright, round and homogeneously condensed in the Hoechst channel as shown in the Hoechst channel image (contained in blue rectangular box) of Fig. 4.1. In the BF channel it is round as can be seen in BF channel (contained in blue rectangular box) image in Fig. 4.1. It is probably lesser round in BF channel than cell states showing mitosis phenomenon.
- Finally, in the last step, a late apoptotic nucleus or fragmentation of DNA (indicated by number 4 in Fig. 4.2) may be detected based on the presence of signal from PI channel. A fragmented cell is shown in the Hoechst channel by many small independently distinguishable highly fluorescent spots describing the fragmentation of the nucleus in encirclement indicated by number 4 in Fig. 4.2. In the case of late apoptosis, a signal from PI channel is present as can be seen from red box in Fig. 4.1.

The different states of nuclei in the Hoechst channel can be seen in Fig. 4.2, which shows normal healthy nucleus against the different states of nuclei during different processes whereby the difference between their intensity values, areas and shapes can be visually observed. Moreover, using separate channel information, these states are shown in Fig. 4.3.

During the earliest stage of mitosis, an elongated and bright nucleus is detected in the Hoechst channel whereas in the BF channel it appears round. Ultimately, the division of nucleus into two followed by the division of cell into two daughter cells are detected that are also bright in the Hoechst channel which become less bright in later stages. Furthermore, it is also necessary to detect necrotic cells in additions to the normal cells, apoptotic cells and cells produced by mitosis. For that, presence of signal from PI channel can be used to indicate necrotic cells as there is a loss of integrity of cell membrane since the signal acquired from Hoechst channel is less bright than early apoptotic cells.

Moreover, the phenomenon of overlapping of cells and the number of apoptotic cells in their neighborhood as shown in the Fig. 4.4, are also subject to further investigation using image processing techniques. The overlapping may occur in the early or late apoptosis stage.

Some challenges in image processing involves the detection of fragmented nucleus as belonging to one cell and clear demarcation of the boundaries of nuclei extremely close to each other. Another problem is that in the late apoptosis the nucleus may not be completely fragmented into distinguishable separate fragments rather it could be just one irregularly shaped object with high fluorescence.

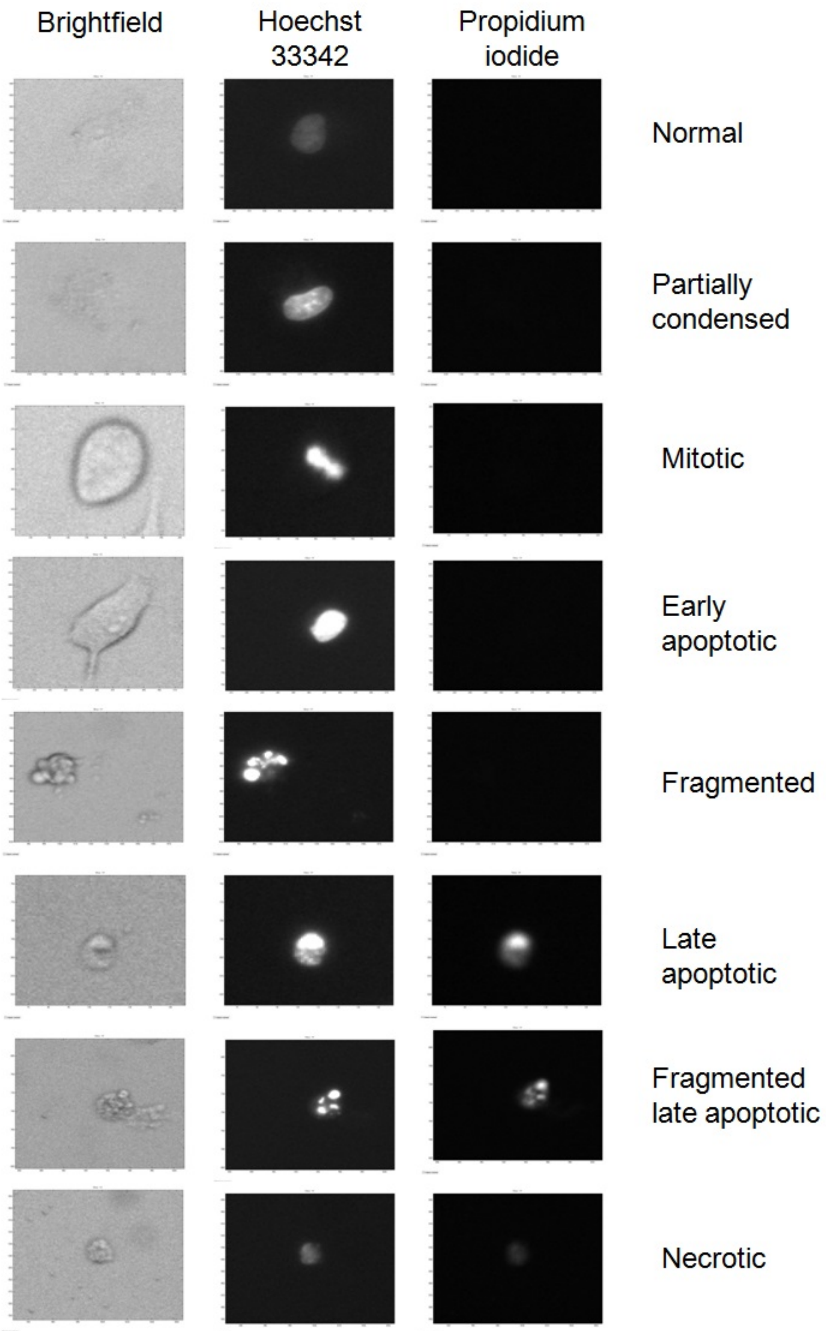


Figure 4.3: Typical example of each cell state using different channels. Color code: Green - normal cells, red - partially condensed DNA, blue - homogeneously condensed DNA, orange - fragmented DNA and gray - overlapping cells.

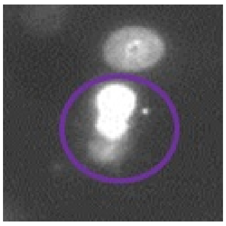


Figure 4.4: Overlapping of two cells.

4.4 Goals of image analysis

The aim here is to establish an image processing routine to classify apoptotic cell states in addition to other processes like mitosis and necrosis. The desired functionality of the proposed image analysis scheme involves:

- Detection of total number of cells in each of the Hoechst channel images.
- Classification of cell types and DNA states:
    1. Apoptotic
        - Partially condensed
        - Early apoptotic
        - Late apoptotic
        - Fragmented
        - Fragmented late apoptotic
    2. Necrotic
    3. Mitotic
    4. Normal

  This is achieved using signals from BF and PI channels in addition to Hoechst channel. Furthermore, early and late apoptosis can be distinguished using:
- Shape and intensity properties of individual cells using information from Hoechst channel.
- Number of fragments of each cell.

4.5 Normalization of dataset

4.5.1 Types of normalization

Different well-known types of image normalization techniques are used ([114]) depending upon the type of application:

Normalization using minimum and maximum of given intensities:

For a given grayscale image with pixel values I_{ij} (with g columns and m rows), normalization can be performed by rescaling all pixel values using minimum ($I_{ij,min}$) and maximum ($I_{ij,max}$) pixel values employing Eq.(4.1):

$$I_{ij,norm} = \frac{I_{ij} - I_{ij,min}}{I_{ij,max} - I_{ij,min}} \tag{4.1}$$

resulting in pixel values that are in the range of 0 and 1 with $I_{ij,min} < I_{ij} < I_{ij,max}$. This normalization is irrespective of the intensity scale of input grayscale image i.e. it does not matter whether the image is 8-bits or 16-bits, and only depends upon the extreme intensity values of captured image.

Normalization using percentiles:

Percentiles can be introduced to saturate a certain percentage of pixels lying on either side of the intensity distribution of pixel values. This procedure is more helpful when the intensity values of a certain amount of pixels in a given image are already saturated. Two parameters i.e. a and b could be defined to specify the cutoff percentage on the lower and upper side respectively of the intensity histogram. For an image I_{ij}, upper and lower percentiles i.e. $I_{low,a}$ and $I_{high,b}$, can be defined to saturate the pixels that have values lower than $I_{low,a}$ assigning them a pixel value of 0 and vice versa. It can mathematically be written as:

$$I_{ij,norm} = \begin{cases} \frac{I_{ij} - I_{low,a}}{I_{high,b} - I_{low,a}} & \text{if } I_{low,a} < I_{ij} < I_{high,b} \\ 0 & \text{if } I_{ij} \leq I_{low,a} \\ 1 & \text{if } I_{ij} \geq I_{high,b} \end{cases} \tag{4.2}$$

There is a liability in this method to produce erroneous results if the given image does not contain saturated pixels but useful intensity information on the extreme ends of the distribution. Therefore, upper and lower percentiles should be chosen with great care.

Normalization using statistical parameters:

Normalization can also be performed on a given grayscale image with pixel values I_{ij} by using statistical parameters such as standard deviation (σ) of pixel values and mean pixel value (i_m). It is done mathematically as:

$$I_{ij,norm} = \frac{I_{ij} - i_m}{\sigma} \tag{4.3}$$

where,

$$i_m = \frac{1}{m \cdot g} \sum_{ij} I_{ij} \tag{4.4}$$

and

$$\sigma = \sqrt{\frac{1}{m \cdot g} \sum_{ij} (I_{ij} - i_m)^2} \tag{4.5}$$

In case of a random dataset of images, the comparison of these methods in terms of usefulness and efficiency requires to be done. This can be shown in the following subsection where different normalization methods are considered using dataset of images of human cells to identify different cell states acquired using a microscope.

4.5.2 Image dataset from apoptotic cells

The dataset taken in one biological assay consisted of three different channels. The Hoechst channel was of paramount significance since the fluorescence readouts for these images were considerably higher than other channels. There were 32 16-bits images for each channel in the dataset considered. For the sake of brevity, four images were considered from Hoechst channel for the evaluation of normalization. Since each image is of 16 bits, the gray levels range from 0 - 65535 (g_{min} - g_{max}) while size of each image is 1024 × 1344. The average lowest pixel value and highest pixel value are 33211 and 36863 respectively for these four images.

The main objective is to detect all possible cells in each image and count them, and to extract features of each segmented cell. The restriction is to do the segmentation based on pre-processed and normalized images in order to alienate cells from background while retaining enough intensity information to perform proper segmentation. Moreover, for features extraction, the pixel values at critical levels defining each cell class shall be retained in the images used for features extraction based on performed segmentation. However, outliers should be disregarded but they should not be saturated close to critical intensity ranges.

First of all, the images are normalized with respect to corresponding minimum (I_{min}) and maximum (I_{max}) pixel values of each image as shown in Fig. 4.5. For an individual image, it does not matter if it is scaled between g_{min} - I_{max}, I_{min} - I_{max}, I_{min} - g_{max} or g_{min} - g_{max} , since it does not affect the pixels distribution pattern. The histograms for each of these images are shown in Fig. 4.5.

It is clear from Fig. 4.5, that the distribution of foreground pixels do not follow continuous loss of number of pixels towards higher intensities. Moreover, the background is seen to occupy at least 50 % of pixels in all of these images. Secondly, the images are normalized using 5 % percentiles and the resulting images using contrast stretching between the thresholds created by percentiles and their histograms are shown in Fig. 4.5. The shortcoming of percentiles normalization is readily evident as the background is mingled with the foreground pixels.

Arguably, percentile normalization saturating 5 % on each end can be first used for segmentation whereas the image normalized with minimum and maximum intensity levels can be used for feature extraction based on segmented image. Alternatively, the image can be normalized in the full range of available gray levels on the available data i.e. in the range of g_{min} - g_{max}, in which case no saturation is required since the assumption is that none of the pixels are saturated.

Additionally, assumption can be made on the lower and upper sides of intensity levels beyond the minimum and maximum pixel values of the whole dataset and normalize the each image with respect to these two control limits. Moreover, automatic parametric fitting to an estimated background function could also be performed in order to correct background illumination from each image. To find the optimal parameters of surface fitting, feedback-based method proposed in Section 2.3.2 could be used.

In Fig. 4.5, it can be observed that a lot of inherent microscopic background noise is present. This was removed by automatic feedback-based image normalization method described in Section 2.3.2 and the results are presented in Section 3.3.1. The resulting image is then normalized using 5% percentiles according to Eq.(4.2). This image is then used for image segmentation. For feature extraction image obtained one step before was used and normalized according to Eq.(4.1). The outcome after normalization and shading correction is evaluated and presented in detail in Section 3.3.1 as this is the Benchmark dataset 2.

4.6 Cell segmentation

As mentioned earlier, the dataset consists of three channels. The main aim is to segment the information present in different channels, fuse it and uses it to classify different cell states. Therefore, first all three channels should be segmented separately. Then, features extraction and feature selection is performed to classify different cell states. The employed image analysis scheme is divided into three different processing stages. These stages include image normalization, seg-

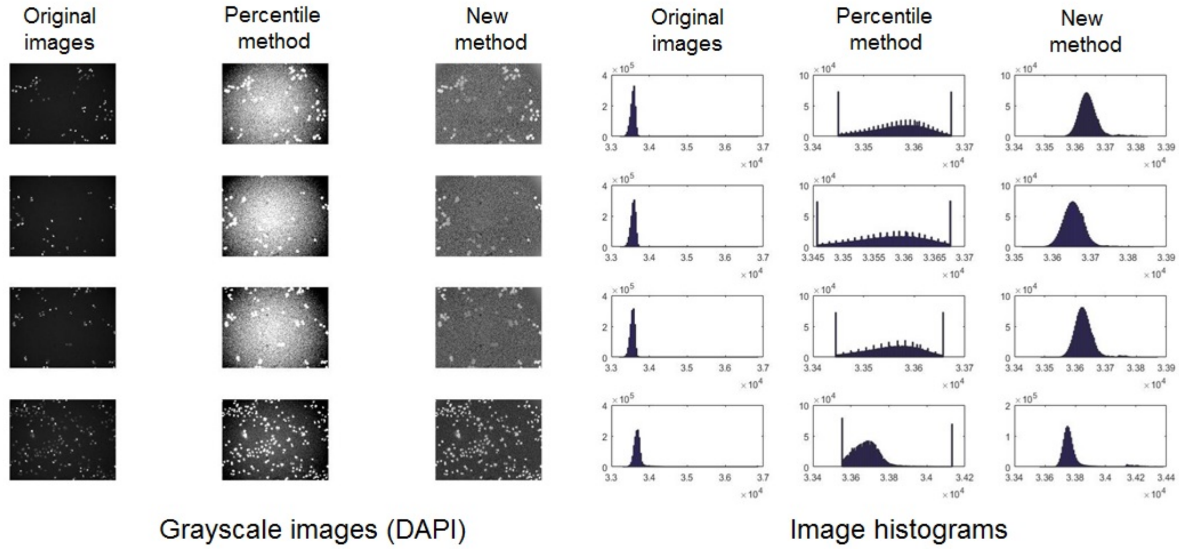


Figure 4.5: Images from Hoechst channel (in first column) normalized with 5 % percentiles with their corresponding histograms (in second column) to show the effect of inherent shading. This shading is removed using new automatic feedback-based image normalization method (in third column) described in Section 2.3.2. Fourth, fifth and sixth columns are corresponding histograms of first, second and third columns respectively.

mentation and feature extraction from:

1. Hoechst 33342 channel
2. PI channel
3. BF channel

4.6.1 Hoechst 33342 channel segmentation:

The image segmentation pipeline for Hoechst channel is shown in the Fig. 4.6. The foremost functionality of the analysis scheme is to segment all the recognizable cells in the image from Hoechst channel and count their number. The given pipeline performs two main steps: Normalization and Segmentation. Furthermore, the image data is used to calculate the shape measurement features and intensity features for each of these individual cells detected. The design algorithms are supervised designed based on *a priori* knowledge about the shape and intensity features of the cells to be found. For such tasks, a lot of manual steps are necessary based on *a priori* knowledge. Feedback-based algorithms can support the parametrization of algorithms by finding out the optimal parameters based only on this knowledge.

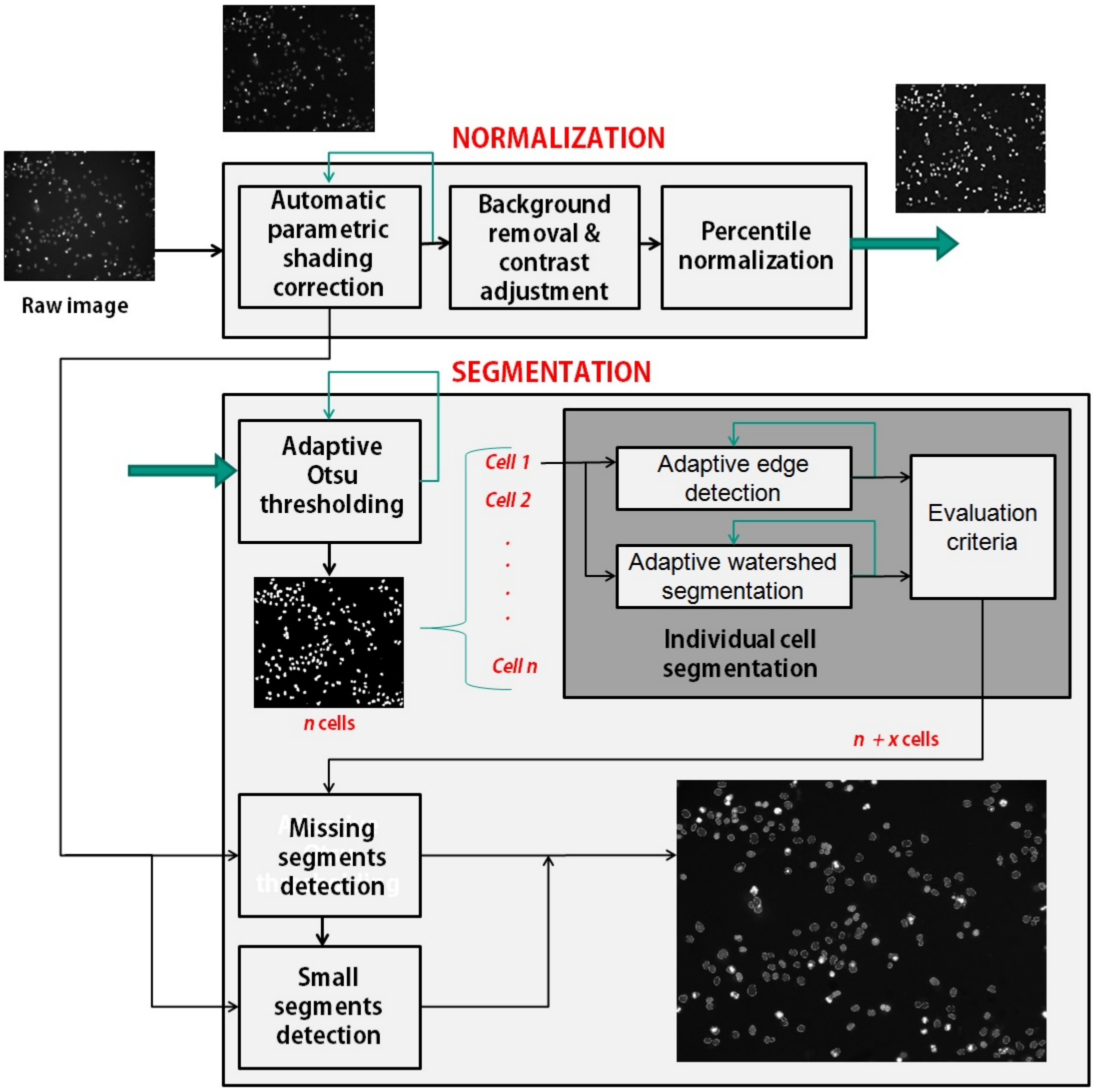

NORMALIZATION
Raw image
Automatic parametric shading correction
Background removal & contrast adjustment
Percentile normalization
SEGMENTATION
Adaptive Otsu thresholding
n cells
Cell 1
Cell 2
Cell n
Adaptive edge detection
Adaptive watershed segmentation
Evaluation criteria
Individual cell segmentation
n + x cells
Missing segments detection
Small segments detection

Figure 4.6: Image analysis scheme for cell segmentation (Hoechst 33342 channel). Green arrows show the feedback-based adaptive image processing steps.

The scheme is implemented in MATLAB using both Gait-CAD and m-file environment. The functionality of detection and counting of cells in a single image has already been implemented in Gait-CAD.

Stepwise procedure

The image segmentation routine to detect and count the number of cells in a given grayscale image from Hoechst channel is given in a stepwise fashion as follows:

1. A raw input grayscale image is fed into the pipeline for normalization.

2. Pre-processing/image normalization is done through various steps and checks. Firstly, automatic parametric shading correction is done using parameter adaptation technique (see Fig. 4.7) described in Section 2.3.2 in Chapter 2. This is done to remove the background noise. Then for further noise removal, image opening with a large structuring element is done followed by contrast enhancement of image such that 1% data is saturated at low and high intensities of input image. This is done to remove further background noise and to make the foreground objects less obscure from the background. Finally, percentile normalization is done using 2% and 98% quantiles in order to improve the contrast further. The results are shown in Fig. 4.8.

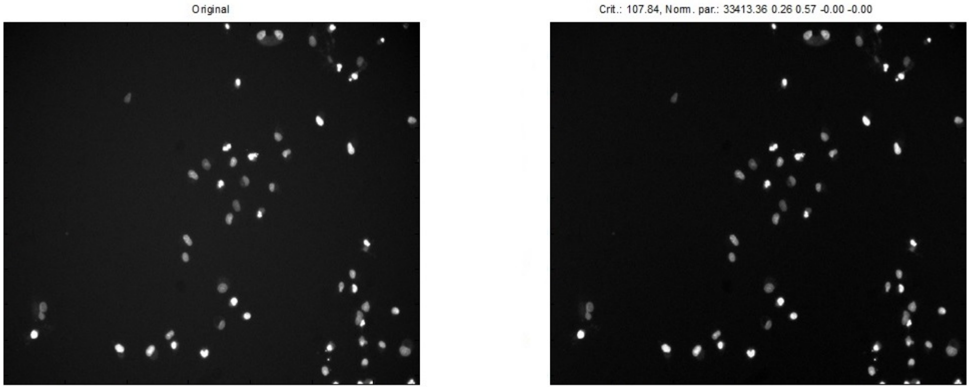


Figure 4.7: Automatic parametric adaptation for additive background shading removal using feedback-based normalization procedure described in Section 2.3.2.

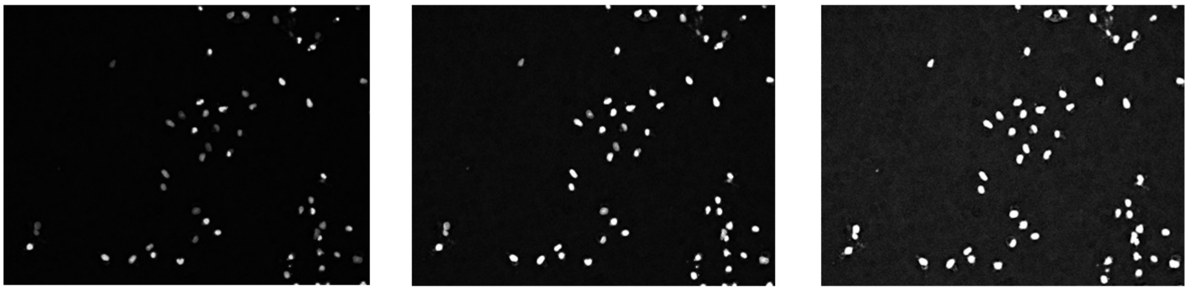


Figure 4.8: From left to right: Background noise removal using procedure given in Section 2.3.2, contrast adjustment and percentile normalization.

3. The resulting image is then fed to an adaptive scheme for Otsu's thresholding method using feedback mechanism described in Section 2.4.2. A priori fuzzy knowledge of area and median intensity features of BLOBS are first extracted using standard Otsu's application. Then, this knowledge is used to design a criterion based on which segmentation result is selected. The parameter to be adapted is the threshold for intensity in standard Otsu's method. The outcome that maximizes the fuzzy criteria is selected and the result is shown

in Fig. 4.9. This would produce n BLOBs of cells. Based on the segmented image, a priori knowledge is recalculated for further use.

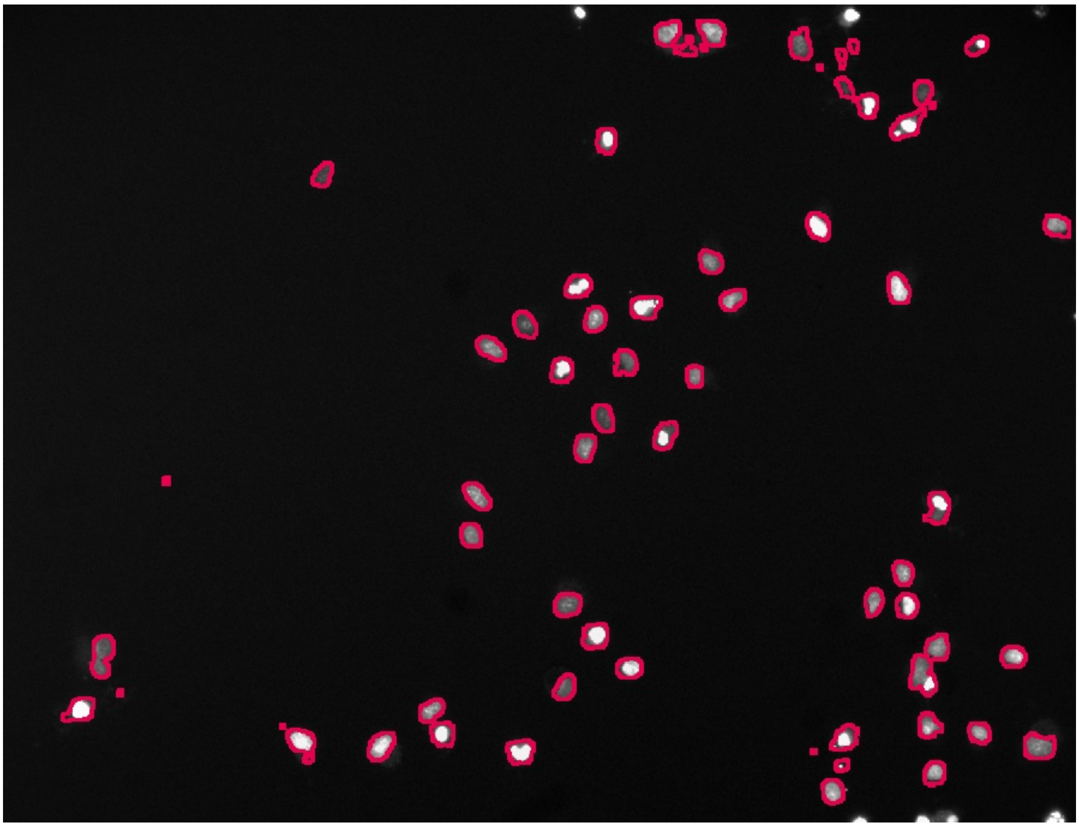


Figure 4.9: Result of parameter adaptation for intensity thresholding in standard Otsu's method using feedback-based technique described in Section 2.4.2.

4. In order to separate the bigger BLOBs (i.e. they may be representing two overlapping cells or cells having their boundaries in contact with each other), each BLOB is passed sequentially for further segmentation separately through Sobel edge detection (Edge) and watershed algorithm (WS). Both of them are used to check whether which algorithm achieves better cell separation. An evaluation criterion is placed to select the better outcome. Each method is defined by variable $segtype$ .Parameter adaptation is used in both implementations using proposed feedback-based method in Section 2.4.2. In case of watershed, intensity threshold is adapted to compute regional maxima, which is then used as a map in watershed algorithm. Results are based on criteria maximization ((2.42) with $h = 2$) using a priori knowledge about area and intensity features of the BLOBs created at the end of previous step. Similarly, Sobel edge detection is also applied in a feedback mechanism by varying image intensity producing a varying gradient magnitude. The evaluation criterion is same based on a priori fuzzy features extracted in last step. For all criteria, abstract ground truth for area and number of cells was used. Fuzzy formulation using trapezoidal

functions was done to describe a priori features in the same fashion as described in Section 3.3.1.

For each BLOB result, a selection criterion is put in place to decide between the segmentation resulting from both aforementioned techniques. The aim here is to separate overlapping or closely lying segments while also dealing with the issues of under-segmentation. The criterion is based on more number of cells with larger areas. This is illustrated by an example shown in Fig. 4.10. A higher number of cells with larger areas are expected (in order to avoid small noisy BLOBs or over-segmentation). This may produce x cells in addition, or x could be 0 in case where both algorithms fail to find additional cells. Therefore, $n + x$ cells are forwarded to next step.

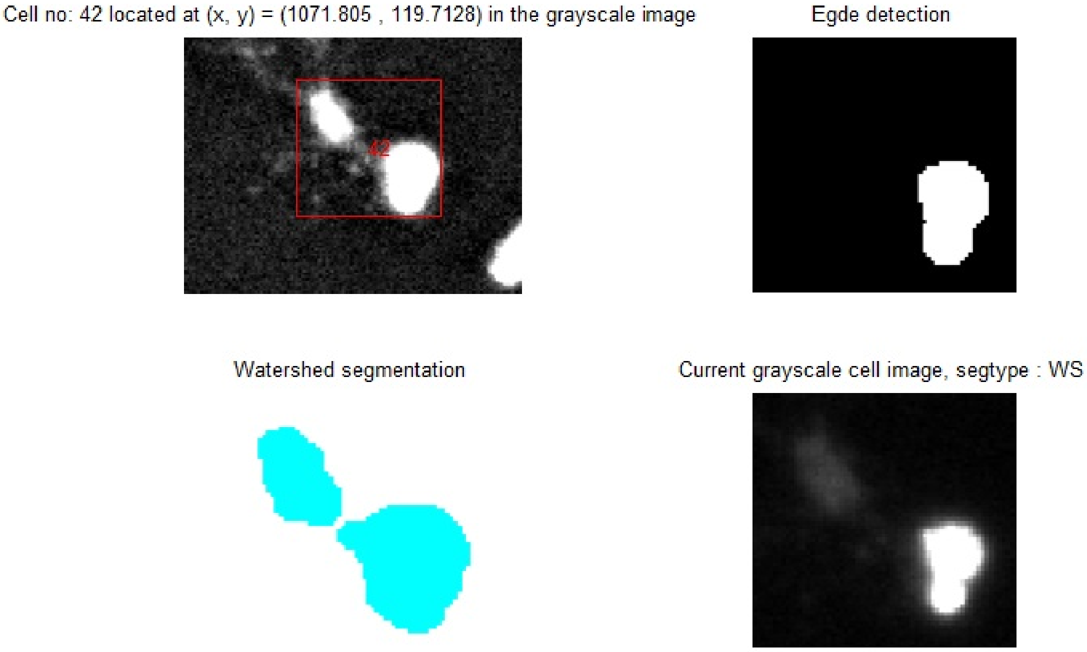

Cell no: 42 located at (x, y) = (1071.805 , 119.7128) in the grayscale image
42
Egde detection
Watershed segmentation
Current grayscale cell image, segtype : WS

Figure 4.10: Result of parameter adaptation for using both watershed algorithm and Sobel edge detection and corresponding criterion-based selection using (2.42) with a priori reference features i.e. area and number. The result for the method (watershed selected in this case indicated by $segtype$: WS) which maximizes this criterion for each segment considered for further segmentation is selected.

5. Next step is to check if any detectable cell is left behind. Before this step, very small segments that are below the predefined lower limit of area detection are removed. To check missing segments, output image from automatic shading correction block is used in conjunction with output from the previous step in order to segment only those areas that are not covered by $n + x$ BLOBs. The resulting image is shown in Fig. 4.11. It can be seen that nearly all segments were detected except for the ones at borders (excluded inten-

tionally). Therefore, the BLOBs smaller than specified minimum size threshold $f_{area,min}$[2] only correspond to parts of the segments detected in previous step. If they are not able to recombine with their parent segments, they are deleted considering the size threshold. Otherwise, missing segments are added if their area is large enough. Missing segments are added to the segments coming out of previous step.

Figure 4.11: Missing segments detection.

6. Based on resulting binary image and intensity image coming out of automatic shading correction, small bright segments are detected that may be the part of fragmented nuclei. However, this step may produce some left out parts of the already detected cells and should be therefore carefully parametrized. Additional small segments are then added to binary image coming out of previous step.

7. In the last step, all small segments ($f_{area,i}$ < $f_{area,min}$) are removed from binary image once again to eliminate any redundant tiny BLOBs and final segmentation is obtained as shown in Fig. 4.12.

Every cell in an image has its own label and all its shape and intensity features are stored in an array with respect to its label. The intensity and shape features are the further used for the classifications of cell states of detected cells in the input image from Hoechst channel as described above.

[2] $f_{area,min}$ is based on a priori area features (i.e. done through manual inspection / expert labeling).

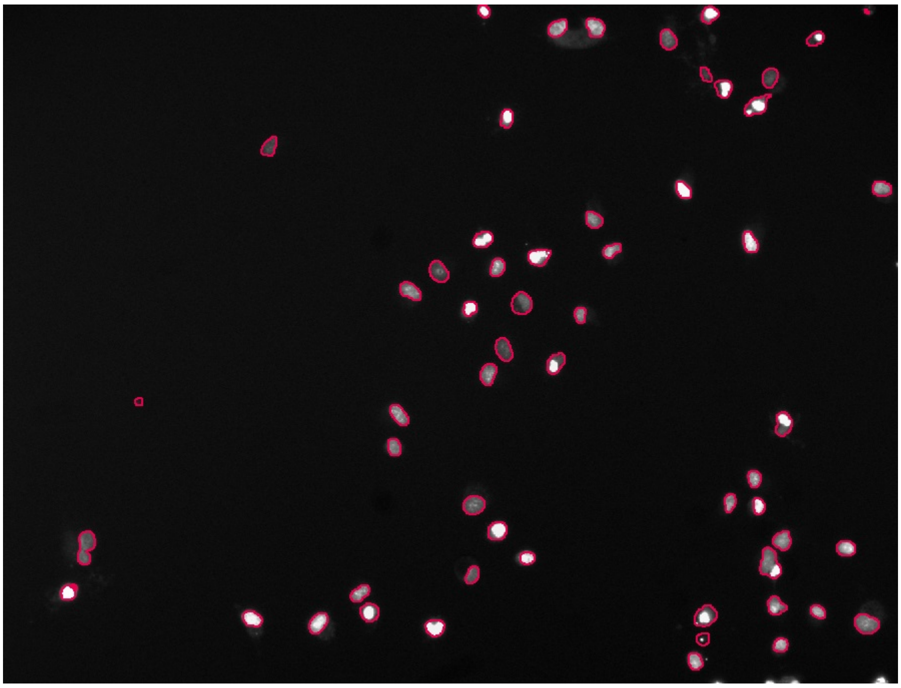


Figure 4.12: Final cell segmentation in the nuclear staining channel.

Sub-structure segmentation

In order to distinguish between different cell states, it is advisable to extract more usable features present in the sub-structures of cells. For example in partially condensed or early apoptotic cell states, there are few bright spots that could be formulated as a feature of sub-structure. An intensity threshold t_{in} at each level n is adapted for an individual cell i on $n_{max} = 5$ different levels in order to explore sub-structures such that:

$$t_{in} = \begin{cases} t_{i,min} & \text{if } n = 1 \\ t_{i,max} & \text{if } n_{max} > 1 \,\&\, n = n_{max} \\ \frac{t_{i,max} - t_{i,min}}{n-2} & \text{if } n_{max} > 2 \,\&\, n < n_{max} \end{cases} \tag{4.6}$$

where, t_{in} is the vector containing local threshold level values from lowest image intensity level $t_{i,min}$ to highest image intensity level $t_{i,max}$. The result for such a cell state and corresponding sub-structure segmentation is given in Fig. 4.13.

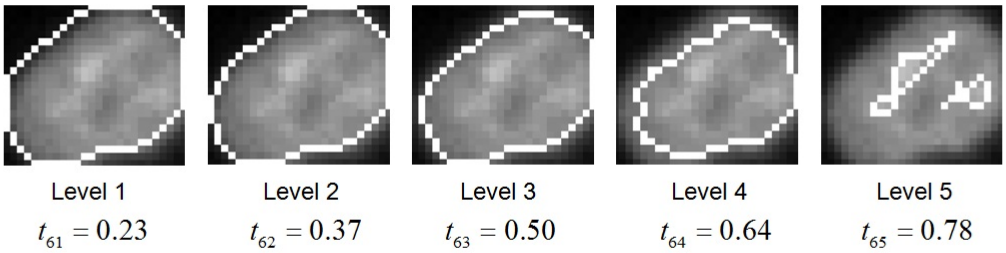


Figure 4.13: Sub-structure segmentation using adaptive thresholding for each individual cell using minimum and maximum of the given intensity. At level 5, at t_{in} = 0.78, two segments were observed. Number 6 in the subscript refers to the cell segment number of the original Hoechst channel final segmentation.

4.6.2 Bright field (BF) channel segmentation:

In BF channel, morphological changes to cell structure can be observed. This channel can be used in addition to Hoechst channel to extract certain new features that could help user to differentiate between potentially different cell states. For e.g., mitotic cells have round boundaries (referring to eccentricity as a possible feature) with clear membrane structure in BF channel. Therefore, it is also essential to segment BF channel.

The challenge here is to extract the protruded structures correctly from the channel images such that useful features could be extracted from them. Here, an important feature is also needed to be defined that can differentiate between different cell states. The pipeline used for BF channel segmentation is given in Fig. 4.14.

Stepwise procedure

The stepwise procedure to segment BF channel is given as follows:

1. A raw BF channel image is fed into the pipeline for segmentation (see image on left in Fig. 4.15).
2. For noise removal, automatic background shading correction is done as with nuclear staining channel using feedback-based parameter adaptation technique proposed in Section 2.3.2 in Chapter 2. The results are given in the right image of Fig. 4.15.
3. Min-max normalization is applied and the image is scaled between 0 and 1.
4. Gradient magnitude recursive Gaussian image filter is then applied to the resulting image. This filter computes the magnitude of the image gradient at each pixel location. This is equivalent to an image smoothing process using convolution with a Gaussian kernel and

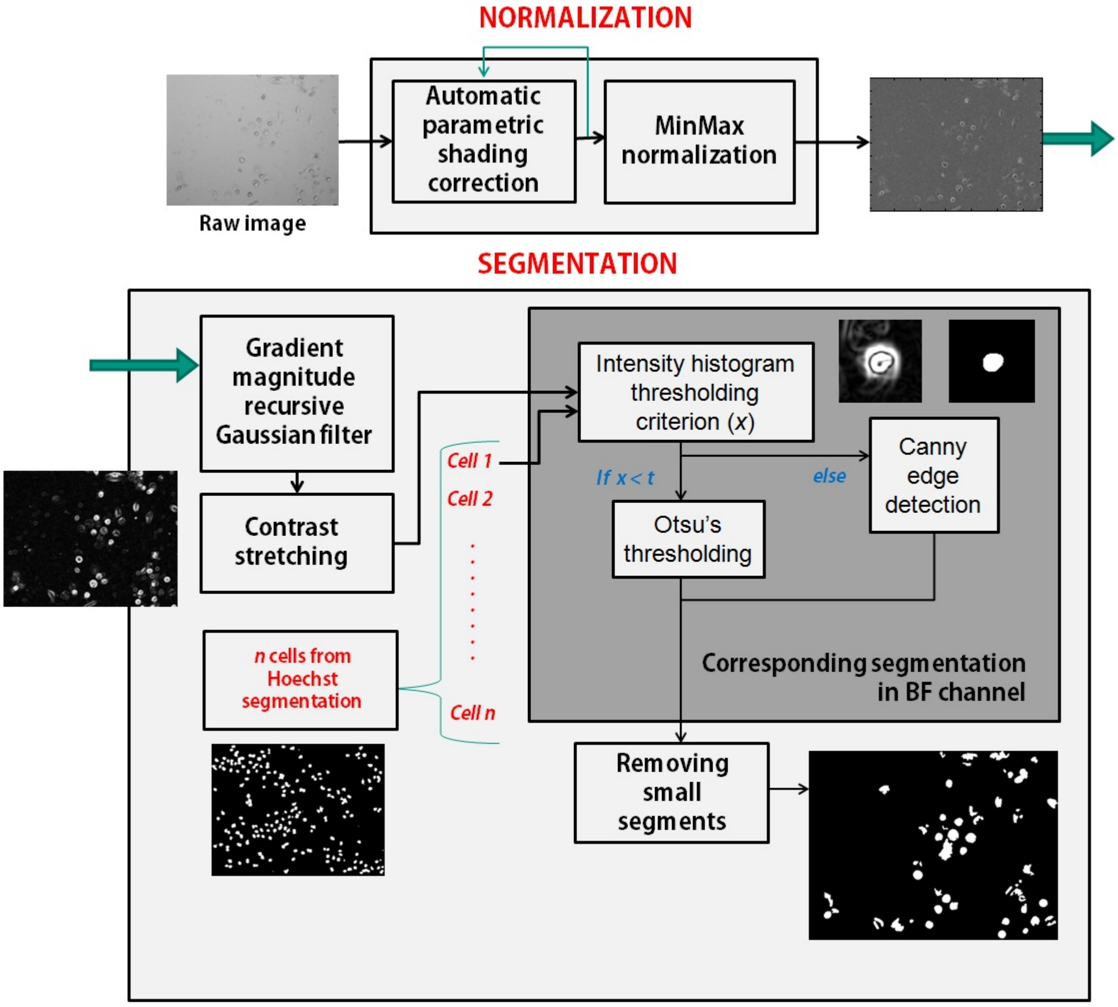

NORMALIZATION
Automatic parametric shading correction
MinMax normalization
Raw image
SEGMENTATION
Gradient magnitude recursive Gaussian filter
Contrast stretching
n cells from Hoechst segmentation
Cell 1
Cell 2
Cell n
Intensity histogram thresholding criterion (*x*)
If *x* < *t*
else
Otsu's thresholding
Canny edge detection
Corresponding segmentation in BF channel
Removing small segments

Figure 4.14: Image analysis scheme for cell segmentation (BF channel).

then applying a differential operator on the resulting image. The result is shown on left in Fig. 4.16. Then, contrast stretching is done in order to make foreground objects more visible (see right image in Fig. 4.16).

5. In the next step, the segmented image from Hoechst channel is used to extract information from BF channel. Since most of the information about cell states is present in Hoechst channel, it is quite reasonable to use its segmented cells for the extraction of segments obtained by BF segmentation. For each Hoechst segment, BF information is extracted exactly at the same pixel locations specified by a certain BLOB in the Hoechst channel. For each BLOB in Hoechst, BF segmentation result is extracted in the same window. Intensity histogram features are extracted in order to design a criterion that enables the algorithm to choose between thresholding and edge detection operation. The purpose of this further segmentation step is to ensure that the right structures from BF channels are extracted in terms of their roundness factor. Based on histogram, it is seen that if the ratio between

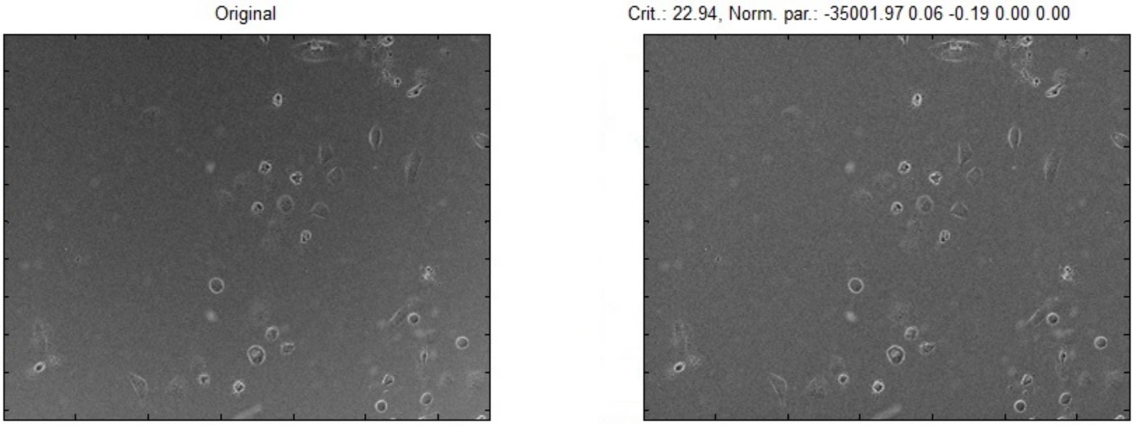


Figure 4.15: On left: Original complemented BF channel image. On right: result of automatic shading removal in BF channel image.

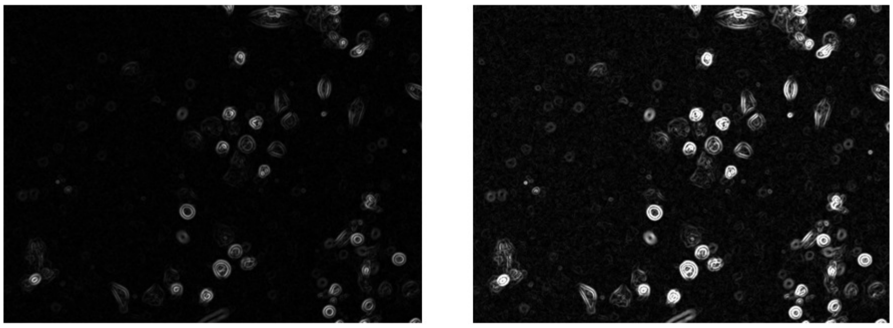


Figure 4.16: Left: BF channel image segmentation using gradient magnitude recursive Gaussian image filter. Right: contrast stretching

higher 20% intensity bins (i.e. number of pixels within a specific intensity level) and lower 20% intensity bins is larger than a certain boundary threshold p_x, then edge detection is selected for that particular segment otherwise Otsu's thresholding is used to deliver objects for eccentricity calculation in the BF channel. Few examples of this criterion and corresponding objects segmented are shown in Fig. 4.17 along with their eccentricity values.

6. The segments obtained from previous step form the new BF segmented image. Finally, small irrelevant segments (i.e. $f_{area,i} < f_{area,min}$) are removed from it to get the final BF channel segmentation as shown in Fig. 4.18

4.6.3 Propidium Iodide (PI) channel segmentation:

PI channel is used to detect signals in case of apoptotic and necrotic cell states. Therefore, it is necessary to segment this channel to classify certain dying cell states. This channel does not

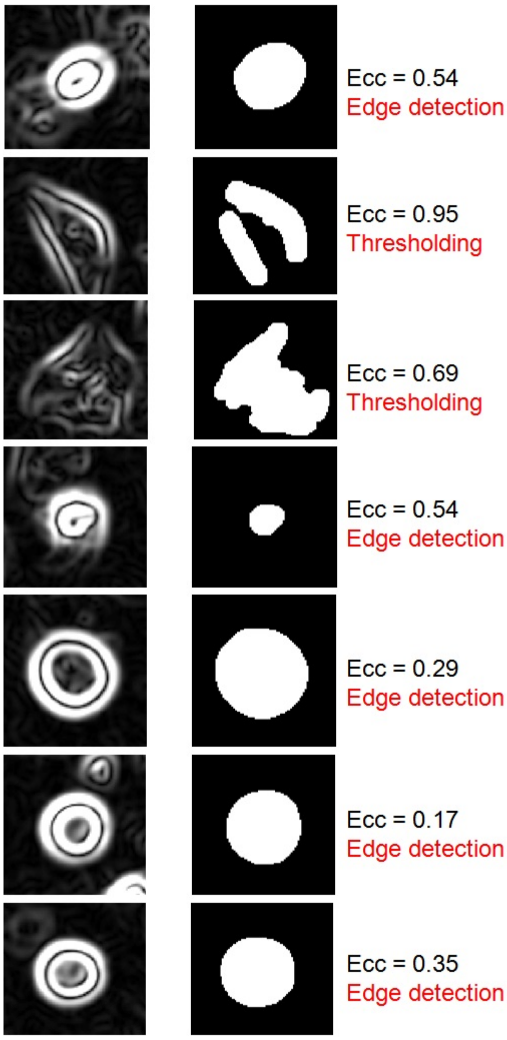


Figure 4.17: First column: Original segments from BF channel image segmentation using gradient magnitude recursive Gaussian image filter. Second column: Delivered objects with their eccentricity values and the method chosen for segmentation. The threshold value ($p_x = 5$) was selected.

contain a lot of information, so normalization and segmentation processes are relatively easier. The pipeline used is shown in the Fig. 4.19.

Stepwise procedure

1. A raw PI image (on right in Fig. 4.20) is inputted to the PI channel segmentation pipeline.
2. Automatic background shading correction is done just like for the images of other channels. The resulting image is shown on left in Fig. 4.20.

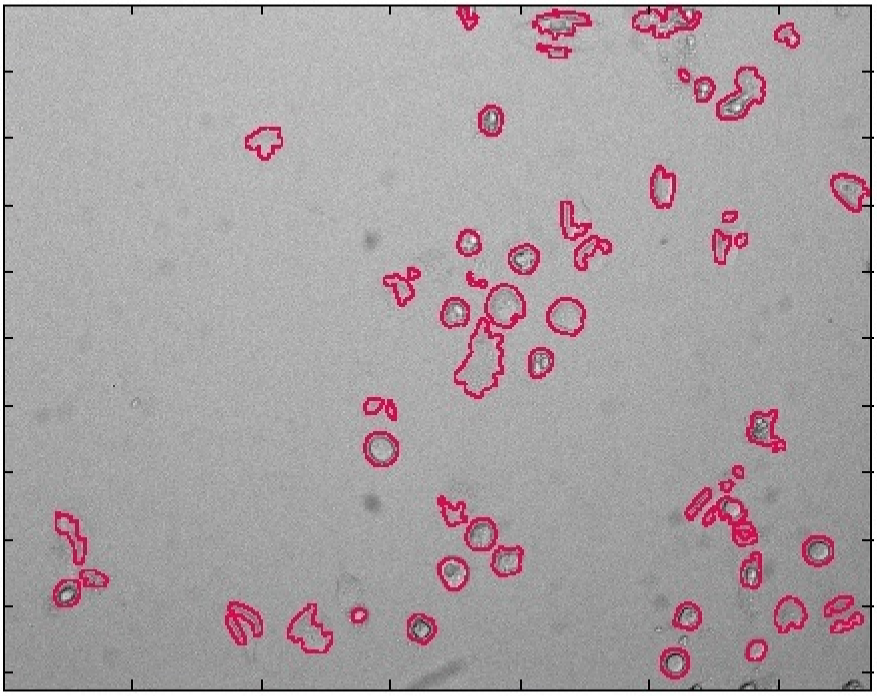


Figure 4.18: Final BF channel segmentation

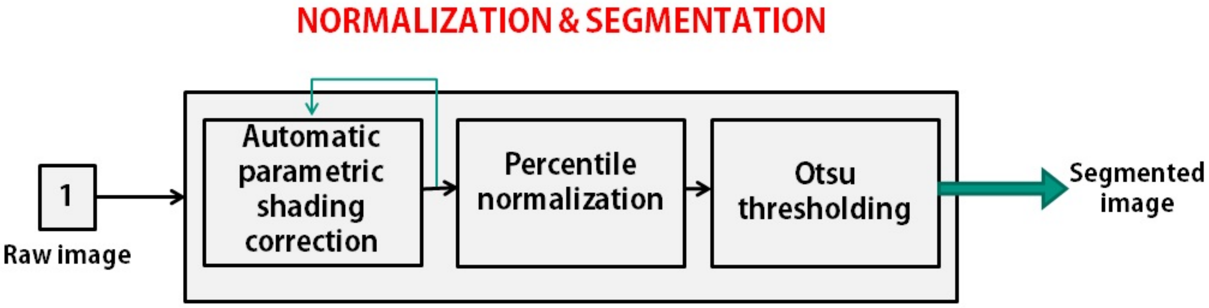

NORMALIZATION & SEGMENTATION
1
Raw image
Automatic parametric shading correction
Percentile normalization
Otsu thresholding
Segmented image

Figure 4.19: Image analysis scheme for cell segmentation (PI channel).

3. Percentile normalization is then done using only the quantile (1%) at the lower side of intensity. At higher side, outlier removal was avoided because the intensity information contained in PI image is fairly scarce with very bright spots (i.e. signals are traced) that might be removed if upper quantile is used.

4. Otsu's thresholding is then done followed by an image dilation to get the required BLOBs. The final segmentation result is shown in Fig. 4.21.

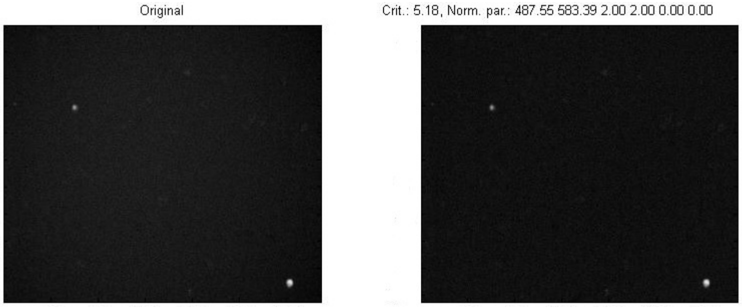


Figure 4.20: On left: Original PI channel image. On right: result of automatic shading removal in PI channel image.

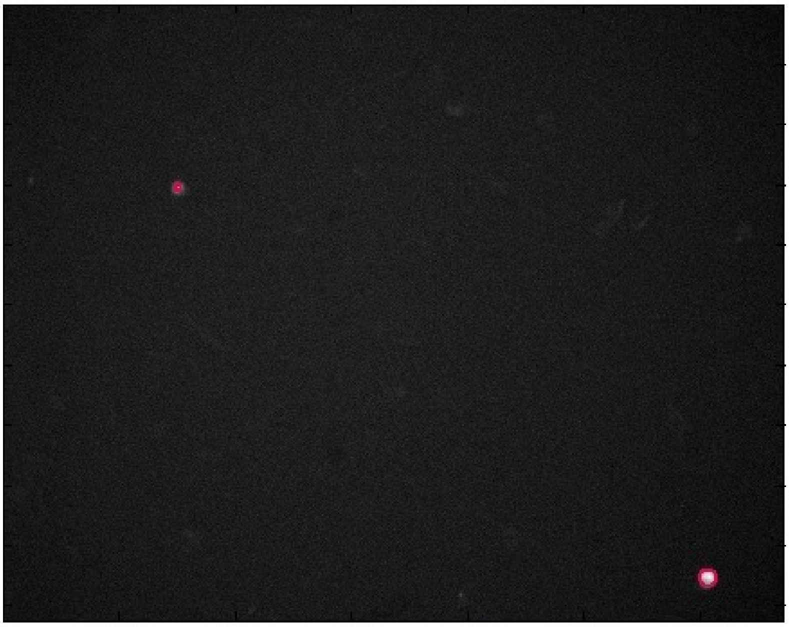


Figure 4.21: Final PI channel segmentation

4.7 Segment assignment

The devised pipeline in Section 2.4.3 is applied to the segments obtained. The dataset contains some images of fragmented nuclei from cells that also show some distinctive features in the BF channel. Some images in this dataset contain 5 to 10 % of cells that are fragmented and the fragments belong to the same nuclei. Therefore, room to improve the segmentation accuracy is evident. The results are shown in Figs. 4.22 and 4.23.

The goal of this processing pipeline is to annotate all segments correctly and assign only one label to all those nuclear segments that belong to one cell while giving each nucleus a unique label. All such segments belonging to each other are assigned the same number and exact same color in the final annotated image (a section of such an image is shown in Fig. 4.22). Here, the label 77 and specific red color tone is assigned to two segments that belong to each other. Therefore, it is easier for users to instantly see which segments belong to each other and vice versa.

Some *critical cases* using different channels separately and combined are shown in Fig. 4.23. *Critical cases* are those that require immediate assignment but were not obtained by using only Hoechst segments. The *critical cases* are a combination of:

1. Fragmented nuclei (as defined by manual inspection) which however were labeled differently from each other using Hoechst segmentation only.
2. Over-segmentation of the nuclei thus generating a number of Hoechst segments not corresponding to the true number of nuclei.

In Fig. 4.23, each column belongs to one of the several *critical cases*. The first row shows critical nuclear segments in the Hoechst channel and their corresponding appearance in the BF channel is given in the second row. The overlaid images are displayed in the third row and show the segments from Hoechst in red over the grayscale BF image. The resulting annotation shows segments that belong to each other encircled together in the final row. From Fig. 4.23, it is clear that with such a method *critical cases* can be resolved Tab. 4.1.

However, the improvement in segmentation results is highly dependent upon the number of *critical cases* with respect to the total number of segments present in an image. If a higher number of fragmented nuclei belonging to the same cell is present, such an algorithm can highly improve the segmentation result Tab. 4.1.

The improvement in segmentation results by employing this method is given with respect to both *critical cases* and per image. Percentage improvement (ψ_{image}) in segmentation results using the total number of correct segments with respect to the total number of segments detected originally in a given image is given by :

$$\psi_{image} = 100 \cdot \frac{1 - (n_t - \Delta)}{n_t} \tag{4.7}$$

Here, n_t is the total Hoechst segments without using BF channel and Δ is the difference of correct and incorrect number of segments after reassignment.

In terms of *critical cases*, percentage of segmentation success (ψ_{case}) is given by:

$$\psi_{case} = 100 \cdot \frac{n_+ - n_-}{n_{seg}} \tag{4.8}$$

Here, n_+ and n_- are used for number of segments reassigned correctly and incorrectly respectively whereas total number of reassigned segments per image is denoted by n_{seg}.

	n_t	***Critical cases***		**Reassigned**		**Correction (%)**	
Image name	*from Hoechst*	*no. of cases*	*no. of segments*	n_+	n_-	ψ_{image}	ψ_{case}
E1P2	266	11	24	9	4	1.88	69.2
H1P3	82	1	2	1	0	1.22	100
H3P4	40	7	16	8	1	17.5	88.89
H8P4	88	4	9	5	0	5.80	100
H9P1	104	7	17	7	3	3.85	70
H9P4	138	5	11	5	1	2.90	83.3

Table 4.1: Results of segment reassignment and segmentation improvement. The first column shows names of images containing *critical cases*. Some of these cases could be seen in Fig. 4.23. The second column shows the number of nuclear segments found in the Hoechst channel automatically as described in Section 4.6.1. The third column shows the number of *critical cases* solved by using segmentation in the BF channel in addition to the already available Hoechst segmentation (e.g. the three *critical cases* shown in Fig. 4.23). Here, number of cases represents the total number of *critical cases* in a given image and number of segments represents the total number of segments that are involved in *critical cases*. The fourth column shows the total number of segments (belonging to *critical cases*) that are now assigned correctly using this methodology e.g., in column 1 and row 4 of Fig. 4.23 the two lower segments are now correctly assigned to the same cell shown within green encirclements. However, not in all *critical cases* the improved algorithm could correctly assign the segments as manual inspection revealed (listed in Tab. 4.1 as incorrect reassignment). The last column shows the improvement in segmentation results using this method. Here, the first sub-column is assigned to ψ_{image} according to (4.7) and the second sub-column is dedicated to ψ_{case} according to (4.8 [74]).

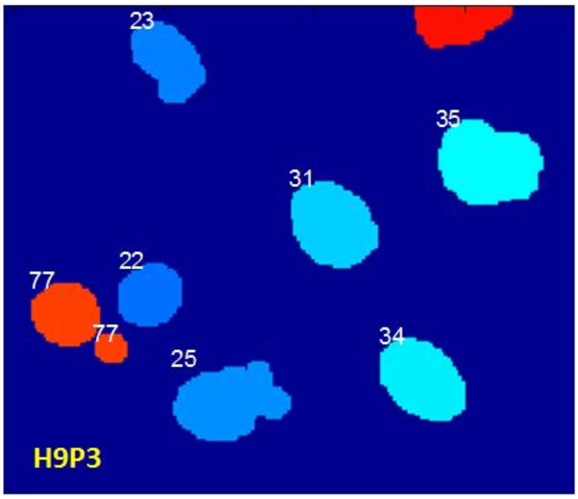


Figure 4.22: Output image annotated with colors and cell numbers. The cell number and color code is same for fragments which belong to each other and is used to distinguish between different cells.

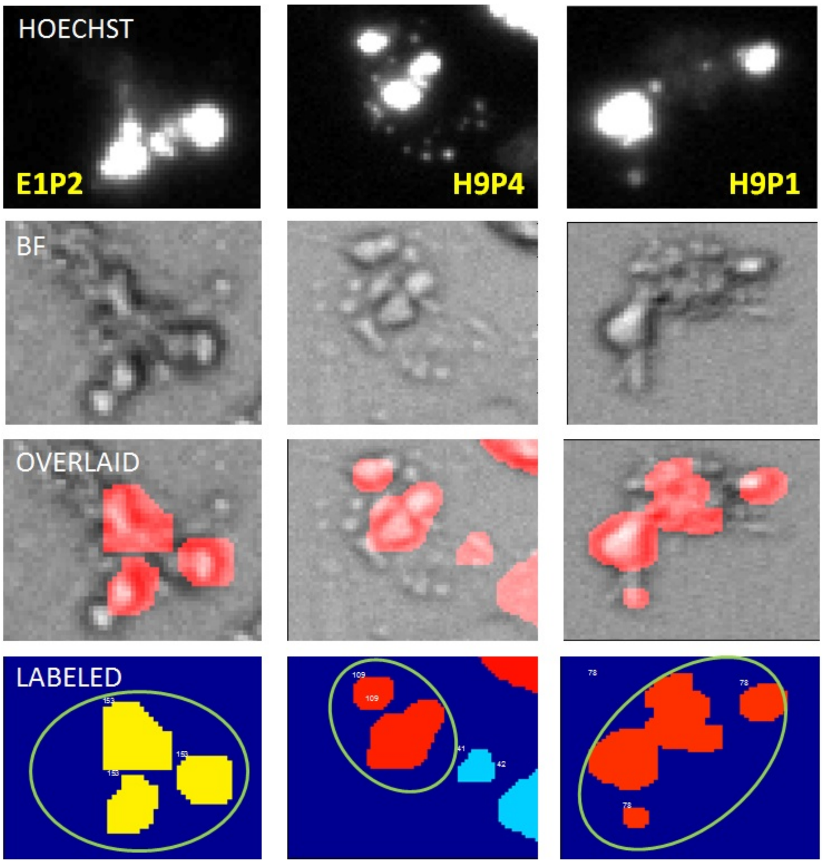


Figure 4.23: Solved *critical cases* from given dataset (1 case per column). The first row shows *critical cases* in the Hoechst channel. The second row shows the same *critical cases* in BF channel. The third row represents an overlay of the BF channel with the segments from the Hoechst channel (shown in red color). The final row represents annotated images whereby segments belonging to each other are encircled together (in green colored encirclements) [74]

4.8 Feature extraction and feature selection

After segmentation process, feature extraction is done based on segments obtained from Hoechst channel. Corresponding features from BF and PI channels are also extracted for these segments. The features are based on shape and intensity of the obtained segments. A total of 67 features are extracted from all channels. More than 20 features are extracted from each channel for each individual segment. The features in Hoechst channel are mean, maximum, median intensity, eccentricity, perimeter, standard deviation of intensity, major axis, minor axis etc. Similar features are extracted in other two channels, too. Additionally, features from sub-structures are also extracted from Hoechst channels. Number of objects in different sublevels (obtained using 4.6) are aggregated to give an average feature named total number of sublevel objects.

MANOVA was used to automatically select top 9 features for the classification purpose. The feature relevances are given in Tab. 4.2.

No.	Feature	Quality
1	Standard deviation in intensity f_3 (Hoechst)	0.735
2	Standard deviation in intensity (PI)	0.888
3	Max intensity (Hoechst)	0.943
4	Min intensity f_7 (Hoechst)	0.954
5	Number of sublevel objects f_8 (Hoechst)	0.962
6	Mean intensity f_1 (Hoechst)	0.966
7	Max intensity (BF)	0.969
8	Mean intensity (BF)	0.971

Table 4.2: MANOVA features relevance table using a forward selection. The quality value indicates the quality of the previously selected features and the new feature.The features which were used for manual selection later were f_1, f_3, f_7 and f_8.

Moreover, this could be done using more number of features using manual selection based on statistical values of the features in different instances. This was done in Gait-CAD. The features selected were:

1. Mean intensity f_1 (Hoechst)
2. Mean intensity f_2 (PI)
3. Standard deviation in intensity f_3 (Hoechst)
4. Eccentricity f_4 (BF)
5. Eccentricity f_5 (Hoechst)
6. Area f_6 (Hoechst)
7. Minimum intensity f_7 (Hoechst)
8. Number of sublevel objects f_8 (Hoechst)
9. Eccentricity criterion f_9 (BF)

4.9 Cell state classification

The main aim is to classify different cell states with good accuracy. The approach used here is supervised, since some labeled examples for each type are desired. The information is this dataset is multi-channel, so it is useful to combine information given in different channels into one overlaid image as shown in the Fig. 4.24. This would efficiently and instantly enable an expert to label cell states and/or see the differences in different class features manually. There were different overlaid channel color schemes to be chosen by the user when analyzing dataset in Gait-CAD (see Fig. 4.25).

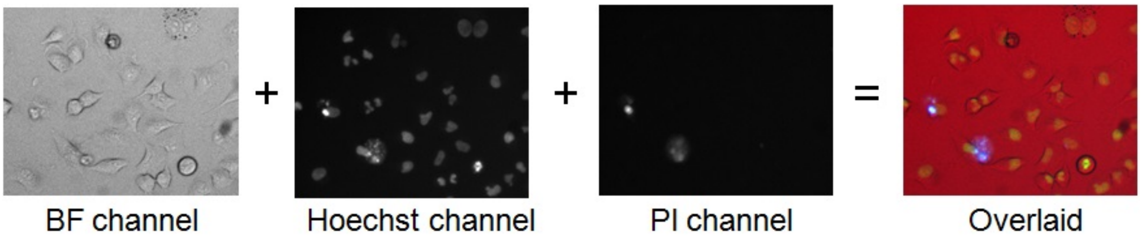


Figure 4.24: Overlaid channel. Purple color in overlaid image shows the presence of PI signal.

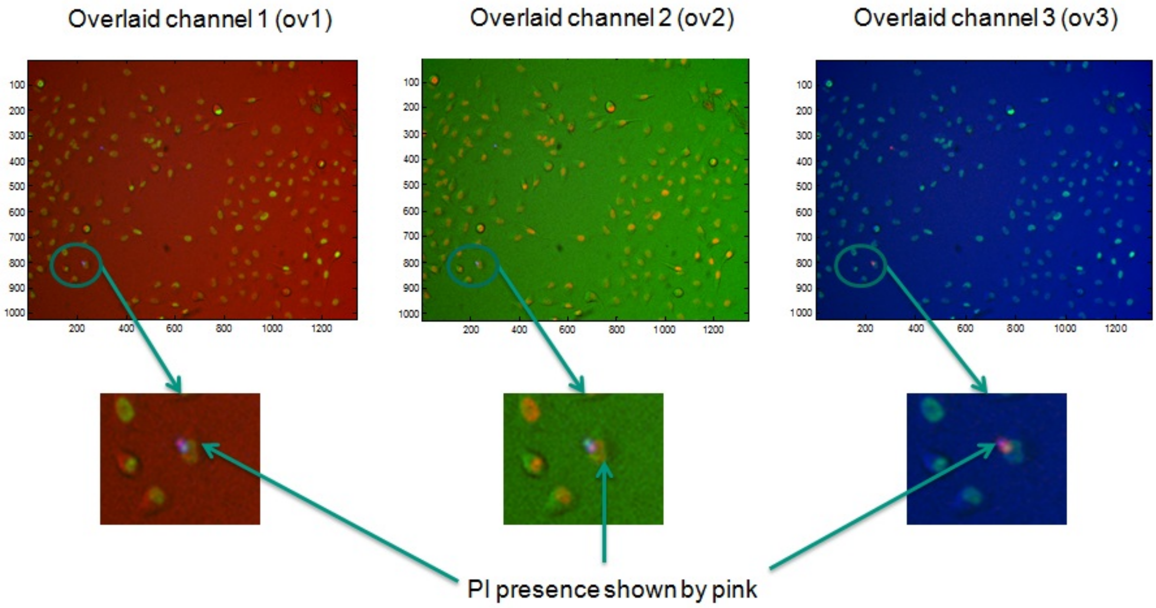


Figure 4.25: Different color schemes in overlaid channel.

4.9.1 Labeling

All different cell states were labeled by biologists according to the features that need to be targeted according to experts. However, only few labels were available since manual labeling is time consuming, hectic and individual expert-dependent. Two different experts were used to label cell states based on different channels. A rough estimate of total number of cells to be found in the whole dataset is between 23 to 24 thousand. Around 350 cells were labeled. Some labeled cell states are given in Fig. 4.26.

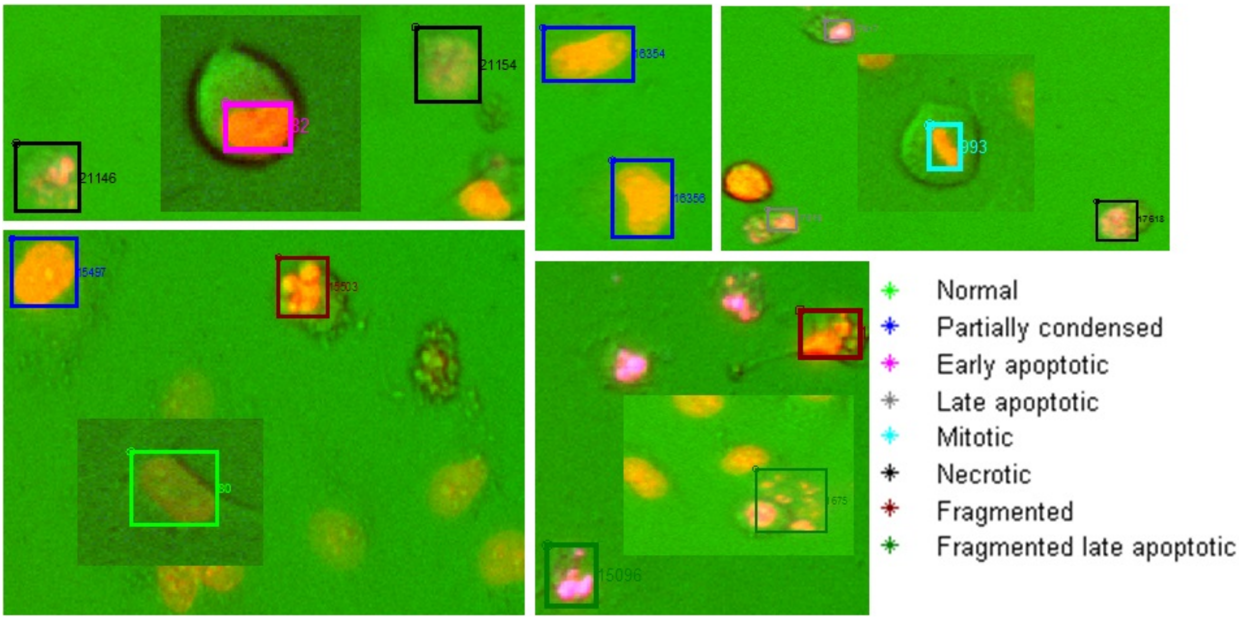


Figure 4.26: Different cell state labels shown in overlaid channel

4.9.2 Training

All examples were first tried with different classifiers like Bayes and SVM for the proof-of-concept. However, required accuracy for many cell states was not achieved due to smaller number of the labeled cells. The feedback-based optimization of image processing parameters based on classification was discarded due to the same reason. The rule-based classifier was then opted for. This classifier was based on expert rules developed on feature values obtained from segmentation results. Expert rules for all cell states are given as follows:

if $(f_1 < 0.45)$ & $(f_2 < 0.05)$ & $(f_6 \geq 300)$; cell state = $Normal$

if $(f_1 \geq 0.45)$ & $(f_2 < 0.05)$ & $(f_3 < 0.25)$ & $(f_7 < 0.3)$; cell state = *Partially condensed*

if $(f_1 < 0.45)$ & $(f_3 \geq 0.17)$ & $((f_3 \geq 0.25) \mid (f_7 \geq 0.3) \mid (f_8 \leq 1) \mid (f_9 < 0.3))$; cell state = *Early apoptotic*

if $(f_2 \geq 0.05)$ & $(f_2 < 0.05)$ & $(f_3 \geq 0.17)$ & $(f_8 \leq 1)$; cell state = *Late apoptotic*

if $(f_1 < 0.45)$ & $(f_2 < 0.05)$ & $(f_3 < 0.17)$; cell state = *Necrotic*

if $(f_2 < 0.05)$ & $(f_9 > 0.1)$ & $(f_5 > 0.6)$; cell state = *Mitotic*

if $(f_2 < 0.05)$ & $(f_8 > 1)$& $((f_3 \geq 0.25) \mid (f_7 \geq 0.3))$; cell state = *Fragmented*

if $(f_2 \geq 0.05)$ & $(f_8 > 1)$& $(f_3 \geq 0.17)$; cell state = *Fragmented late apoptotic*

The feature space used to distinguish between classes is higher dimensional. Therefore, only two examples are shown in Fig. 4.27. Two cell states i.e. normal and partially condensed could be differentiated based on features shown in Fig. 4.27. Two feature (i.e. f_1 and f_6) were selected to show the distribution in the feature space formed by these features for each cell state. In the case of normal cell state, it can be seen that the bounds chosen for these feature (i.e. 0.45 as an upper bound for f_1 and 300 as a lower bound for f_5) are logical. This can be seen in the image on the left hand side of Fig. 4.27. However, the same feature f_1 can be used again with a different bound to differentiate between normal and partially condensed cell states. This can be seen in the image on the right hand side of Fig. 4.27. Therefore, two of the features (i.e. f_1 and f_3) forming the feature space were also shown in Fig. 4.27 to demonstrate the difference in feature space of different classes.

In such a fashion, feature spaces of all cell classes were thoroughly observed to design the rule base for the classifier manually.

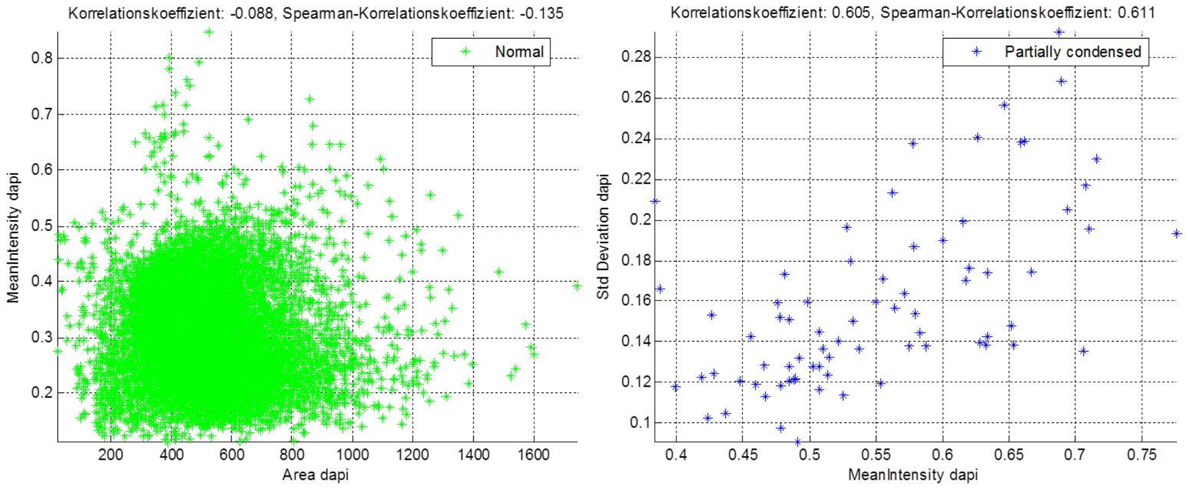


Figure 4.27: Examples of feature space used in expert rules for classifying normal and partially condensed cell states. Left: feature space formed by f_1 (mean intensity in Hoechst channel) and f_5 (cell area in pixels in Hoechst channel). Right: feature space formed by f_1 (mean intensity in Hoechst channel) and f_3 (standard deviation in cell intensity in Hoechst channel).

The values present in the aforementioned rule are based on careful observation of features resulting from final segments in each channel. Similarly, rules for other cell states are defined and the labeled dataset is used to train the classifier.

4.9.3 Testing

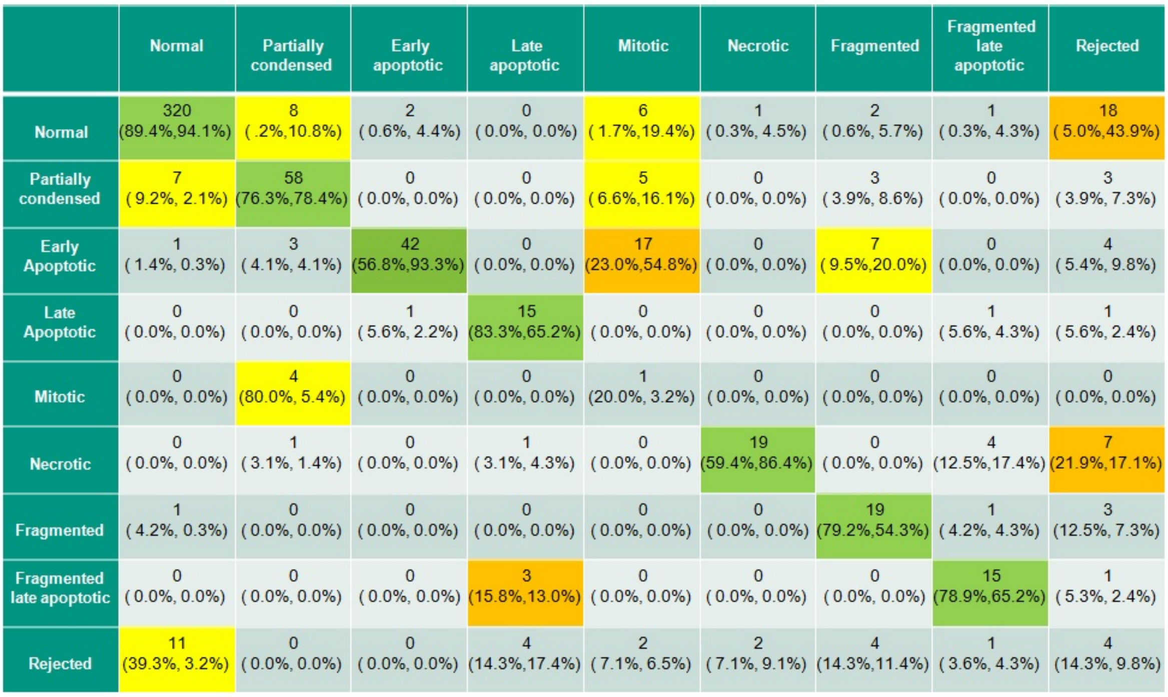
The manually labeled examples are first used to train the classifier. The classification results for different cell states are given in a table shown in Fig. 4.28. It could be seen that normal cell states are accurately classified up to 90%. Partially condensed cell states have also a good accuracy in detection using labels. Moreover, early and late apoptotic states could also be detected with a good accuracy but for other states nothing conclusive could be said due to lesser number of examples per cell state.

Classifier trained on labeled examples is then used to classify other cells in the whole dataset. It is then compared to a commercial software Olympus ScanR results. In ScanR, the features are chosen manually. The segmentation was done by selecting a region in a feature space of PI intensity and Hoechst intensity, such that most of the nuclei could be detected. However, due to the difference in segmentation of both softwares, number of cells detected could also be different in addition to the class labels. The classification results from new feedback-based method and ScanR in Gait-CAD are shown in the Fig. 4.29. Rejected class is the one containing cell features that should definitely be excluded from the classification of the labeled cell states.

Due to different number of cells detected in both cases, a correspondence of cases is necessary to be developed. The complete description of all possible cases is given in Fig. 4.30.

This enables us to instantly see the differences between the segmentation and classification results from both softwares. Since ScanR can only detect five classes as shown in Fig. 4.32, whenever a mitotic cell state is detected correctly, it always improves the classification result (see Fig. 4.33). In Fig. 4.32, % represents the distribution of cell types along the rows and columns respectively. The text highlighted yellow in the table is for the emphasis. Since $1 : 1$ correspondence is very important, it has been highlight. Similarly, there are additional necrotic cell states detected in ScanR which are absent in Gait-CAD. In this regard, an improvement can be further made because these cell states were rejected in new methodology (see Fig. 4.31). However, additionally more cell states were detected which was not possible in ScanR. In a nutshell, the algorithms described here in Chapter 4 and implemented in Gait-CAD are much better in detecting new classes and improves classification is some classes where as it could be a bit worse some other classes compared to ScanR. Since there is no ground truth, so the results are validated by experts.

Mitotic cell states are really important to be classified in the provided dataset. Therefore, in Fig. 4.33, red encirclements show that an additional mitotic class has been detected. This is also observed by the roundness in BF channel. Moreover, in Gait-CAD a successful comparison between the classification results of both methods can be seen. In terms of a confusion matrix, the comparison is given in Fig. 4.34.



	Normal	Partially condensed	Early apoptotic	Late apoptotic	Mitotic	Necrotic	Fragmented	Fragmented late apoptotic	Rejected
Normal	320 (89.4%,94.1%)	8 (.2%,10.8%)	2 (0.6%, 4.4%)	0 (0.0%, 0.0%)	6 (1.7%,19.4%)	1 (0.3%, 4.5%)	2 (0.6%, 5.7%)	1 (0.3%, 4.3%)	18 (5.0%,43.9%)
Partially condensed	7 (9.2%, 2.1%)	58 (76.3%,78.4%)	0 (0.0%, 0.0%)	0 (0.0%, 0.0%)	5 (6.6%,16.1%)	0 (0.0%, 0.0%)	3 (3.9%, 8.6%)	0 (0.0%, 0.0%)	3 (3.9%, 7.3%)
Early Apoptotic	1 (1.4%, 0.3%)	3 (4.1%, 4.1%)	42 (56.8%,93.3%)	0 (0.0%, 0.0%)	17 (23.0%,54.8%)	0 (0.0%, 0.0%)	7 (9.5%,20.0%)	0 (0.0%, 0.0%)	4 (5.4%, 9.8%)
Late Apoptotic	0 (0.0%, 0.0%)	0 (0.0%, 0.0%)	1 (5.6%, 2.2%)	15 (83.3%,65.2%)	0 (0.0%, 0.0%)	0 (0.0%, 0.0%)	0 (0.0%, 0.0%)	1 (5.6%, 4.3%)	1 (5.6%, 2.4%)
Mitotic	0 (0.0%, 0.0%)	4 (80.0%, 5.4%)	0 (0.0%, 0.0%)	0 (0.0%, 0.0%)	1 (20.0%, 3.2%)	0 (0.0%, 0.0%)	0 (0.0%, 0.0%)	0 (0.0%, 0.0%)	0 (0.0%, 0.0%)
Necrotic	0 (0.0%, 0.0%)	1 (3.1%, 1.4%)	0 (0.0%, 0.0%)	1 (3.1%, 4.3%)	0 (0.0%, 0.0%)	19 (59.4%,86.4%)	0 (0.0%, 0.0%)	4 (12.5%,17.4%)	7 (21.9%,17.1%)
Fragmented	1 (4.2%, 0.3%)	0 (0.0%, 0.0%)	0 (0.0%, 0.0%)	0 (0.0%, 0.0%)	0 (0.0%, 0.0%)	0 (0.0%, 0.0%)	19 (79.2%,54.3%)	1 (4.2%, 4.3%)	3 (12.5%, 7.3%)
Fragmented late apoptotic	0 (0.0%, 0.0%)	0 (0.0%, 0.0%)	0 (0.0%, 0.0%)	3 (15.8%,13.0%)	0 (0.0%, 0.0%)	0 (0.0%, 0.0%)	0 (0.0%, 0.0%)	15 (78.9%,65.2%)	1 (5.3%, 2.4%)
Rejected	11 (39.3%, 3.2%)	0 (0.0%, 0.0%)	0 (0.0%, 0.0%)	4 (14.3%,17.4%)	2 (7.1%, 6.5%)	2 (7.1%, 9.1%)	4 (14.3%,11.4%)	1 (3.6%, 4.3%)	4 (14.3%, 9.8%)

Figure 4.28: Classification results from Gait-CAD for labeled examples using a classifier trained on expert rules. Classifier results are along the rows and the expert labeling is along columns. Green colored boxes show good results with high accuracy. Orange colored boxes show high percentage of misclassified examples and yellow boxes show loss percentage of misclassified examples. The two percentages for each box show classification accuracy along rows and columns.

4.10 Conclusions

In this chapter, methods for cell state analysis for human cancer cells were given. Normalization and segmentation procedures were performed in different channels for improved cell segmentation. Method for sub-structure analysis in nuclear staining channel was also done in order to differentiate between classes that show sub-structures. Many features related to intensity and geometry for each cell were extracted from three different channels. Automatic techniques were applied to select top features without worsening the results. Few examples for each targeted cell state were labeled by two experts. A rule-based classifier was designed on these examples and applied consequently on the whole dataset.

By the virtue of this work, an interactive and efficient comparison of the new feedback-based image classification method with commercial software was made possible. The new method was implemented in Gait-CAD and results were compared to a commercial software named ScanR by Olympus. The comparison using different cell correspondences and confusion matrices was made possible. Detection of additional cell classes such as mitotic, fragmented and fragmented

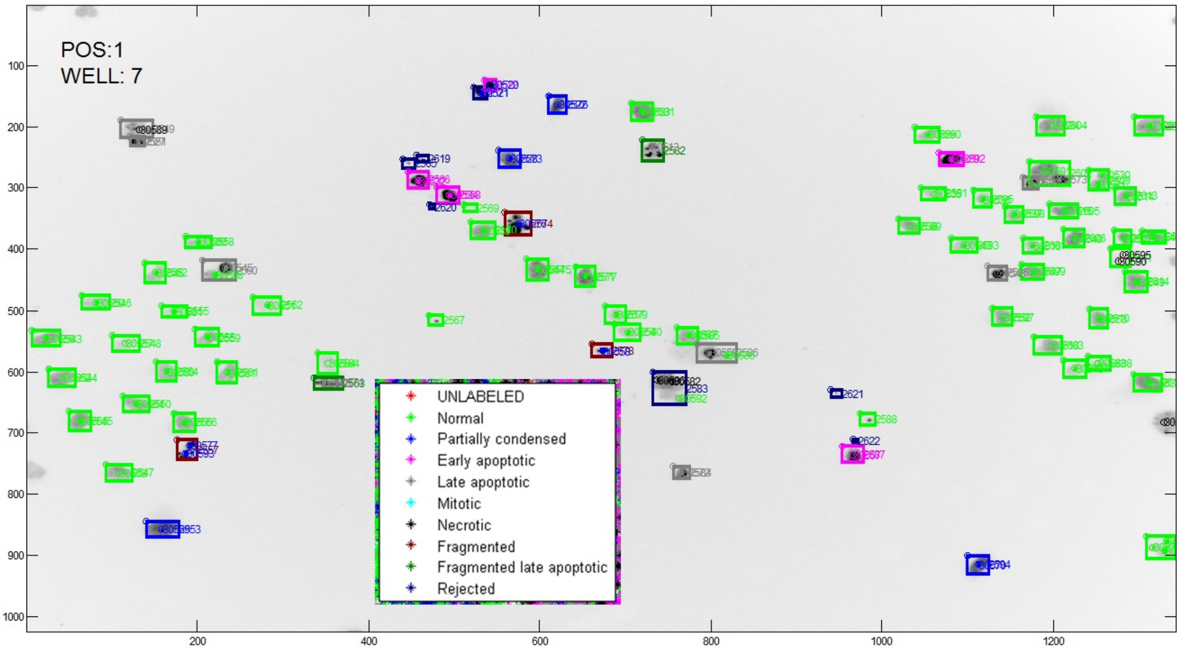
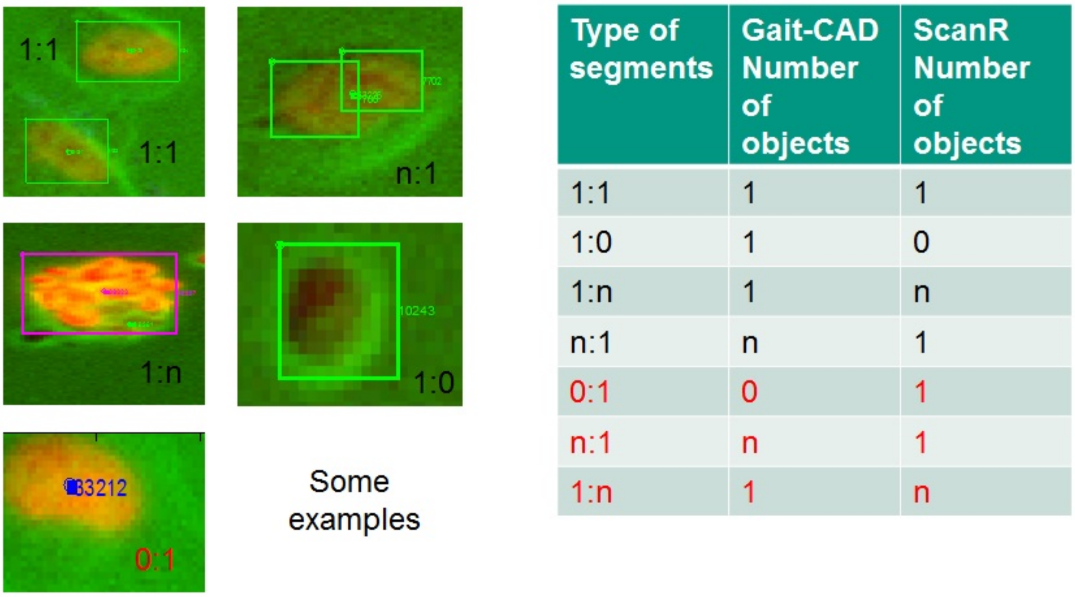


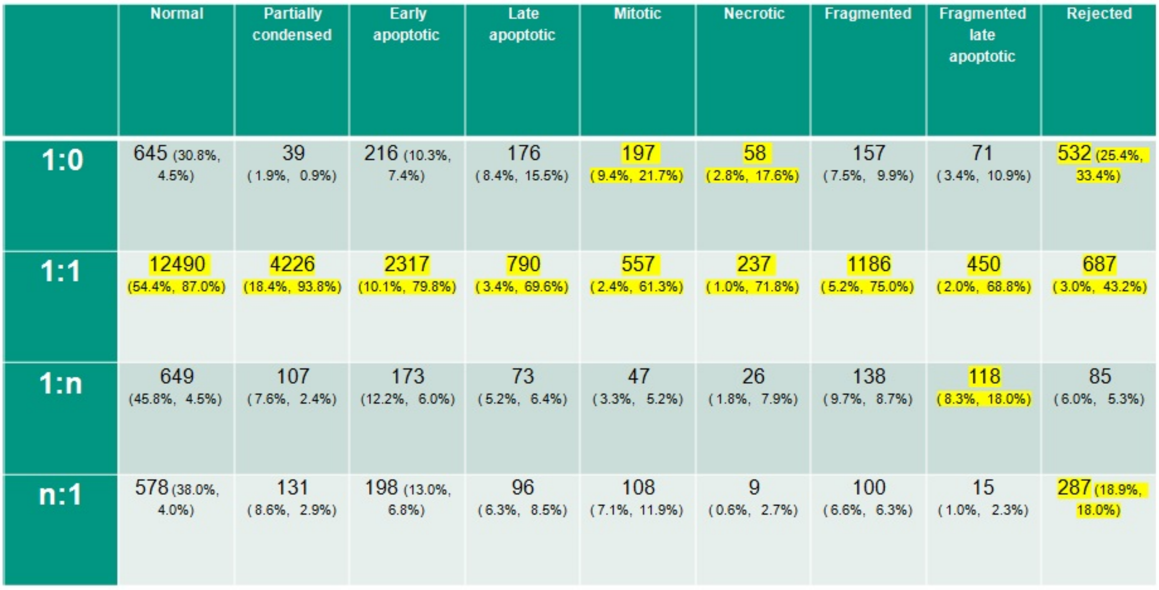
Figure 4.29: Different cell state labels shown in overlaid channel from the new methodology (data points represented by big colored bounding boxes around cells) and ScanR (represented by colored dots present at the center of cells).

late apoptotic was also made possible in Gait-CAD. It is also possible to introduce a rejection class. Total cell count was found to be comparable to that of standard software. A reasonable classification accuracy ($> 80\%$) for normal and partially condensed cells was achieved. Moreover, method for segments reassignment in case of fragmented nuclei using multi-channel information was also proposed and result for cell count was seen to be improved.



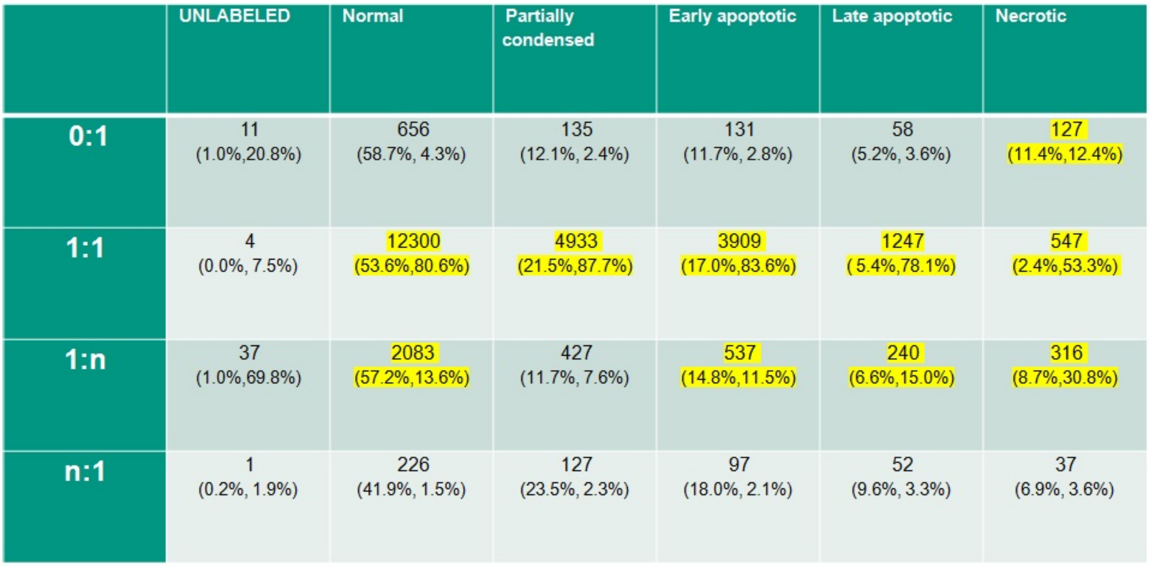
Type of segments	Gait-CAD Number of objects	ScanR Number of objects
1:1	1	1
1:0	1	0
1:n	1	n
n:1	n	1
0:1	0	1
n:1	n	1
1:n	1	n

Figure 4.30: Different cell correspondences for software comparison using some examples of the cases involved. In the table, 1:1 means that for each segment found in Gait-CAD, one corresponding ScanR segment is found. Similarly, 1:0 indicates only Gait-CAD detection and no corresponding ScanR detection. For one segment found in Gait-CAD, if there are more than one segments found in ScanR, 1:n is used. If for more number of segments found in Gait-CAD, there is only one corresponding ScanR segment, it is denoted as n:1. The red colored correspondences describe the same relationship as above but starting for ScanR segments, e.g. 0:1 means for one ScanR segment no Gait-CAD segment was found and so on.



	Normal	Partially condensed	Early apoptotic	Late apoptotic	Mitotic	Necrotic	Fragmented	Fragmented late apoptotic	Rejected
1:0	645 (30.8%, 4.5%)	39 (1.9%, 0.9%)	216 (10.3%, 7.4%)	176 (8.4%, 15.5%)	197 (9.4%, 21.7%)	58 (2.8%, 17.6%)	157 (7.5%, 9.9%)	71 (3.4%, 10.9%)	532 (25.4%, 33.4%)
1:1	12490 (54.4%, 87.0%)	4226 (18.4%, 93.8%)	2317 (10.1%, 79.8%)	790 (3.4%, 69.6%)	557 (2.4%, 61.3%)	237 (1.0%, 71.8%)	1186 (5.2%, 75.0%)	450 (2.0%, 68.8%)	687 (3.0%, 43.2%)
1:n	649 (45.8%, 4.5%)	107 (7.6%, 2.4%)	173 (12.2%, 6.0%)	73 (5.2%, 6.4%)	47 (3.3%, 5.2%)	26 (1.8%, 7.9%)	138 (9.7%, 8.7%)	118 (8.3%, 18.0%)	85 (6.0%, 5.3%)
n:1	578 (38.0%, 4.0%)	131 (8.6%, 2.9%)	198 (13.0%, 6.8%)	96 (6.3%, 8.5%)	108 (7.1%, 11.9%)	9 (0.6%, 2.7%)	100 (6.6%, 6.3%)	15 (1.0%, 2.3%)	287 (18.9%, 18.0%)

Figure 4.31: Classification results from Gait-CAD for all cell correspondences found in ScanR segmentation/classification results. The yellow color highlights the important results in the table. $1:1$ correspondence is important in comparing the detection in both ScanR and Gait-CAD. $1:0$ is here important as it shows the detection of cell states by Gait-CAD that were missing in ScanR.



	UNLABELED	Normal	Partially condensed	Early apoptotic	Late apoptotic	Necrotic
0:1	11 (1.0%,20.8%)	656 (58.7%, 4.3%)	135 (12.1%, 2.4%)	131 (11.7%, 2.8%)	58 (5.2%, 3.6%)	127 (11.4%,12.4%)
1:1	4 (0.0%, 7.5%)	12300 (53.6%,80.6%)	4933 (21.5%,87.7%)	3909 (17.0%,83.6%)	1247 (5.4%,78.1%)	547 (2.4%,53.3%)
1:n	37 (1.0%,69.8%)	2083 (57.2%,13.6%)	427 (11.7%, 7.6%)	537 (14.8%,11.5%)	240 (6.6%,15.0%)	316 (8.7%,30.8%)
n:1	1 (0.2%, 1.9%)	226 (41.9%, 1.5%)	127 (23.5%, 2.3%)	97 (18.0%, 2.1%)	52 (9.6%, 3.3%)	37 (6.9%, 3.6%)

Figure 4.32: Classification results from ScanR for all cell correspondences. Some important results are highlighted using yellow color in the table.

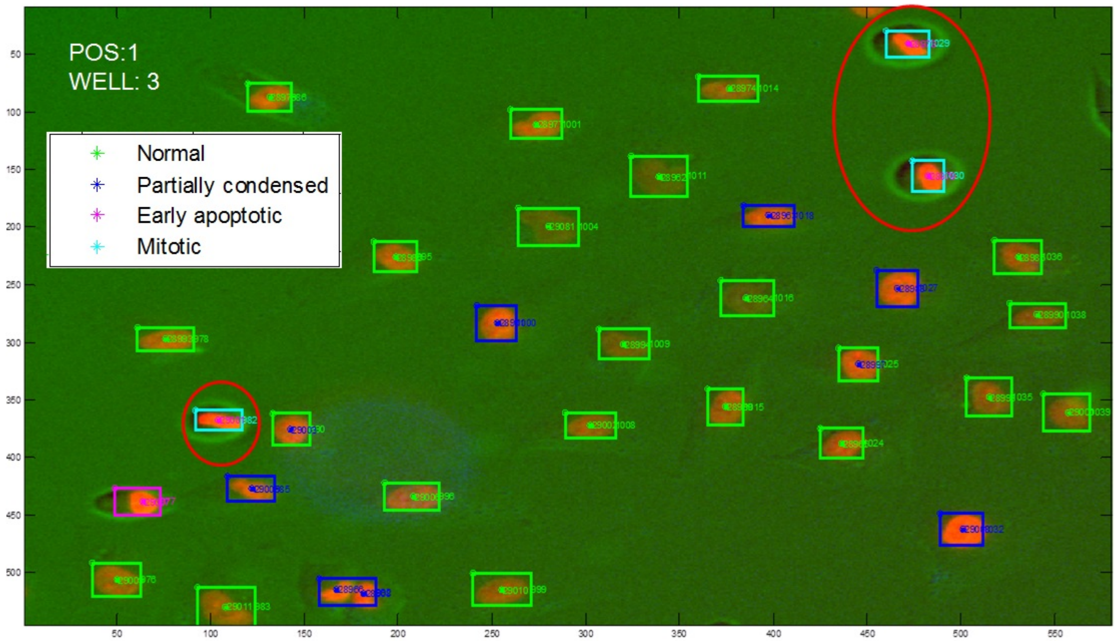
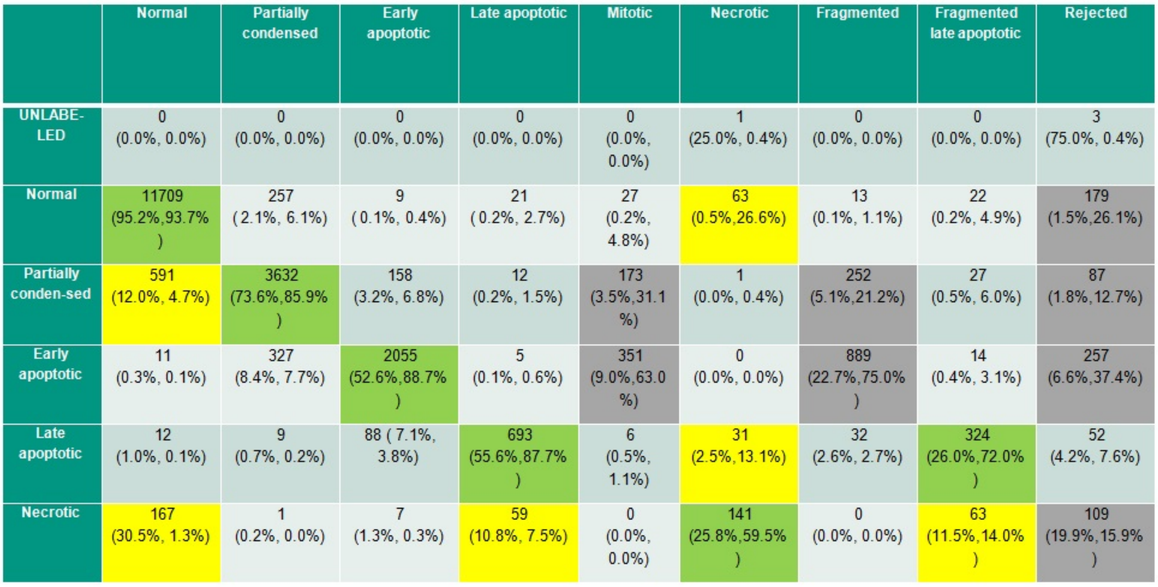


Figure 4.33: Classification results from the new methodology and ScanR for all cell correspondences. Detection of additional class i.e. mitotic by the new software (encircled red).



	Normal	Partially condensed	Early apoptotic	Late apoptotic	Mitotic	Necrotic	Fragmented	Fragmented late apoptotic	Rejected
UNLABE-LED	0 (0.0%, 0.0%)	0 (0.0%, 0.0%)	0 (0.0%, 0.0%)	0 (0.0%, 0.0%)	0 (0.0%, 0.0%)	1 (25.0%, 0.4%)	0 (0.0%, 0.0%)	0 (0.0%, 0.0%)	3 (75.0%, 0.4%)
Normal	11709 (95.2%,93.7%)	257 (2.1%, 6.1%)	9 (0.1%, 0.4%)	21 (0.2%, 2.7%)	27 (0.2%, 4.8%)	63 (0.5%,26.6%)	13 (0.1%, 1.1%)	22 (0.2%, 4.9%)	179 (1.5%,26.1%)
Partially conden-sed	591 (12.0%, 4.7%)	3632 (73.6%,85.9%)	158 (3.2%, 6.8%)	12 (0.2%, 1.5%)	173 (3.5%,31.1%)	1 (0.0%, 0.4%)	252 (5.1%,21.2%)	27 (0.5%, 6.0%)	87 (1.8%,12.7%)
Early apoptotic	11 (0.3%, 0.1%)	327 (8.4%, 7.7%)	2055 (52.6%,88.7%)	5 (0.1%, 0.6%)	351 (9.0%,63.0%)	0 (0.0%, 0.0%)	889 (22.7%,75.0%)	14 (0.4%, 3.1%)	257 (6.6%,37.4%)
Late apoptotic	12 (1.0%, 0.1%)	9 (0.7%, 0.2%)	88 (7.1%, 3.8%)	693 (55.6%,87.7%)	6 (0.5%, 1.1%)	31 (2.5%,13.1%)	32 (2.6%, 2.7%)	324 (26.0%,72.0%)	52 (4.2%, 7.6%)
Necrotic	167 (30.5%, 1.3%)	1 (0.2%, 0.0%)	7 (1.3%, 0.3%)	59 (10.8%, 7.5%)	0 (0.0%, 0.0%)	141 (25.8%,59.5%)	0 (0.0%, 0.0%)	63 (11.5%,14.0%)	109 (19.9%,15.9%)

Figure 4.34: Classification results in terms of a confusion matrix between ScanR and Gait-CAD classes. Green colored boxes are direct comparison for same cell states for both softwares. Yellow colored boxes show areas where Gait-CAD needs improvements. Gray colored boxes show areas where Gait-CAD was able to show improvements over ScanR software.

5 Summary

The aim in this research was to present an automatic image analysis to solve and comprehend technical and natural processes and challenging tasks associated to them. The efficiency and usefulness of automatic image processing techniques are proposed to fine tune the already existing methods and providing different platforms to evaluate the standard algorithms. A new methodology was hereby introduced that facilitates the automatic image analysis by using feedback concepts. Parameters associated to various algorithms of different image processing steps were automatically tuned by using feedback through an evaluation mechanism based on the expected outcome. This helped to select good algorithmic parameters for each algorithm involved. In an image processing/analysis pipeline, different steps were handled using the parameter adaptation to show an improvement in the overall outcome depicted according to the problem description. This was done mostly in terms of object type classification in an image or data set. In order to show robustness of proposed methods, image benchmark datasets were developed having progressive artifact levels with absolute ground truth about the segments boundaries and classes. Standard image processing algorithms were first applied to these datasets and using evaluation criteria, robustness was calculated. Later, the parameters involved in these algorithms were adapted iteratively and optimal parameter sets were used to evaluate robustness on benchmark datasets with increasing artifact levels. It was found that, a user can tune parameters of standard algorithms automatically using the feedback to improve the outcome in noisy images.

Moreover, a challenging biological dataset containing human cells was used to perform complex image processing. It was shown that different image processing steps can be tuned to find better parameters to solve complex tasks like overlapping/touching segments. Moreover, segments reassignment using multichannel information and sub-structure segmentation for advanced feature extraction was also performed to fuse information to find potentially different cell states. Furthermore, in the absence of explicit ground truth, abstract ground truth about segments to be found in a dataset was formulated using fuzzy logic and it was shown that user can perform object based segmentation and classification using parameter tuning.

In a nutshell, a new feedback-based automatic parameter tuning method was proposed, that made the selection of optimal parameters for standard image processing algorithms possible. This was never shown or performed before for standard image segmentation methods such as edge detection and thresholding.

The following claims have been made in this doctoral research work:

1. Supervised feedback-based automatic parameter tuning at different stages in an image processing pipeline and integration of results to improve image object classification
2. Fuzzy formulation of *a priori* knowledge in conjunction with feedback-based automatic parameter tuning in addition to the fuzzy evaluation criteria in cases for both image normalization and image segmentation
3. Robust supervised feedback-driven design of normalization/segmentation techniques through parameters tuning
4. Application of devised feedback-based automatic image processing techniques and algorithms to challenging datasets containing human cancer cell images thereby segmenting challenging new cell classes
5. Development of benchmark dataset (with absolute ground truth) containing hard items with progressive shading and artificial noise levels to evaluate standard image segmentation algorithms
6. Synthesis of benchmark dataset (with absolute ground truth) containing artificial cell types to evaluate automatic feedback-based image classification and comparison with standard feedforward methods
7. Classification of additional cell states in human cancer cells in comparison to standard commercial software
8. Outperformance of feedback-based parameter adaptation of image processing algorithms in comparison to standard feedforward algorithms using benchmark datasets with natural shading and artificial noise
9. Implementation of devised supervised feedback-based automatic image analysis algorithms and their integration into an open-sourced data mining and image analysis GUI, namely Gait-CAD implemented in MATLAB was achieved
10. Upgradation and enhancement of Gait-CAD by importing the cell image analysis results from standard state-of-the-art softwares into Gait-CAD environment thereby allowing interactive and efficient comparison of our methods with standard market software
11. Cell image data labeling, faulty cells and/or faulty objects rejection was done automatically with more than 90 % accuracy using robust automatic image analysis techniques
12. Multichannel cell image data processing for robust image analysis and identification of potentially new cell types using robust feature extraction and intelligent feature selection was also performed

The proposed methods using feedback could be extended to any number of algorithms using definite parameters affecting the outcome of the algorithm. Optimal parameter set can be selected for any image processing algorithm based on the carefully designed evaluation criteria.

Since the new feedback-based parameter tuning method works for the image segmentation/classification against a backdrop of noisy data, this could very well be applied to data acquired under real conditions with a lot of inherent noise.

Application of biological datasets for cell segmentation could also be improved using feedback-based methods depending upon the defined evaluation criteria. In the future, the feedback-based image segmentation algorithms would be applied to 3D data from microscope.

Altogether, this work paves the way for finding optimal parameters in image analysis using different image processing algorithms for the noisy data as the parameter adaptation could work better and is more robust to illumination and noise variations.

A Important Symbols and Names

Symbol	Description
A	overall artifact level present in pixel value of $\mathbf{I}_{truth}$ at indices i and j an image
$A(n_1, n_2, n_3)$	overall artifact level as a function of n_1, n_2 and n_3
$\mathbf{a}$	parameter set for normalization
$\hat{\mathbf{a}}$	parameter vector of coefficients of polynomial function
a_i	parameter for normalization
$\hat{a}_i$	optimal parameter for automatic feedback normalization
$\hat{\mathbf{a}}_{opt}$	optimal parameter set for automatic feedback normalization
$\hat{\mathbf{a}}_{total}$	parameter vector of coefficients and exponents of polynomial function
$\hat{\mathbf{a}}_{opt,fuzzy,norm}$	optimal parameter set based on fuzzy quality criterion for normalization
ANOVA	(univariate) ANalysis Of VAriances
AutoEdge	automatic feedback parameter adaptation of Sobel edge detection method
AutoOtsu	automatic feedback parameter adaptation of Otsu's method
B	total grades of grayscale shading in an image
$\mathcal{B}$	set defining all segments found in bright field channel
BF	Bright Field channel
$\mathcal{B}_i$	set containing pixel positions belonging to segments of bright field channel image
b	running index for grades of grayscale shading in an image
C_u	total number of pixels in $\mathbf{S}_u$
C_v	total number of pixels in $\mathbf{T}_v$
c	parameter defining the curvature (or exponential power of a function)
$\mathcal{D}$	set defining all segments found in nuclear staining channel
$\mathcal{DB}$	set containing all $\mathcal{DB}_i$
$\mathcal{DB}_i$	set containing union of all $\mathcal{D}_j$ sets
DAPI	nuclear staining channel
DAPIposBF	a set containing segments in DAPI channel that are also segmented in BF channel
$\mathcal{D}_j$	set containing pixel positions belonging to segments of nuclear staining channel image

Symbol	Description
$\mathcal{F}$	feature set of image objects
f	individual single feature of an image object
$f_{area,i}$	area of i^{th} BLOB
$f_{area,min}$	specified minimum size threshold
f_u	self-defined mathematical function
g	number of columns in a grayscale image
GUI	Graphical User Interface
h	total number of image object features
$\mathbf{I}$	grayscale image
$\mathbf{I}_A$	acquired grayscale image
$\mathbf{I}_{SF}$	shading-free grayscale image
$\mathbf{I}_{seg}$	segmented grayscale image
$\mathbf{I}_{truth}$	image with ground truth
i	running index for image rows
$\mathbf{i}_a^*$	vectorized form of $\mathbf{I}_A$
I_{ij}	pixel values for image $\mathbf{I}$ with indices i and j respectively
$I_{ij,max}$	maximum pixel value of an image $\mathbf{I}$
$I_{ij,min}$	minimum pixel value of an image $\mathbf{I}$
$I_{ij,norm}$	normalized pixel value of an image $\mathbf{I}$
$I_{high,b}$	upper intensity percentile of an image $\mathbf{I}$
$I_{low,a}$	lower intensity percentile of an image $\mathbf{I}$
$I_{ij,truth}$	pixel values of $\mathbf{I}_{truth}$
i_m	mean pixel value of an image $\mathbf{I}$
$I_{seg,ij}$	pixel value of $\mathbf{I}_{seg}$ at indices i and j
$\hat{\mathbf{i}}_{SF}$	vectorized form of $\mathbf{I}_{SF}$
j	running index for image columns
K_n	total number of object classes to be found in an image
K_u	class label for *uth* ground truth object
$\hat{K}_u$	class assignment for segmented object overlapping with *u-th* ground truth object
m	number of rows in a grayscale image
$\max\{\cdot\}$	maximum operator
$\mathrm{med}\{\cdot\}$	median operator
$\min\{\cdot\}$	minimum operator
MANOVA	(Multivariate) ANalysis Of VAriances
N	total noise levels induced in an image
N_d	total number of data points
n	running index for noise levels induced in an image

Symbol	Description
n_1	variance of Gaussian noise with zero mean
n_2	variance of Speckle noise
n_3	noise density of Salt and Pepper noise
n_b	total number of segments in bright field channel image
n_c	total number of segments detected in an image
n_d	total number of segments in nuclear staining channel image
n_j	criterion to select membership of segments belonging to both DAPI and BF channel images
n_{ref}	total number of reference segments in an image
n_seg	total number of reassigned segments per image
n_t	total number of segments detected in Hoechst channel
O	maximum number of objects to be found in an image
n_+	correctly reassigned segments
n_-	incorrectly reassigned segments
$\mathbf{p}$	parameters set to be tuned for image processing algorithm
$\mathbf{p}_{opt}$	optimal parameter set for segmentation based on segmentation quality criterion
$\mathbf{p}_{opt,ed}$	optimal parameter set based on Q_{ed}
$\mathbf{p}_{opt,fuzzy}$	optimal parameter set based on Q_{fuzz}
p_j	parameter assigning value to membership of segments that are present in both DAPI and BF channel images
p_x	a boundary threshold value for the selection between edge detection and Otsu's thresholding method
$pr_q(\cdot)$	percentile operator for lower *q-th* percentile
$pr_{100-q}(\cdot)$	percentile operator for upper *(100-q)-th* percentile
Q	overall quality measure of segmentation outcome of a certain segmentation algorithm
Q_{ed}	quality criterion for segmentation evaluation based on Euclidean distance between reference and segmented features
Q_{fuzz}	fuzzy quality criterion for automatic image segmentation based on fuzzy membership function features
$Q_{fuzz,norm}$	fuzzy quality criterion for normalization evaluation
$Q_{fuzz,seg}$	fuzzy quality criterion for segmentation evaluation
Q_{norm}	quality criterion for normalization evaluation
$Q(n_1, n_2, n_3)$	quality measure for segmentation evaluation when n_1, n_2 and n_3 are used
$Q(n_1, n_2, n_3, \mathbf{p})$	quality measure for segmentation evaluation obtained using a parameter set **p** during parametric tuning
Q_{seg}	segmentation quality criterion using μ_1 and μ_2
Q_{S1}	least square minimization criterion containing parameters only as coefficients

Symbol	Description
Q_{S2}	least square minimization criterion containing parameters only as exponents
q	percentile value
q_1	segmentation criterion defining the difference in total number of objects detected in comparison to ground truth
q_2	segmentation quality criterion defining non-overlapping pixels of the detected objects with the pixels of ground truth
q_3	quality criterion describing the class match between ground truth and segmented object
R	robustness value of an algorithms
R_o	maximum number of scenes in a benchmark image dataset
r	scene number in a benchmark image dataset
rnd	random number
$\mathcal{S}$	set containing all BLOBs of $\mathbf{I}_{seg}$
S_a	additive shading component in an image
$\hat{S}_a$	polynomial function for automatic feedback normalization
$\hat{\mathbf{S}}_a$	vectorized form of polynomial function for automatic feedback normalization
$\mathcal{S}_c$	set containing elements that fulfill a criterion based on the overlap of *u-th* BLOB set with all BLOB sets of T_v
S_m	multiplicative shading component in an image
$\mathbf{S}_u$	*u-th* BLOB of $\mathbf{I}_{seg}$ defined by its pixel indices
$S_{ux,i}$	row pixel index of $\mathbf{S}_u$
$S_{uy,i}$	column pixel index of $\mathbf{S}_u$
s	size of structuring element (predefined as a disc) in an image opening operation
$\hat{\mathbf{s}}^*_a$	vectorized form of $\hat{\mathbf{S}}_a$
SE	Structuring Element for morphological operations
StdEdge	standard feedforward Edge detection method
StdOtsu	standard feedforward Otsu segmentation method
$\mathcal{T}$	set containing all BLOBs of $\mathbf{I}_{truth}$
$\mathbf{T}_v$	*v-th* BLOB of $\mathbf{I}_{truth}$ defined by its pixel indices
$T_{vx,i}$	row pixel index of $\mathbf{T}_v$
$T_{vy,i}$	column pixel index of $\mathbf{T}_v$
t	image intensity threshold value
t_{in}	intensity threshold for i_{th} BLOB at level n
$t_{i,max}$	highest image threshold level for i_{th} BLOB
$t_{i,min}$	lowest image threshold level for i_{th} BLOB
U	total number of BLOBs found in $\mathbf{I}_{seg}$

Symbol	Description
u	running index for total number of BLOBs in $\mathbf{I}_{seg}$
V	total number of BLOBs found in $\mathbf{I}_{truth}$
v	running index for total number of BLOBs in $\mathbf{I}_{truth}$
w	convolution filter size defined as symmetric $w \times w$ square matrix having elements equal to $\frac{1}{w^2}$
$\hat{\mathbf{w}}$	vector of weights for weighting features of a given segment
$\mathbf{X}$	data matrix for additive noising of an image
x	index for x-coordinate of an image
$\mathbf{x}_i$	features vector for each segmented object *i* in an image
$\mathbf{x}_{ref}$	vector of alpha numeric references for each segment *j* in an image
y	index for y-coordinate of an image
α	upper bound of a monotonic function defined by μ
β	lower bound of a monotonic function defined by μ
$\delta_{truth,ij}$	binary pixel values of $\mathbf{I}_{truth}$ at indices *i* and *j*
$\delta_{seg,ij}$	binary pixel values of $\mathbf{I}_{seg}$ at indices *i* and *j*
Δ	difference of correct and incorrect number of segments after reassignment
θ	parameter defining image property to be used in a fuzzy membership function
σ	standard deviation for a normally distributed variable
σ_{GD}	deviation from ground truth in terms of number of segments
μ	fuzzy membership function
μ_c	fuzzy membership function for total object count in an image
$\mu_j(x_j)$	trapezoidal fuzzy membership function for each feature *j* of each segment
$\hat{\mu}_1$	fuzzy membership function of segmentation quality criterion q_1
$\hat{\mu}_2$	fuzzy membership function of segmentation quality criterion q_2
$\hat{\mu}_3$	fuzzy membership function of segmentation quality criterion q_3
$\hat{\mu}_1(\mathbf{p})$	fuzzy membership function of segmentation quality criterion q_1 obtained using a parameter set **p** during parametric tuning
$\hat{\mu}_1(\mathbf{p})$	fuzzy membership function of segmentation quality criterion q_2 obtained using a parameter set **p** during parametric tuning
$\hat{\mu}_1(\mathbf{p})$	fuzzy membership function of segmentation quality criterion q_3 obtained using a parameter set **p** during parametric tuning
ψ_{image}	percentage improvement in segmentation results with respect to the total number of segments detected originally in a given image
ψ_{case}	percentage improvement in segmentation results with respect to *critical cases*

Bibliography

[1] AL-KOFAHI, Y.; LASSOUED, W.; LEE, W.; ROYSAM, B.: Improved Automatic Detection and Segmentation of Cell Nuclei in Histopathology Images. *IEEE Transactions on Biomedical Engineering* 57 (2010) 4, S. 841–852.

[2] ALLAN, C.; BUREL, J.; MOORE, J.; BLACKBURN, C.; LINKERT, M.; LOYNTON, S.; MACDONALD, D.; MOORE, W.; NEVES, C.; PATTERSON, A.; PORTER, M.; TARKOWSKA, A.; LORANGER, B.; AVONDO, J.; LAGERSTEDT, I. AND. LIANAS, L.; LEO, S.; HANDS, K.; HAY, R. T.; PATWARDHAN, A.; BEST, C.; KLEYWEGT, G. J.; ZANETTI, G.; SWEDLOW, J. R.: OMERO: Flexible, Model-driven Data Management for Experimental Biology. *Nature Methods* 9 (2012) 3, S. 245–253.

[3] ANDERSON, E.: The Irises of the Gaspé Peninsula. *Bulletin of the American Iris Society* 59 (1935), S. 2–5.

[4] BAMFORD, P.: Empirical Comparison of Cell Segmentation Algorithms using an Annotated Dataset. In: *International Conference on Image Processing, 2003. ICIP 2003. Proceedings. 2003*, Bd. 2, S. II–1073, IEEE, 2003.

[5] BARTSCHAT, A.; HÜBNER, E.; REISCHL, M.; MIKUT, R.; STEGMAIER, J.: XPIWIT - An XML Pipeline Wrapper for the Insight Toolkit. *Bioinformatics* 32 (2016) 2, S. 315–317.

[6] BAUER, K. D.; DE LA TORRE-BUENO, J.; DIEL, I. J.; HAWES, D.; DECKER, W. J.; PRIDDY, C.; BOSSY, B.; LUDMANN, S.; YAMAMOTO, K.; MASIH, A. S.; ET AL.: Reliable and Sensitive Analysis of Occult Bone Marrow Metastases Using Automated Cellular Imaging. *Clinical Cancer Research* 6 (2000) 9, S. 3552–3559.

[7] BELLER, M.; STOTZKA, R.; MÜLLER, T.: Application of an Interactive Feature-Driven Segmentation. *Biomedizinische Technik* 49 (E2) (2004), S. 210–211.

[8] BERTHOLD, M. R.; CEBRON, N.; DILL, F.; GABRIEL, T. R.; KÖTTER, T.; MEINL, T.; OHL, P.; SIEB, C.; THIEL, K.; WISWEDEL, B.: KNIME: The Konstanz Information Miner. In: *Data Analysis, Machine Learning and Applications*, S. 319–326, Springer, 2008.

[9] BEUCHER, S.; MEYER, F.: *Mathematical Morphology in Image Processing*, Kap. The Morphological Approach to Segmentation: The Watershed Transformation., S. 433–481. 1993.

[10] BHANU, B.; LEE, S.; MING, J.: Adaptive Image Segmentation Using a Genetic Algorithm. In: *IEEE Transactions on Systems, Man and Cybernetics 25*, 1995.

[11] BLANCO, J. A.: In: Tree Growth: Influences, Layers and Types ISBN 978-1-60741-784-2 Editor: Wesley P. Karam© 2009 Nova Science Publishers, Inc. *Tree Growth: Influences, Layers and Types* (2009), S. 79.

[12] BOCK, R.; GROTE, H.; NOTZ, D.; REGLER, M.: *Data Analysis Techniques for High-Energy Physics Experiments*. Academic Press, 2009.

[13] VON BORSTEL, S.; MIKUT, R.; SCHMITZ, K.; REISCHL, M.: Feature Extraction and Plausibility Check of Spots in Small-molecule Microarray Images. *Biomedizinische Technik* 56 (2011) S1, S. 28ff.

[14] BROSNAN, T.; SUN, D.: Improving Quality Inspection of Food Products by Computer Vision - a Review. *Journal of Food Engineering* 61 (2004), S. 3–16.

[15] BUADES, A.; COLL, B.; MOREL, J.-M.: A Review of Image Denoising Algorithms, With a New One. *Multiscale Modeling & Simulation* 4 (2005) 2, S. 490–530.

[16] BUSTINCE, H.; PAGOLA, M.; JURIO, A.; BARRENECHEA, E.; FERNÁNDEZ, J.; COUTO, P.; MELO-PINTO, P.: A Survey of Applications of the Extensions of Fuzzy Sets to Image Processing. In: *Bio-inspired Hybrid Intelligent Systems for Image Analysis and Pattern Recognition*, S. 3–32, Springer, 2009.

[17] CANNY, J.: A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8, No. 6 (1986), S. 679–698.

[18] CARPENTER, A.; JONES, T.; LAMPRECHT, M.; CLARKE, C.; KANG, I.; FRIMAN, O.; GUERTIN, D.; CHANG, J.; LINDQUIST, R.; MOFFAT, J.; GOLLAND, P.; SABATINI, D. M.: CellProfiler: Image Analysis Software for Identifying and Quantifying Cell Phenotypes. *Genome Biology* 7 (2006) 10, S. R100.

[19] CASELLES, V.; KIMMEL, R.; SAPIRO, G.: Geodesic Active Contours. *International Journal of Computer Vision* 22 (1997) 1, S. 61–79.

[20] DE CHAUMONT, F.; DALLONGEVILLE, S.; CHENOUARD, N.; HERVÉ, N.; POP, S.; PROVOOST, T.; MEAS-YEDID, V.; PANKAJAKSHAN, P.; LECOMTE, T.; LE MONTAGNER, Y.; LAGACHE, T.; DUFOUR, A.; OLIVO-MARIN, J. C.: Icy: An Open Bioimage Informatics Platform for Extended Reproducible Research. *Nature Methods* 9 (2012) 7, S. 690–696.

[21] CHEN, Y.; LIN, W.; HAN, F.; YANG, Y.-H.; SAFAR, Z.; LIU, K.: A Cheat-Proof Game Theoretic Demand Response Scheme for Smart Grids. In: *IEEE International Conference on Communications (ICC)*, S. 3362–3366, 2012.

[22] CHENG, P.-C.: The Contrast Formation in Optical Microscopy. In: *Handbook Of Biological Confocal Microscopy*, S. 162–206, Springer, 2006.

[23] CHUN, D. N.; YANG, H. S.: Robust Image Segmentation Using Genetic Algorithm With a Fuzzy Measure. *Pattern Recognition* 29 (1996) 7, S. 1195–1211.

[24] COELHO, L.; SHARIFF, A.; MURPHY, R.: Nuclear Segmentation in Microscope Cell Images: A Hand-Segmented Dataset and Comparison of Algorithms. In: *IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2009. ISBI'09.*, S. 518–521, IEEE, 2009.

[25] DAM, E. B.; LOOG, M.: Efficient Segmentation by Sparse Pixel Classification. *Medical Imaging, IEEE Transactions on* 27 (2008) 10, S. 1525–1534.

[26] DEFINIENS, A.: Definiens Developer 7 User Guide. *München: Definiens AG* (2009).

[27] DENG, J.; DONG, W.; SOCHER, R.; LI, L.; LI, K.; FEI-FEI, L.: Imagenet: A Large-scale Hierarchical Image Database. In: *Proc., IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, S. 248–255, Ieee, 2009.

[28] DEY, V.; ZHANG, Y.; ZHONG, M.: A Review on Image Segmentation Techniques with Remote Sensing Perspective. In: *Proceedings of the International Society for Photogrammetry and Remote Sensing Symposium (ISPRS10)*, Bd. 38, S. 5–7, 2010.

[29] DI RUBERTO, C.; DEMPSTER, A.; KHAN, S.; JARRA, B.: Analysis of Infected Blood Cell Images Using Morphological Operators. *Image and Vision Computing* 20 (2002) 2, S. 133–146.

[30] DONAUER, J.; SCHRECK, I.; LIEBEL, U.; WEISS, C.: Role and Interaction of p53, BAX and the Stress-activated Protein Kinases p38 and JNK in Benzo(a)pyrene-diolepoxide Induced Apoptosis in Human Colon Carcinoma Cells. *Archives of Toxicology* 86(2) (2012), S. 329–337.

[31] DU, C.; SUN, D.: Recent Developments in the Applications of Image Processing Techniques for Food Quality Evaluation. *Trends in Food Science and Technology* 15 (2004), S. 230–249.

[32] EDELSTEIN, A.; AMODAJ, N.; HOOVER, K.; VALE, R.; STUURMAN, N.: Computer Control of Microscopes Using μManager. *Curr Protoc Mol Biol* (2010).

[33] EISENMANN, K.; HARRIS, E.; KITCHEN, S.; HOLMAN, H.; HIGGS, H.; ALBERTS, A. S.: Dia-interacting Protein Modulates Formin-mediatedactin Assembly at the Cell Cortex. *Curr. Biol* 17 (2007), S. 579–591.

[34] EL-SHANAWANY, A.: *The Development of Generative Bayesian Models for Classification of Cell Images*. Dissertation, UCL (University College London), 2010.

[35] ELICEIRI, K.; BERTHOLD, M.; GOLDBERG, I.; IBÁÑEZ, L.; MANJUNATH, B.; MARTONE, M.; MURPHY, R.; PENG, H.; PLANT, A.; ROYSAM, B.; STUURMAN, N.; SWEDLOW, J. R.; PAVEL TOMANCAK, P.; CARPENTER, A. E.: Biological Imaging Software Tools. *Nature Methods* 9 (2012) 7, S. 697–710.

[36] ESCALANTE, H. J.; HERNÁNDEZ, C. A.; GONZALEZ, J. A.; LÓPEZ-LÓPEZ, A.; MONTES, M.; MORALES, E. F.; ENRIQUE SUCAR, L.; VILLASEÑOR, L.; GRUBINGER, M.: The Segmented and Annotated IAPR TC-12 Benchmark. *Computer Vision and Image Understanding* 114 (2010) 4, S. 419–428.

[37] EVERINGHAM, M.; MULLER, H.; THOMAS, B.: Evaluating Image Segmentation Algorithms using the Pareto Front. In: *Computer Vision - ECCV 2002*, S. 34–48, Springer, 2002.

[38] FARMER, M.; JAIN, A.: A Wrapper-based Approach in Image Segmentation and Classification. *IEEE Transactions on Image Processing* 14 (12) (2005), S. 2060–2072.

[39] FENECH, M.: Cytokinesis-block Micronucleus Cytome Assay. *Nature Protocols* 2 (2007) 5, S. 1084–1104.

[40] FENG, D.; MARSHBURN, D.; JEN, D.; WEINBERG, R.; TAYLOR II, R.; BURETTE, A.: Stepping into the Third Dimension. *The Journal of Neuroscience* 27 (2007) 47, S. 12757–12760.

[41] FIALA, J.: Reconstruct: A Free Editor for Serial Section Microscopy. *Journal of Microscopy* 218 (2005) 1, S. 52–61.

[42] FORINA, M. ET AL: PARVUS - An Extendible Package for Data Exploration, Classification, and Correlation. Institute of Pharmaceutical and Food Analysis and Technologies, Via Brigata Salerno, 16147 Genoa, Italy, 1991.

[43] FU, K.; MUI, J.: A Survey on Image Segmentation. *Pattern Recognition* 13 (1981) 10, S. 845–854.

[44] GELASCA, E. D.; BYUN, J.; OBARA, B.; MANJUNATH, B.: Evaluation and Benchmark for Biological Image Segmentation. In: *IEEE International Conference on Image Processing*, 2008.

[45] GENÇTAV, A.; AKSOY, S.; ÖNDER, S.: Unsupervised Segmentation and Classification of Cervical Cell Images. *Pattern Recognition* 45 (2012) 12, S. 4151–4168.

[46] GOLDBERG, I.; ALLAN, C.; BUREL, J.; CREAGER, D.; FALCONI, A.; HOCHHEISER, H.; JOHNSTON, J.; MELLEN, J.; SORGER, P.; SWEDLOW, J.: The Open Microscopy Environment (OME) Data Model and XML File: Open Tools for Informatics and Quantitative Analysis in Biological Imaging. *Genome Biology* 6 (2005) 5.

[47] GOLSTEIN, P.; KROEMER, G.: Cell Death by Necrosis: Towards a Molecular Definition. *Trends in Biochemical Sciences* 32 (2007) 1, S. 37–43.

[48] GONZALEZ, R. C.; WOODS, R. E.; EDDINS, S. L.: *Digital Image Processing Using MATLAB*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 2003.

[49] GRADY, L.: Random Walks for Image Segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 28 (2006) 11, S. 1768–1783.

[50] GREEN, J.; COLLINS, C.; KYZAR, E.; PHAM, M.; ROTH, A.; GAIKWAD, S.; CACHAT, J.; STEWART, A.; LANDSMAN, S.; GRIECO, F.; ET AL.: Automated High-throughput Neurophenotyping of Zebrafish Social Behavior. *Journal of Neuroscience Methods* 210 (2012) 2, S. 266–271.

[51] GRIGORESCU, S.; RISTIC-DURRANT, D.; VUPPALA, S.; GRÄSER, A.: Closed-loop Control in Image Processing for Improvement of Object Recognition. In: *Proc., 17th IFAC World Congress*, 2008.

[52] GUTHAUS, M. R.; RINGENBERG, J. S.; ERNST, D.; AUSTIN, T. M.; MUDGE, T.; BROWN, R. B.: MiBench: A Free, Commercially Representative Embedded Benchmark Suite. In: *IEEE International Workshop on Workload Characterization, 2001. WWC-4. 2001*, S. 3–14, IEEE, 2001.

[53] H.-J-BOEHME; BAUMANN, U.; CORRADINI, A.; GROSS, H.: Person Localization and Posture Recognition for Human-Robot-Interaction. In: *Gesture Based Communication in Human-Computer Interaction* (ET AL., A. B., Hg.), Springer Verlag, 1999.

[54] HALL, E. L.; KRUGER, R. P.; DWYER, S. J.; HALL, D. L.; MCLAREN, R. W.; LODWICK, G. S.: A Survey of Preprocessing and Feature Extraction Techniques for Radiographic Images. *IEEE Transactions on Computers* 100 (1971) 9, S. 1032–1044.

[55] VAN HAM, T. J.; MAPES, J.; KOKEL, D.; PETERSON, R. T.: Live Imaging of Apoptotic Cells in Zebrafish. *The FASEB Journal* 24 (2010) 11, S. 4336–4342.

[56] HANAHAN, D.; WEINBERG, R. A.: Hallmarks of Cancer: The Next Generation. *Cell* 144 (2011) 5, S. 646–674.

[57] HARDER, N.; BATRA, R.; DIESSL, N.; GOGOLIN, S.; EILS, R.; WESTERMANN, F.; KÖNIG, R.; ROHR, K.: Large-scale Tracking and Classification for Automatic Analysis of Cell Migration and Proliferation, and Experimental Optimization of High-throughput Screens of Neuroblastoma Cells. *Cytometry Part A* (2015).

[58] HARDER, N.; EILS, R.; ROHR, K.: Automated Classification of Mitotic Phenotypes of Human Cells using Fluorescent Proteins. *Methods in Cell Biology* 85 (2008), S. 539–554.

[59] HARDER, N.; MORA-BERMÚDEZ, F.; GODINEZ, W. J.; WÜNSCHE, A.; EILS, R.; ELLENBERG, J.; ROHR, K.: Automatic Analysis of Dividing Cells in Live Cell Movies to Detect Mitotic Delays and Correlate Phenotypes in Time. *Genome Research* 19 (2009) 11, S. 2113–2124.

[60] HAYRAPETYAN, N.: *Adaptive Re-segmentation Strategies for Accurate Bright Field Cell Tracking*. Dissertation, Utah State University, 2012.

[61] HELD, M.; SCHMITZ, M. H.; FISCHER, B.; WALTER, T.; NEUMANN, B.; OLMA, M. H.; PETER, M.; ELLENBERG, J.; GERLICH, D. W.: CellCognition: Time-Resolved Phenotype Annotation in High-Throughput Live Cell Imaging. *Nature Methods* 7 (2010) 9, S. 747–754.

[62] HOFMANN, P.; BLASCHKE, T.; STROBL, J.: Quantifying the Robustness of Fuzzy Rule Sets in Object-based Image Analysis. *International Journal of Remote Sensing* 32 (2011) 22, S. 7359–7381.

[63] HSU, C.-T.; LI, C.-Y.: Relevance Feedback Using Generalized Bayesian Framework with Region-based Optimization Learning. *IEEE Transactions on Image Processing* 14 (2005) 10, S. 1617–1631.

[64] HUH, S.; KER, D.; BISE, R.; CHEN, M.; KANADE, T.: Automated Mitosis Detection of Stem Cell Populations in Phase-Contrast Microscopy Images. *IEEE Transactions on Medical Imaging* 30 (2011) 3, S. 586–596.

[65] IBANEZ, L.; SCHROEDER, W.; NG, L.; CATES, J.; ET AL.: *The ITK Software Guide*. Citeseer, 2005.

[66] JAN, E.; BYRNE, S. J.; CUDDIHY, M.; DAVIES, A. M.; VOLKOV, Y.; GUNKO, Y. K.; KOTOV, N. A.: High-content Screening as a Universal Tool for Fingerprinting of Cytotoxicity of Nanoparticles. *ACS Nano* 2 (2008) 5, S. 928–938.

[67] KAMAVISDAR, P.; SALUJA, S.; AGRAWAL, S.: A Survey on Image Classification Approaches and Techniques. *International Journal of Advanced Research in Computer and Communication Engineering* 2 (2013) 1.

[68] KANKAANPÄÄ, P.; PAAVOLAINEN, L.; TIITTA, S.; KARJALAINEN, M.; PÄIVÄRINNE, J.; NIEMINEN, J.; MARJOMÄKI, V.; HEINO, J.; WHITE, D.: BioImageXD: An Open, General-Purpose and High-Throughput Image-Processing Platform. *Nature Methods* 9 (2012) 7, S. 683–689.

[69] KASS, M.; WITKIN, D.; TERZOPOULOS: Snakes: Active Contour Models. *International Journal for Computer Vision* 1 (1988), S. 321–331.

[70] KAVOUSSANOS, M.; POULIEZZOS: An Intelligent Robotic System for Depalletizing and Emptying Polyethylene Sacks. *International Journal of Advanced Manufacturing Technology* 14 (1998), S. 358–362.

[71] KEPP, O.; GALLUZZI, L.; LIPINSKI, M.; YUAN, J.; KROEMER, G.: Cell Death Assays for Drug Discovery. *Nature Reviews Drug Discovery* 10 (2011) 3, S. 221–237.

[72] KHAN, A.; MIKUT, R.; SCHWEITZER, B.; WEISS, C.; REISCHL, M.: Automatic Tuning of Image Segmentation Parameters by Means of Fuzzy Feature Evaluation. *Synergies of Soft Computing and Statistics for Intelligent Data Analysis* (2012), S. 459–467.

[73] KHAN, A.; REISCHL, M.; SCHWEITZER, B.; WEISS, C.; MIKUT, R.: Feedback-Driven Design of Normalization Techniques for Biological Images Using Fuzzy Formulation of a Priori Knowledge. *Studies in Computational Intelligence* 445 (2013), S. 167–178.

[74] KHAN, A.; WEISS, C.; SCHWEITZER, B.; HANSJOSTEN, I.; MIKUT, R.; REISCHL, M.: Multimodal Image Segmentation of Cellular Fragmentation Using Edge Detector and Morphological Operators. *Biomedizinische Technik - Biomedical Engineering* 59 (2014) S1, S. S518–S521.

[75] KHAN, A. U. M.; MIKUT, R.; REISCHL, M.: A Benchmark Data Set to Evaluate the Illumination Robustness of Image Processing Algorithms for Object Segmentation and Classification. *PLoS ONE* 10 (2015) 7, S. e0131098.

[76] KHAN, A. U. M.; MIKUT, R.; REISCHL, M.: A New Feedback-Based Method for Parameter Adaptation in Image Processing Routines. *PLOS ONE* 11 (2016) 10, S. 1–23.

[77] KIM, D.; CHAE, Y.-S.; KIM, S. J.: High Content Cellular Analysis for Functional Screening of Novel Cell Cycle Related Genes. In: *BioMedical Engineering and Informatics, 2008. BMEI 2008. International Conference on*, Bd. 1, S. 148–152, IEEE, 2008.

[78] KREMER, J.; MASTRONARDE, D.; MCINTOSH, J.; ET AL.: Computer Visualization of Three-dimensional Image Data using IMOD. *Journal of Structural Biology* 116 (1996) 1, S. 71–76.

[79] LASSOUAOUI, N.; HAMAMI, L.; ZERGUERRAS, A.: Segmentation and Classification of Biological Cell Images by a Multifractal Approach. *International Journal of Intelligent Systems* 18 (2003) 6, S. 657–678.

[80] LEE, S. U.; YOON CHUNG, S.; PARK, R. H.: A Comparative Performance Study of Several Global Thresholding Techniques for Segmentation. *Computer Vision, Graphics, and Image Processing* 52 (1990) 2, S. 171–190.

[81] LI, J.: *Division Tracking By Live Cell Imaging and Image Processing*. Dissertation, University of New South Wales, 2011.

[82] LIESCHKE, G. J.; CURRIE, P. D.: Animal Models of Human Disease: Zebrafish Swim into View. *Nature Reviews. Genetics* 8 (2007) 5, S. 353–67, pMID: 17440532.

[83] LIKAR, B.; MAINTZ, J.; VIERGEVER, M.; PERNUS, F.; ET AL.: Retrospective Shading Correction based on Entropy Minimization. *Journal of Microscopy* 197 (2000) 3, S. 285–295.

[84] LIU, T.; LI, G.; NIE, J.; TAROKH, A.; ZHOU, X.; GUO, L.; MALICKI, J.; XIA, W.; WONG, S.: An Automated Method for Cell Detection in Zebrafish. *Neuroinformatics* 6 (2008) 1, S. 5–21.

[85] LOU, X.; KOETHE, U.; WITTBRODT, J.; HAMPRECHT, F.: Learning to Segment Dense Cell Nuclei with Shape Prior. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012*, S. 1012–1018, IEEE, 2012.

[86] LU, D.; WENG, Q.: A Survey of Image Classification Methods and Techniques for Improving Classification Performance. *International Journal of Remote Sensing* 28 (2007) 5, S. 823–870.

[87] MAI, F.; CHANG, C.; LIU, W.; XU, W.; HUNG, Y.: Segmentation-based Retrospective Shading Correction in Fluorescence Microscopy *E. coli* Images for Quantitative Analysis. In: *Proc., International Symposium on Multispectral Image Processing and Pattern Recognition*, Bd. 7498, S. 74983O–1, 2009.

[88] MALAMAS, E.; PETRAKIS, E.; ZERVAKIS, M.; PETIT, L.; LEGAT, J.: A Survey on Industrial Vision Systems, Applications and Tools. *Image and Vision Computing* 21(2) (2003), S. 171–188.

[89] MCINERNEY, T.; TERZOPOULOS, D.: Deformable Models in Medical Image Analysis: A Survey. *Medical Image Analysis* 1(2) (1996), S. 91–108.

[90] MEZARIS, V.; KOMPATSIARIS, I.; STRINTZIS, M.: Still Image Objective Segmentation Evaluation using Ground Truth. In: *Fifth COST 276 Workshop on Information and Knowledge Management for Integrated Media Communication*, S. 9–14, 2003.

[91] MIKULA, K.; PEYRIÉERAS, N.; REMEŠÍKOVÁ, M.; SMÍŠEK, M.: 4D Numerical Schemes for Cell Image Segmentation and Tracking. *Finite Volumes for Complex Applications VI-Problems & Perspectives* (2011), S. 693–701.

[92] MIKULA, K.; REMEŠÍKOVÁ, M.; STAŠOVÁ, O.; PEYRIÉRAS, N.: Segmentation and Analysis of 3D Zebrafish Cell Image Data. In: *3rd International Congress on Image and Signal Processing (CISP), 2010*, Bd. 3, S. 1444–1448, IEEE, 2010.

[93] MIKUT, R.: *Data Mining in der Medizin und Medizintechnik*. Universitätsverlag Karlsruhe, 2008.

[94] MIKUT, R.; BURMEISTER, O.; BRAUN, S.; REISCHL, M.: The Open Source Matlab Toolbox Gait-CAD and its Application to Bioelectric Signal Processing. In: *Proc., DGBMT-Workshop Biosignalverarbeitung, Potsdam*, S. 109–111, 2008.

[95] NASSE, M. J.; WOEHL, J. C.: Realistic Modeling of the Illumination Point Spread Function in Confocal Scanning Optical Microscopy. *JOSA A* 27 (2010) 2, S. 295–302.

[96] NAVALTA, J. W.; MOHAMED, R.; EL-BAZ, A.; MCFARLIN, B. K.; LYONS, T. S.: Exercise-induced Immune Cell Apoptosis: Image-based Model for Morphological Assessment. *Eur J Appl Physiol* 110 (2010) 2, S. 325–331.

[97] NEUMANN, B.; HELD, M.; LIEBEL, U.; ERFLE, H.; ROGERS, P.; PEPPERKOK, R.; ELLENBERG, J.: High-throughput RNAi Screening by Time-lapse Imaging of Live Human Cells. *Nature methods* 3 (2006) 5, S. 385–390.

[98] NEUMANN, B.; WALTER, T.; HÉRICHÉ, J.-K.; BULKESCHER, J.; ERFLE, H.; CONRAD, C.; ROGERS, P.; POSER, I.; HELD, M.; LIEBEL, U.; ET AL.: Phenotypic Profiling of the Human Genome by Time-lapse Microscopy Reveals Cell Division Genes. *Nature* 464 (2010) 7289, S. 721–727.

[99] NICKEL, K.; STIEFELHAGEN, R.: Real-time Person Tracking and Pointing Gesture Recognition for Human-Robot Interaction. In: *Computer Vision in Human Computer Interaction*, 2004.

[100] NÜSSLEIN-VOLHARD, C.; WIESCHAUS, E.: Mutations Affecting Segment Number and Polarity in Drosophila. *Nature* 287 (1980) 5785, S. 795–801.

[101] OBUCHOWICZ, A.; HREBIEŃ, M.; NIECZKOWSKI, T.; MARCINIAK, A.: Computational Intelligence Techniques in Image Segmentation for Cytopathology. In: *Computational Intelligence in Biomedicine and Bioinformatics*, S. 169–199, Springer, 2008.

[102] OH, S.; HOOGS, A.; PERERA, A.; CUNTOOR, N.; CHEN, C.-C.; LEE, J. T.; MUKHERJEE, S.; AGGARWAL, J.; LEE, H.; DAVIS, L.; ET AL.: A Large-scale Benchmark Dataset for Event Recognition in Surveillance Video. In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, S. 3153–3160, IEEE, 2011.

[103] OTSU, N.: A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man and Cybernetics* 9 (1979), S. 62–66.

[104] PAPADEMETRIS, X.; JACKOWSKI, M.; RAJEEVAN, N.; CONSTABLE, R. T.; STAIB, L.: BioImage Suite: An Integrated Medical Image Analysis Suite. *The Insight Journal* (2005).

[105] PAPPAS, T. N.: An Adaptive Clustering Algorithm for Image Segmentation. *IEEE Transactions on Signal Processing* 40 (1992) 4, S. 901–914.

[106] PATIL, G.; CHAUDHARI, D.: SIFT Based Approach: Object Recognition and Localization for Pick-and-Place System. *International Journal of Advanced Research in Computer Science and Software Engineering* 3 (3) (2013), S. 196–201.

[107] PAU, G.; WALTER, T.; NEUMANN, B.; HÉRICHÉ, J.-K.; ELLENBERG, J.; HUBER, W.: Dynamical Modelling of Phenotypes in a Genome-wide RNAi Live-cell Imaging Assay. *BMC Bioinformatics* 14 (2013) 1, S. 308.

[108] PENG, H.; RUAN, Z.; LONG, F.; SIMPSON, J.; MYERS, E.: V3D Enables Real-time 3D Visualization and Quantitative Analysis of Large-scale Biological Image Data Sets. *Nature Biotechnology* 28 (2010) 4, S. 348–353.

[109] PENG, J.-Y.; CHEN, Y.-J.; GREEN, M. D.; SABATINOS, S. A.; FORSBURG, S. L.; HSU, C.-N.: PombeX: Robust Cell Segmentation for Fission Yeast Transillumination Images. *PLoS ONE* 8 (2013) 12, S. e81434.

[110] PENG, J.-Y.; LIN, C.-C.; CHEN, Y.-J.; KAO, L.-S.; LIU, Y.-C.; CHOU, C.-C.; HUANG, Y.-H.; CHANG, F.-R.; WU, Y.-C.; TSAI, Y.-S.; ET AL.: Automatic Morphological Subtyping Reveals New Roles of Caspases in Mitochondrial Dynamics. *PLoS Computational Biology* 7 (2011) 10, S. e1002212.

[111] PINIDIYAARACHCHI, A.; WÄHLBY, C.: Seeded Watersheds for Combined Segmentation and Tracking of Cells. In: *Image Analysis and Processing–ICIAP 2005*, S. 336–343, Springer, 2005.

[112] PLÄSIER, B.; LLOYD, D. R.; PAUL, G. C.; THOMAS, C. R.; AL-RUBEAI, M.: Automatic Image Analysis for Quantification of Apoptosis in Animal Cell Culture by Annexin-V Affinity Assay. *J Immunol Methods* 229 (1999) 1-2, S. 81–95.

[113] POLOGRUTO, T. A.; SABATINI, B. L.; SVOBODA, K.: ScanImage: Flexible Software for Operating Laser Scanning Microscopes. *Biomed. Eng. Online* 2 (2003) 1, S. 13.

[114] REISCHL, M.; ALSHUT, R.; MIKUT, R.: On Robust Feature Extraction and Classification of Inhomogeneous Data Sets. In: *Proc., 20. Workshop Computational Intelligence*, S. 124–143, KIT Scientific Publishing, 2010.

[115] REYES-ALDASORO, C. C.: Retrospective Shading Correction Algorithm Based on Signal Envelope Estimation. *Electronics Letters* 45 (2009) 9, S. 454–456.

[116] RIZZI, B.; CAMPANA, M.; ZANELLA, C.; MELANI, C.; CUNDERLIK, R.; KRIVA, Z.; BOURGINE, P.; MIKULA, K.; PEYRIERAS, N.; SARTI, A.: 3D Zebrafish Embryo Image Filtering by Nonlinear Partial Differential Equations. In: *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, S. 6251–6254, 2007.

[117] ROSSET, A.; SPADOLA, L.; RATIB, O.: OsiriX: An Open-source Software for Navigating in Multidimensional DICOM Images. *Journal of Digital Imaging* 17 (2004) 3, S. 205–216.

[118] RUEDEN, C.; ELICEIRI, K. W.; WHITE, J. G.: VisBio: A Computational Tool for Visualization of Multidimensional Biological Image Data. *Traffic* 5 (2004) 6, S. 411–417.

[119] Ruusuvuori, P.; Lehmussola, A.; Selinummi, J.; Rajala, T.; Huttunen, H.; Yli-Harja, O.: Benchmark set of synthetic images for validating cell image analysis algorithms. In: *Proceedings of the 16th European Signal Processing Conference, EUSIPCO*, 2008.

[120] Schindelin, J.; Arganda-Carreras, I.; Frise, E.; Kaynig, V.; Longair, M.; Pietzsch, T.; Preibisch, S.; Rueden, C.; Saalfeld, S.; Schmid, B.; Tinevez, J.-Y.; White, D. J.; Hartenstein, V.; Eliceiri, K.; Tomancak, P.; Cardona, A.: Fiji: An Open-Source Platform for Biological-Image Analysis. *Nature Methods* 9 (2012) 7, S. 676–682.

[121] Scholtens, T. M.: *Automated Classification and Enhanced Characterization of Circulating Tumor Cells by Image Cytometry*. Universiteit Twente, 2012.

[122] Scorcioni, R.; Lazarewicz, M. T.; Ascoli, G. A.: Quantitative Morphometry of Hippocampal Pyramidal cells: Differences Between Anatomical Classes and Reconstructing Laboratories. *Journal of Comparative Neurology* 473 (2004) 2, S. 177–193.

[123] Sethian, J. A.: A Fast Marching Level Set Method for Monotonically Advancing Fronts. *Proceedings of the National Academy of Sciences* 93 (1996) 4, S. 1591–1595.

[124] Sezan, M. I.; Tekalp, A. M.: Survey of Recent Developments in Digital Image Restoration. *Optical Engineering* 29 (1990) 5, S. 393–404.

[125] Shen, S.; Kepp, O.; Michaud, M.; Martins, I.; Minoux, H.; Metivier, D.; Maiuri, M.; Kroemer, R.; Kroemer, G.: Association and Dissociation of Autophagy, Apoptosis and Necrosis by Systematic Chemical Study. *Oncogene* 30 (2011) 45, S. 4544–4556.

[126] Shih, F. Y.; Cheng, S.: Automatic Seeded Region Growing for Color Image Segmentation. *Image and Vision Computing* 23 (2005) 10, S. 877–886.

[127] Shin, M. C.; Goldgof, D. B.; Bowyer, K. W.: Comparison of Edge Detector Performance Through Use in an Object Recognition Task. *Computer Vision and Image Understanding* 84 (2001) 1, S. 160–178.

[128] Siang Tan, K.; Mat Isa, N. A.: Color Image Segmentation Using Histogram Thresholding–Fuzzy C-means Hybrid Approach. *Pattern Recognition* 44 (2011) 1, S. 1–15.

[129] Slattery, S. D.; Newberg, J. Y.; Szafran, A. T.; Hall, R. M.; Brinkley, B. R.; Mancini, M. A.: A Framework for Image-Based Classification of Mitotic Cells in Asynchronous Populations. *Assay and drug development technologies* 10 (2012) 2, S. 161–178.

[130] Smith, K.; Li, Y.; Piccinini, F.; Csucs, G.; Balazs, C.; Bevilacqua, A.; Horvath, P.: CIDRE: An Illumination-correction Method for Optical Microscopy. *Nature Methods* 12 (2015) 5, S. 404–406.

[131] SOMMER, C.; GERLICH, D. W.: Machine Learning in Cell Biology – Teaching Computers to Recognize Phenotypes. *Journal of Cell Science* 126 (2013) 24, S. 5529–5539.

[132] SOMMER, C.; STRAEHLE, C.; KOTHE, U.; HAMPRECHT, F.: ilastik: Interactive Learning and Segmentation Toolkit. In: *Proc., IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, S. 230–233, IEEE, 2011.

[133] SPANN, M.; WILSON, R.: A Quad-tree Approach to Image Segmentation Which Combines Statistical and Spatial Information. *Pattern Recognition* 18 (1985) 3-4, S. 257–269.

[134] STALLKAMP, J.; SCHLIPSING, M.; SALMEN, J.; IGEL, C.: The German Traffic Sign Recognition Benchmark: A Multi-class Classification Competition. In: *The 2011 International Joint Conference on Neural Networks (IJCNN)*, S. 1453–1460, IEEE, 2011.

[135] STEGMAIER, J.: *New Methods to Improve Large-Scale Microscopy Image Analysis with Prior Knowledge and Uncertainty*. Dissertation, Karlsruhe Institute of Technology, 2016.

[136] STEGMAIER, J.; ALSHUT, R.; REISCHL, M.; MIKUT, R.: Information Fusion of Image Analysis, Video Object Tracking, and Data Mining of Biological Images using the Open Source MATLAB Toolbox Gait-CAD. *Biomedizinische Technik (Biomedical Engineering)* 57 (S1) (2012), S. 458–461.

[137] STEGMAIER, J.; KHAN, A.; REISCHL, M.; MIKUT, R.: Challenges of Uncertainty Propagation in Image Analysis. In: *Proc., 22. Workshop Computational Intelligence, Dortmund*, S. 55–69, 2012.

[138] STEGMAIER, J.; MIKUT, R.: Fuzzy-based Propagation of Prior Knowledge to Improve Large-Scale Image Analysis Pipelines. arxiv.org, submitted paper, 2016.

[139] STEGMAIER, J.; OTTE, J. C.; KOBITSKI, A.; BARTSCHAT, A.; GARCIA, A.; NIENHAUS, G. U.; STRÄHLE, U.; MIKUT, R.: Fast Segmentation of Stained Nuclei in Terabyte-Scale, Time Resolved 3D Microscopy Image Stacks. *PLoS ONE* 9 (2014) 2, S. e90036.

[140] SVOBODA, D.; KAŠÍK, M.; MAŠKA, M.; HUBENỲ, J.; STEJSKAL, S.; ZIMMERMANN, M.: On Simulating 3D Fluorescent Microscope Images. In: *Computer Analysis of Images and Patterns*, S. 309–316, Springer, 2007.

[141] SVOBODA, D.; KOZUBEK, M.; STEJSKAL, S.: Generation of Digital Phantoms of Cell Nuclei and Simulation of Image Formation in 3D Image Cytometry. *Cytometry Part A* 75 (2009) 6, S. 494–509.

[142] SVOBODA, D.; ULMAN, V.: Generation of Synthetic Image Datasets for Time-lapse Fluorescence Microscopy. In: *Image Analysis and Recognition*, S. 473–482, Springer, 2012.

[143] SZÉNÁSI, S.; VÁMOSSY, Z.: Implementation of a Distributed Genetic Algorithm for Parameter Optimization in a Cell Nuclei Detection Project. *Acta Polytechnica Hungarica* 10 (2013) 4, S. 59–86.

[144] TOMAŽEVIČ, D.; LIKAR, B.; PERNUŠ, F.: Comparative Evaluation of Retrospective Shading Correction Methods. *Journal of Microscopy* 208 (2002) 3, S. 212–223.

[145] UDUPA, J. K.; LEBLANC, V. R.; ZHUGE, Y.; IMIELINSKA, C.; SCHMIDT, H.; CURRIE, L. M.; HIRSCH, B. E.; WOODBURN, J.: A Framework for Evaluating Image Segmentation Algorithms. *Computerized Medical Imaging and Graphics* 30 (2006) 2, S. 75–87.

[146] VINCENT, L.; SOILLE, P.: Watersheds in Digital Spaces: An Efficient Algorithm based on Immersion Simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13 (1991) 6, S. 583–598.

[147] VOVK, U.; PERNUS, F.; LIKAR, B.: A Review of Methods for Correction of Intensity Inhomogeneity in MRI. *IEEE Transactions on Medical Imaging* 26 (2007) 3, S. 405–421.

[148] WALTER, T.; SHATTUCK, D. W.; BALDOCK, R.; BASTIN, M. E.; CARPENTER, A. E.; DUCE, S.; ELLENBERG, J.; FRASER, A.; HAMILTON, N.; PIEPER, S.; ET AL.: Visualization of Image Data from Cells to Organisms. *Nature Methods* 7 (2010), S. S26–S41.

[149] WAN, Y.; OTSUNA, H.; CHIEN, C.-B.; HANSEN, C.: FluoRender: An Application of 2D Image Space Methods for 3D and 4D Confocal Microscopy Data Visualization in Neurobiology Research. In: *Pacific Visualization Symposium (PacificVis), 2012 IEEE*, S. 201–208, IEEE, 2012.

[150] WANG, W. H.; LIU, X. Y.; SUN, Y.: Robust Contact Detection in Micromanipulation Using Computer Vision Microscopy. In: *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE*, S. 2219–2222, 2006.

[151] YASNOFF, W. A.; BACUS, J.: Scene-segmentation Algorithm Development Using Error Measures. *Analytical and Quantitative Cytology* 6 (1984) 1, S. 45–58.

[152] YASNOFF, W. A.; MUI, J. K.; BACUS, J. W.: Error Measures for Scene Segmentation. *Pattern Recognition* 9 (1977) 4, S. 217–231.

[153] YUKSEL, M.; BORLU, M.: Accurate Segmentation of Dermoscopic Images by Image Thresholding Based on Type-2 Fuzzy Logic. *IEEE Transactions on Fuzzy Systems* 17 (2009) 4, S. 976–982.

[154] ZANELLA, C.; CAMPANA, M.; RIZZI, B.; MELANI, C.; SANGUINETTI, G.; BOURGINE, P.; MIKULA, K.; PEYRIÉRAS, N.; SARTI, A.: Cells Segmentation from 3D Confocal Images of Early Zebrafish Embryogenesis. *IEEE Transactions on Image Processing* 19 (2010) 3, S. 770–781.

[155] Zanella, C.; Rizzi, B.; Melani, C.; Campana, M.; Bourgine, P.; Mikula, K.; Peyrieras, N.; Sarti, A.: Segmentation of Cells from 3D Confocal Images of Live Zebrafish Embryo. In: *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, S. 6027–6030, 2007.

[156] Zhang, H.; Fritts, J.; Goldman, S.: Image Segmentation Evaluation: A Survey of Unsupervised Methods. *Computer Vision and Image Understanding* 110 (2008) 2, S. 260–280.

[157] Zhang, W.; Morris, Q.; Chang, R.; Shai, O.; Bakowski, M.; Mitsakakis, N.; Mohammad, N.; Robinson, M.; Zirngibl, R.; Somogyi, E.; et al.: The Functional Landscape of Mouse Gene Expression. *Journal of Biology* 3 (2004) 5, S. 21.

[158] Zhang, Y.; Gerbrands, J.: Objective and Quantitative Segmentation Evaluation and Comparison. *Signal Processing* 39 (1994) 1, S. 43–54.

[159] Zhang, Y. J.: A Review of Recent Evaluation Methods for Image Segmentation. In: *Proc. Symp Signal Processing and its Applications, Sixth Int. . 2001*, Bd. 1, S. 148–151, 2001.

[160] Zhou, X.; Li, F.; Yan, J.; Wong, S. T.: A Novel Cell Segmentation Method and Cell Phase Identification Using Markov Model. *IEEE Transactions on Information Technology in Biomedicine* 13 (2009) 2, S. 152–157.

[161] Zhou, X.; Liu, K.-Y.; Bradley, P.; Perrimon, N.; Wong, S. T.: Towards Automated Cellular Image Segmentation for RNAi Genome-wide Screening. In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2005*, S. 885–892, Springer, 2005.

[162] Zwier, J.; Van Rooij, G.; Hofstraat, J.; Brakenhoff, G.: Image Calibration in Fluorescence Microscopy. *Journal of Microscopy* 216 (2004) 1, S. 15–24.