AUTOMATED CELL ANALYSIS IN MICROSCOPY IMAGES

A DISSERTATION SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN
COMPUTER ENGINEERING

By
Can Fahrettin Koyuncu
September 2018

AUTOMATED CELL ANALYSIS IN MICROSCOPY IMAGES
By Can Fahrettin Koyuncu
September 2018

We certify that we have read this dissertation and that in our opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Çiğdem Gündüz Demir(Advisor)

Rengül Atalay

Can Alkan

Ramazan Gökberk Cinbiş

Selim Aksoy

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

AUTOMATED CELL ANALYSIS IN MICROSCOPY IMAGES

Can Fahrettin Koyuncu
Ph.D. in Computer Engineering
Advisor: Çiğdem Gündüz Demir
September 2018

High-throughput microscopy systems have become popular recently, which facilitate to acquire boundless microscopy images without requiring human intervention. However, the analysis of such amount of images using conventional methods is nearly impractical since the analysis can take up to months. Additionally, a considerable amount of observer variability may occur since the analysis completely relies on interpretation of the analysts. As a remedy for that, automated decision support systems, which are objective and rapid, have gained more attention. Since these systems conduct analyses at cellular level, they require a cell segmentation model, results of which directly affect the performance of the entire system.

There are several challenges in cell segmentation, each of which should be addressed carefully in order to have an accurate cell segmentation model. One challenge is that cells can be grown in multilayer on the plate, which makes them appear as clusters on the image. Segmentation of these cells requires extra effort since they should be splitted from each other. Another challenge is the imperfections on the image such as inhomogeneities of pixel intensities in a cell and insufficient pixel intensity differences at the border of overlapping cells. Yet-another challenge is the heterogeneity in the morphological characteristics of cells. Depending on cell line types, cells may appear in various outlooks. Developing a generic cell segmentation model, which can handle different cells' outlooks and imperfections, is an open and challenging problem.

In order to tackle with these challenges, we deal with the cell segmentation problem in two parts: (1) We focus on finding a new representation for microscopy images, helping us simplify the cell segmentation problem, so that imperfections in cells and inhomogeneities in their visual properties can be alleviated, and cell locations can be emphasized better. (2) We focus on developing a more advanced cell segmentation method, with the motivation that it is almost impossible to obtain a perfect representation in practice. Thus, we work on developing

more sophisticated cell segmentation techniques that overcome deficiencies on the representation. To this end, this thesis introduces three new cell segmentation models, two of which introduce a new cell representation technique as well.

In our experiments, we tested our algorithms on various microscopy images obtained under the fluorescence and phase contrast microscopies and compared them with the previous cell segmentation methods. Our experiments show that the proposed algorithms are more effective in segmenting cells and more robust to the aforementioned challenges.

Keywords: Microscopy imaging, fluorescence microscopy, phase contrast microscopy, cell segmentation, cell detection, nucleus segmentation, nucleus detection.

ÖZET

MİKROSKOPİ GÖRÜNTÜLERİNDE OTOMATİK HÜCRE ANALİZİ

Can Fahrettin Koyuncu
Bilgisayar Mühendisliği, Doktora
Tez Danışmanı: Çiğdem Gündüz Demir
Eylül 2018

İnsan müdahelesi gerektirmeyen ve çok fazla miktarda mikroskop görüntüleri elde edebilen yüksek çıktılı mikroskopi sistemleri son zamanlarda popülerlik kazanmaktadır. Öte yandan, bu kadar fazla miktarda görüntünün geleneksel yöntemlerle analiz edilmesi aylarca sürebileceği için, bu geleneksel yöntemlerin uygulanması pratikte neredeyse imkansızdır. Ayrıca, analiz tamamen, analiz eden kişinin yorumuna bağlı olduğu için, sonuçlar önemli miktarda değişkenlik gösterebilir. Buna çözüm olarak objektif ve hızlı olan otomatik karar destek sistemlerinin geliştirilmesine olan ilgi günden güne artmaktadır. Bu sistemler, analizlerini hücre seviyesinde gerçekleştirdiği için hücre bölütlemesi modellerine gereksinim duymaktadır. Hücre bölütlemesinin başarısı da bütün sistemin başarısını etkilemektedir.

Başarılı bir hücre bölütlemesi modeli elde edebilmek için üstesinden gelinmesi gereken zorluklar bulunmaktadır. Bu zorluklardan bir tanesi hücrelerin tabakta çok katmanlı çoğalması sonucu görüntü üzerinde hücrelerin kümelenmiş görülmesidir. Bu hücreler birbirinden ayrılması gerektiği için bunların bölütlenmesi ek bir efor gerektirmektedir. Diğer bir zorluk, hücre içinde bulunan piksellerin yoğunluk değerlerinin görüntü üzerinde homojen dağılmaması ve birbirine dokunan hücrelerde, hücre sınırlarındaki piksel yoğunluk değerlerinin yeterli farklılıkta olmaması gibi kusurlardır. Bir diğer zorluk ise hücrelerin morfolojik karakterlerinin farklılık göstermesidir. Hücre hattı tipine bağlı olarak hücreler farklı görünümlere sahip olabilir. Farklı hücre görünümlerinde çalışabilecek ve hücre içindeki kusurları çözebilecek kapsamlı bir hücre bölütlemesi modeli geliştirmek çözüme açık ve zor bir problemdir.

Bu zorlukların üstesinden gelebilmek için hücre bölütlemesini iki kısımda ele almaktayız. (1) Mikroskopi görüntüleri için, hücre bölütleştirme problemini basitleştirmede bize yardımcı olabilecek yeni gösterimler bulmaya odaklanmaktayız. Bu sayede, hücre içindeki kusurların ve hücre görünümündeki

farklılıkların üstesinden gelebileceğimizi ve hücre konumlarını daha iyi vurgulayabileceğimizi düşünmekteyiz. (2) Hücreler için kusursuz bir gösterim bulmanın pratikte neredeyse imkansız olması motivasyonuyla, daha gelişmiş bir gösterim metodu geliştirme üzerine odaklanmaktayız. Dolayısıyla, gösterim kusurlarının üstesinden gelebilecek, daha gelişmiş hücre bölütleme tekniklerinin geliştirilmesi üzerinde çalışmaktayız. Bu bağlamda, bu tez, üç yeni hücre bölütlemesi modeli sunmaktadır. Bunlardan ikisi, aynı zamanda, yeni gösterim modeli de ortaya koymaktadır.

Deneylerimizde, sunduğumuz algoritmaları, floresan ve faz kontrast mikroskoplarında elde edilmiş görüntülerde test ettik ve önceki hücre bölütlemesi metotlarıyla karşılaştırdık. Deneylerimiz, önerdiğimiz algoritmaların hücre bölütlemesinde daha etkin ve bahsi geçen zorluklara karşı daha gürbüz olduğunu göstermektedir.

Anahtar sözcükler: Mikroskopi Görüntüleme, Floresan Mikroskopi, Faz Kontrast Mikroskopi, Hücre Bölütlemesi, Hücre Tespiti, Çekirdek Bölütlemesi, Çekirdek Tespiti.

Acknowledgement

This thesis is the end of my journey in obtaining my Ph.D. degree. At the end of this journey, I would like to thank all people, who made this thesis possible.

First and foremost I would like to thank my advisor Assoc. Prof. Dr. Çiğdem Gündüz-Demir for every single thing she has done for me, including her guidance throughout my entire academic career and her time for our endless scientific discussion, which are just few of many things I should be thanking her for. I consider myself very lucky to have had such an encounter at this stage of my career. I recognize that her limitless patience, field knowledge, and scientific guidance made this Ph.D. work possible.

In regard to TTC meetings, I would like to thank my tracking committee members; Prof. Dr. Rengül Atalay and Asst. Prof. Dr. Can Alkan for their precious time. I would also like to thank to Asst. Prof. Ramazan Gökberk Cinbiş and Assoc. Prof. Dr. Selim Aksoy for reviewing and commenting on this thesis.

I would be unfair on my part if I let my friends go unaccounted for. I personally thank Burak, Caner, Kıvanç, Naveed, Sitar, Soner, and Volkan. They are my sources of laughter, joy, and support. Their encouragement has been indispensable. I would also like to thank Gözde Nur Güneşli for her indisputable contribution to the deep learning based cell detection study of this thesis.

I would like to thank to the Scientific and Technological Research Council of Turkey (TÜBİTAK) for providing financial assistance during my study, through the BİDEB program.

I owe my deepest gratitude to my parents, Necla Koyuncu and Ahmet Dursun Koyuncu, and my sister, İrem Özçifçi and her husband, Gökçen Özçifçi. Their love, support, sacrifice, and encouragement provided my inspiration and were my driving force. I owe them everything and wish I could show them just how much I love and appreciate them. My little niece, Ada Özçifçi has brought happiness and joy to our family. I wish to inspire her with this academic gift to help her in choosing a career path she will set upon later in her life.

Contents

List of Figures

List of Tables

In reference to John Wiley & Sons, Inc. copyrighted materials which are used with permission in this thesis, Wiley Company does not endorse any of Bilkent University's products or services. Internal or personal use of these materials is permitted. If interested in reprinting/republishing Wiley copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to Copyright Clearance Center and request permission using RightLinks®.

Copyright Information

Chapter 1

Introduction

Cellular biology research is the examination of the structure and function of the cell, which is the basic unit of life, to understand all aspects of disease. Conventionally, the examination starts with the staining of different cellular components for the purpose of visualization under microscope. Employing different types of microscopes and staining techniques yields a wide range of imaging modalities. For instance, fluorescence microscopy is used for observing minuscule organic molecules of a cell by utilizing synthetic and organic fluorescent dyes. Apart from fluorescent dyes, different types of stains can be used with respect to the type of microscopy. To illustrate, Hoechst staining is used to dye genomic DNA under fluorescence microscopy. Another staining method called *Hematoxylin&Eosin* dyes nuclei as blue and cytoplasms as reddish pink. Yet-another staining technique called *Papanicolaou*, which is commonly known as the *pap smear*, is used for examining cells obtained from body fluids. However, staining as such has damaging effect on cells. To refrain from these negative effects, bright-field and phase contrast microscopies have been developed for live cell imaging without the need for any staining. As the technology advances, new staining methodologies, as well as novel microscopy techniques, have been implemented as new imaging modalities, each of which has its own merits and caveats. These advanced imaging techniques are capable of producing high-throughput data for researchers, pathologists, and biologists to study human physiology in disease and in health.

With the advent of high-throughput microscopy systems, boundless microscopy images can be acquired in no time without necessitating human intervention. On the other hand, traditional analysis of these images can take up to weeks, or even months. Besides, the subjectivity problem can occur during the analysis. Considering these drawbacks, such individual analyses are thought of as a bottleneck for the entire pipeline (Figure 1.1[1]), necessitating automated or semi-automated decision support systems. These systems typically need to locate and quantify the cell, which usually makes up the first part of such systems. This thesis focuses on developing novel, effective methods for this first part of automated decision support systems in which cell segmentation/detection[2] is performed.

1.1 Motivation

Quantitative analysis of microscopy cell images plays a pivotal role in the diagnosis and the clinical assessment of a disease. However, traditional cell image analysis has several drawbacks: (1) It is tedious and lengthy as high-throughput microscopy techniques evolve to produce a massive number of images simultaneously, each of which should be analyzed diligently. Individual analysis of microscopy images is usually impractical because it is labor intensive and unlikely to reproduce. (2) It is subjective since the perception of the analyst plays a substantial role in defining the result, which yields inter-observer variability problem. Additionally, It is possible to make opposing decisions on the same sample when the analyst reviews it at different times, which yields intra-observer variability problem. Considering these drawbacks, microscopy cell image analysis necessitates objective and rapid computer-aided systems. Since the systems perform these quantitative analyses at cellular level, there is currently a need for accurate,

[1]The images illustrating each step of the entire pipeline in the figure are obtained from the following links respectively:https://www.labcompare.com/241-TCSPC-Fluorescence-Lifetime-Imaging-Microscopy-Flim-Microscopy/9377088-Easy-Ratio-Pro/, https://www.mpbio.com/featured.php?fid=8, http://www.multivu.com/players/English/7270051-college-of-american-pathologists-cap-14-what-s-next-in-cancer-testing/

[2]Since cell detection and cell segmentation are similar tasks, this thesis uses them interchangeably. However, to be more specific, in the studies explained in Chapters 3 and 4, our aim is the segmentation of cells, whereas in Chapter 5, the aim is the identification of their locations.

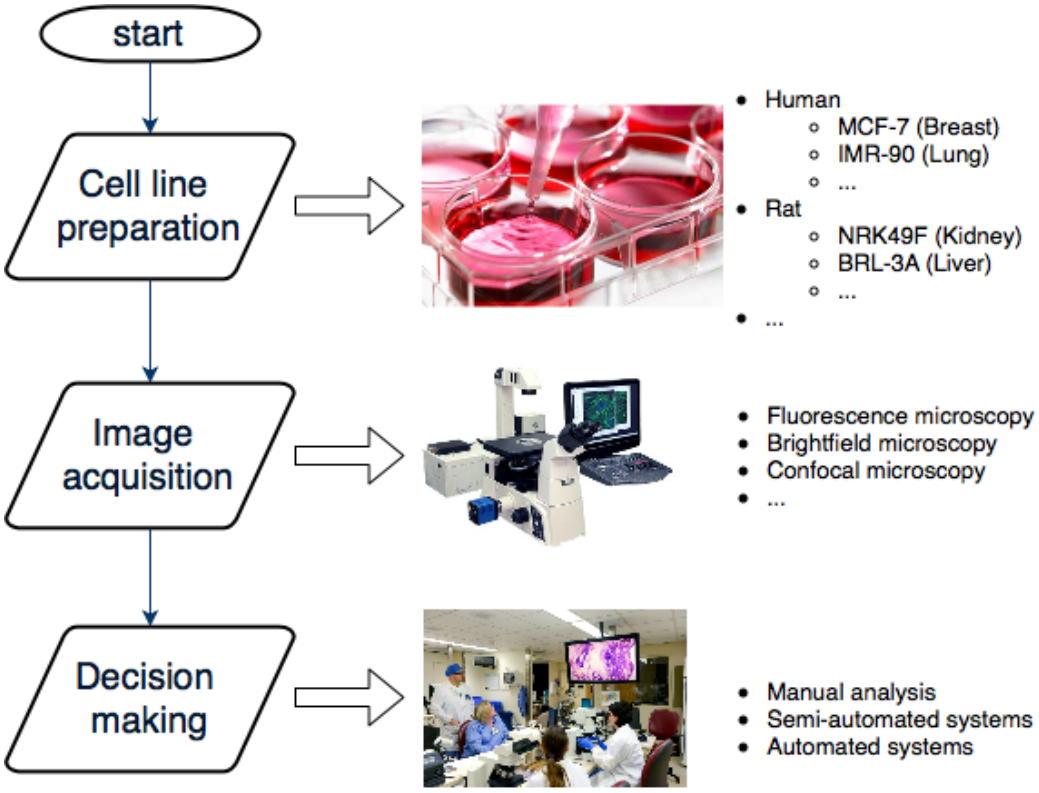


Figure 1.1: General workflow of studying the underlying biologic processes in a disease.

robust, and reproducible cell/nucleus[3] segmentation methods in this field.

In the clinical assessment of a disease, morphological properties of cells have been studied with all details because they store vital information about the disease. For example, in a cancerous tissue, the overall size and shape of cells are often abnormal and vary greatly. Another point is the arrangement of cells which reflects the function of a tissue. In a normal tissue, cells are coordinated with their neighboring cells within the tissue. However, in a cancerous tissue, cells are not coordinated with their neighboring cells and invade the tissue. Such observations are vital for researchers to diagnose the cancer and identify its type. To conduct such analyses, cells must be identified or segmented beforehand.

Cell segmentation can be categorized in two different contexts; segmentation

[3]Cell and nucleus will be used interchangeably.

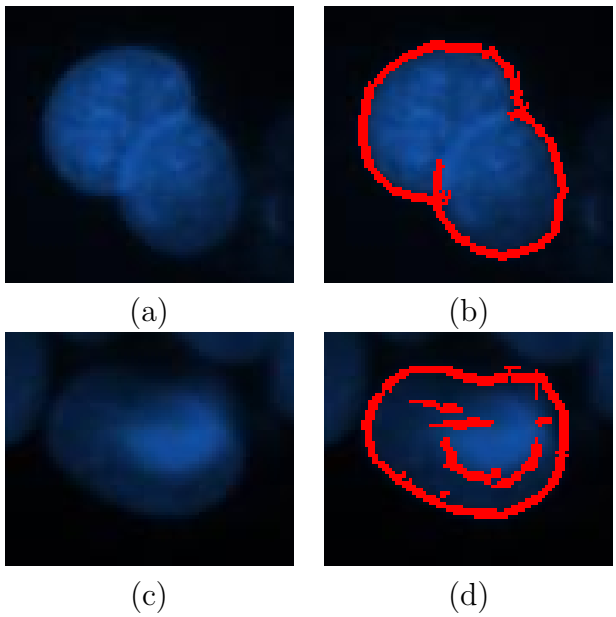
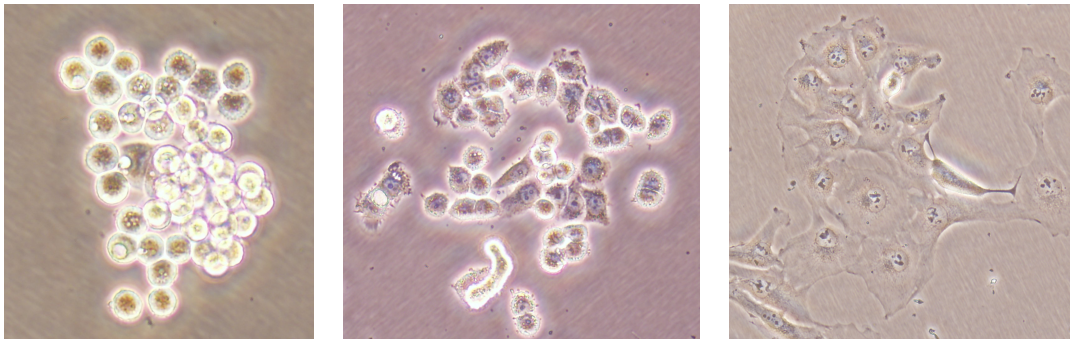


Figure 1.2: Illustrations of imperfections occurred in cells. (a) An example sub-image for insufficient pixel intensity differences at the border of overlapping cells which may cause to miss some true boundary pixels. (b) Identified high-gradient pixels are shown in red. (c) Example sub-image for inhomogeneities of pixel intensities in a cell which may cause to define spurious boundaries. (d) Identified high-gradient pixels are shown in red.

of monolayer isolated cells and segmentation of clumped cells. Segmentation of isolated cells is straightforward. The research segmenting these cells firstly differentiate cell pixels from the background using thresholding or clustering algorithms and then consider the connected components of the cell pixels as segmented cells. On the other hand, segmentation of clumped cells remains an open and challenging problem. Applying thresholding or clustering would not be sufficient for this problem since cells as such had to be split into individual cells in the image. Splitting a cell clump is straightforward when the gradients of boundary pixels in-between the cells are higher than the others. However, imperfections may exist: Inhomogeneities of pixel intensities in a cell may cause to define spurious boundaries resulting in falsely defined cells whereas insufficient pixel intensity differences at the border of overlapping cells may cause to miss some true boundary pixels resulting in missed or under-segmented cells (see Figure 1.2).

In addition to these hurdles, yet-another challenge is heterogeneity in the visual



(a) (b) (c)

Figure 1.3: Example images from human liver and breast cancer cell lines containing cells with different morphological properties: (a) MDA-MB453, (b) MDA-MB-468, and (c) FOCUS cell lines.

characteristics of cells. Visual appearance of cells within a given cell type may vary from one cell to another. In other words, cells may appear in varying colors and irregular shapes within the same cell type. In Figure 1.3, sample images from human liver and breast cancer cell lines are shown. Most of the cells in the MDA-MB-453 cell line have near-circular cells. However, a certain group of these cells contains mostly bright pixels whereas others contain dark pixels inside and bright ones outside (Figure 1.3a). Cells of the MDA-MB-468 cell line are near-circular, as well as non-circular (Figure 1.3b). For the FOCUS cell line, a majority of nuclei are near-circular, containing small low-intensity dots inside. However, there is a second group of cells in the same cell line which has irregular shapes and brighter boundary pixels (Figure 1.3c). The heterogeneities in the morphological properties of cells increase the difficulty level of cell segmentation problem. The difficulty further increases when the cells grow overclumped.

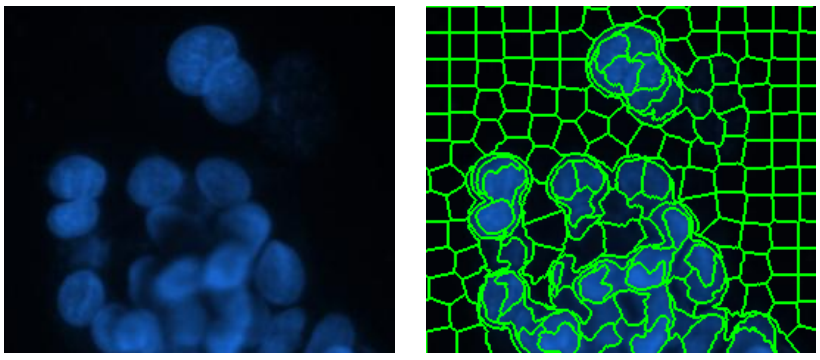
1.2 Contribution

Even though cells are overclumped and contain aforementioned imperfections, their segmentation will still be straightforward when we have a representation illustrating these cells neatly. For instance, it would be ideal if cells were represented on a density map where pixels have brighter intensity values on cell

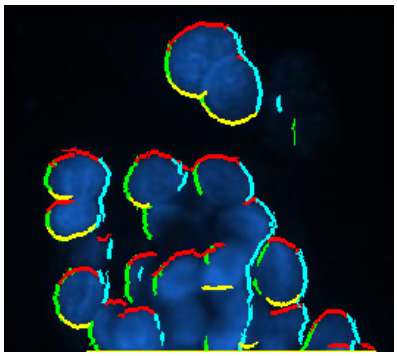
centroids and darker values on the rest. In this case, it will be sufficient to detect regional maxima as location of the cells or to apply a marker-controlled watershed algorithm to segment them. Overall, when we have a more powerful representation, we can put less effort in segmentation of the cells to obtain more accurate results. On the other hand, it is not an easy task to obtain a perfect representation, which renders straightforward approaches such as regional maxima detection and marker-controlled watershed unsuccessful. As a remedy for an imperfect representation, we can utilize more sophisticated segmentation algorithms which compensate for such deficiencies.

Thus, in order to perfect segmentation tools, we should carefully discuss the problem in two different aspects: (1) developing a model which converts the representation of an image into one which is more meaningful and easier to analyze and (2) developing a powerful cell segmentation technique which can compensate for failures in accurate representations. In this thesis, we introduce new powerful cell representations and generate new techniques for accurate cell detection and segmentation on microscopy images. In this context, this thesis has three main contributions.

For our first contribution [1], we introduce a new object-oriented cell segmentation method. The main contributions of this object oriented method are (1) the introduction of reconstructing a fluorescence microscopy image in terms of subregions and edge-objects of different types and (2) the implementation of a new merging algorithm that effectively uses this high-level reconstruction for segmenting cells. As this study works on a high-level representation and employs object-level gradients (i.e. *edge-objects*), the proposed segmentation method is expected to be less vulnerable to the aforementioned pixel-level imperfections compared to the existing studies that directly work on the pixel intensities/gradients. Toward this end, we propose to decompose an image into smaller homogeneous subregions using one of the super-pixel generation algorithms [2] (see Figure 1.4b), define edge-objects at four different orientations to encode the gradient information at the object-level (see Figure 1.4c), and devise a merging algorithm, in which the edge-objects vote for subregion pairs along their orientations and the pairs are iteratively merged if they get sufficient votes from multiple orientations. The



(a) (b)

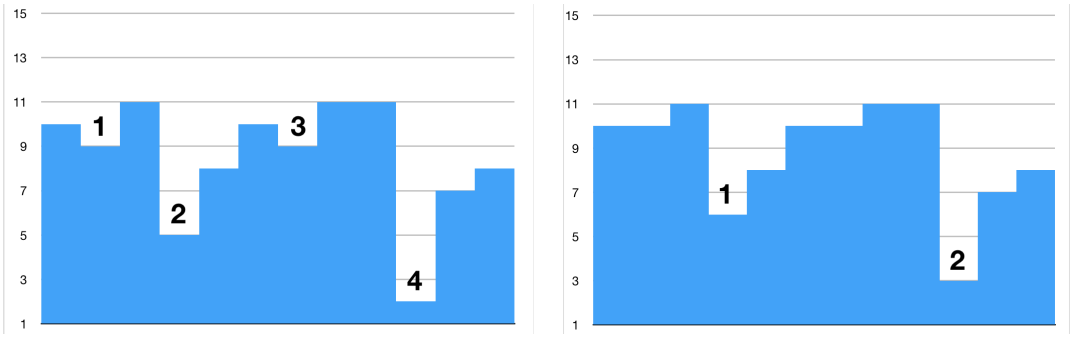


(c)

Figure 1.4: (a) An example fluorescence microscopy image, (b) its subregions obtained using the SLIC algorithm, and (c) its *edge-objects* of different types. Here left, right, top, and bottom *edge-objects* are shown with green, cyan, red, and yellow, respectively.

details of this method are given in Chapter 3.

As for our second contribution [3], we devise a new cell detection method which detects cell locations successfully even though an image contains aforementioned imperfections. The method iteratively identifies nucleus locations making use of h-minima transform, considering a set of different h values. The success of the work comes from the use of different h values, which compensates for the imperfections on a given map. H-minima transform removes regional minima whose depth is smaller than the predefined h value (see Figure 1.5). Our proposed algorithm relies on using multiple h values to identify the markers of a connected component, which corresponds to a nucleus clump on an image. The motivation behind this is the fact that there exists no best h value that can be used to identify all markers of the same connected component due to the possible variations in the



(a) (b)

Figure 1.5: Illustration of h-minima transform in 1-D. (a) Example data with the numbers showing the location of regional minima. (b) After h-minima transform with an h value of 1. As seen in the example, the minima-1 and minima-3 in the left image are eliminated since their depths are smaller than or equal to the selected h value.

nuclei's sizes, shapes, and intensities within the same nucleus clump. We believe that the proposed work can better identify nuclei even when the map contains imperfections. Toward this end, we extend the study on live cell image data. Experimental results show that the iterative approach improves the detection performance slightly in comparison with the conventional h-minima transform [22]. The proposed approach is discussed comprehensively in Chapter 4.

As for our third contribution, we propose to obtain a better representation of live cell images directly on image data using deep learning techniques and to use this representation to successfully detect cell locations. Toward this end, we propose a new model that learns high-level features directly from image data. In particular, this model proposes to pose cell detection as a regression problem, in which the normalized distance from each pixel to the closest background is learned by training a deep convolutional neural network (CNN) on small patches. Then, for a given image, cell detection is achieved by generating a normalized distance map of the image pixels with the trained CNN and by finding regional maxima of the generated map. Since the proposed model automatically learns features from the image data, it is not necessary to redesign any of its steps for new cell types, even when they show different visual characteristics. For a new cell type, our model uses the same CNN architecture and the same steps for cell

detection; it only needs a couple of annotated training images of the new type, on which the network is to be fine-tuned. This work is extended by integrating a cascaded multi-task learning U-Net, instead of CNN. Employing a more sophisticated network in the learning step boosts the efficiency and effectiveness of the whole pipeline. The model is further improved by employing iterative h-minima transform, which is our second contribution in this thesis, instead of conventional h-minima transform. The details of this deep learning model and its extension are given in Chapter 5.

The remainder of this thesis is organized as follows. The images studied in the thesis and the previous studies related with cell segmentation on microscopy images are discussed in Chapter 2. Subsequently, the methodology and experimental results of a new cell representation and segmentation method, named object-oriented segmentation method, are given in detailed in Chapter 3. Then, in Chapter 4 a cell detection method, named iterative h-minima based detection, is deeply discussed together with its experimental results. As for our last contribution, the deep learning based new cell representation method and relevant experimental results are given in Chapter 5. Lastly, a summary of our proposed models is given together with their possible future directions in Chapter 6.

Chapter 2

Background

In this thesis, we propose fully automated cell segmentation tools focusing on microscopy images. Before these proposed tools are elaborated in the next chapters, the details of the microscopy images which we made use of are given in Section2.1. Then, in Section 2.2, we provide with a survey of related studies within the context of cell segmentation of microscopy images based on two main categories: segmentation of isolated cells and segmentation of overclumped cells.

2.1 Microscopy Images

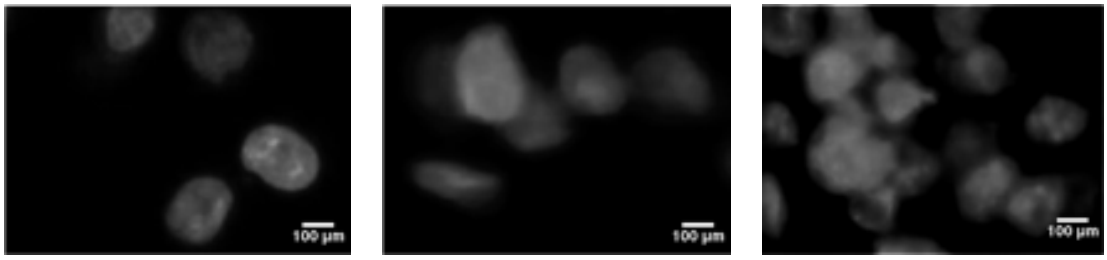
In this thesis, we focus on fluorescence microscopy and phase contrast microscopy images, all of which were obtained from human carcinoma cell lines. The images belong to two major types of cancer cell lines, namely breast cancer and liver cancer (Table 2.1), which have pervasive effect on human life. Liver cancer is one of the leading causes of cancer-related death worldwide, and breast cancer is one of the most common cancers diagnosed in women. Thus, research on especially these two cancer types requires more attention because of their devastating impact on the society worldwide.

Table 2.1: Name of microscopies and the cancer cell lines together with their types used in this thesis.

Microscopy type	**Cancer type**	**Cell line**
Fluorescence	Liver	Huh7
Fluorescence	Liver	HepG2
Phase contrast	Liver	FOCUS
Phase contrast	Liver	Huh7
Phase contrast	Liver	SkHep1
Phase contrast	Breast	MDA-MB-453
Phase contrast	Breast	MDA-MB-468
Phase contrast	Breast	CAMA-1

There are eight different data sets used within the scope of this thesis (Table 2.1). In the models explained in Chapter 3 and Chapter 4, fluorescence microscopy images were analyzed. The data set contains isolated and overclumped cells of the Huh7 and HepG2 cell lines. Example images showing both isolated and overclumped cells are given in Figure 2.1. As seen in this figure, all cells are round-like shape, but isolated cells were grown monolayer on the plate and appeared isolated on the image (Figure 2.1a). Overclumped cells were grown in multilayer on the plate. These overclumped cells could be less-confluent (Figure 2.1a), where some overlaps appear along the boundaries of their nuclei, or more-confluent (Figure 2.1b), where the nuclei appear as clusters on the image. Please note that in fluorescence microscopy images cytoplasm region of cells do not appear because of the staining technique. Blue regions are actually correspond to nucleus part of cells .

In the study explained in Chapter 5, phase contrast microscopy images were used. There are six different types of cell lines, examples of which are given in Figure 2.2. As seen in the examples, visual characteristics may change from one cell type to another. Additionally, in the first three cell lines, cells have irregular shapes and heterogeneous intensities. Small low-intensity dots represent cell nuclei but there also exist similar-looking small dark subregions, especially towards



(a) (b) (c)

Figure 2.1: Example fluorescence images from human liver cancer cell lines used in our experiments: (a) Monolayer cells whose nuclei appear as isolated (Huh7), (b) less-confluent cells for which some overlaps appear along the boundaries of their nuclei (Huh7), and (c) more-confluent cells whose nuclei appear as clusters (HepG2).

cellular boundaries. In the last three cell lines, cell shapes are less irregular. However, this time they appear differently. For instance, in Figure 2.2d, there are mostly near-circular cells, which sometimes contain more than occasionally bright pixels but sometimes contain dark pixels inside and bright ones outside. On the other hand, in Figures 2.2e and 2.2f, there are near-circular as well as non-circular cells. For such images, it would not be easy to use a single model to detect cells with different morphologies.

2.2 Related Work on Cell Segmentation for Microscopy Images

Segmentation of cells in microscopy images typically starts with differentiating nuclear pixels from background to obtain a binary mask. For that, it is usually adequate to apply simpler techniques such as thresholding [4, 5, 6, 7, 8, 9] and clustering [10, 11, 12, 13] on pixel intensities when there is a huge intensity difference between foreground and background pixels in images. However, images usually exhibit poor contrast, degrading in differentiating the foreground pixels from the background. There are several approaches that apply preprocessing

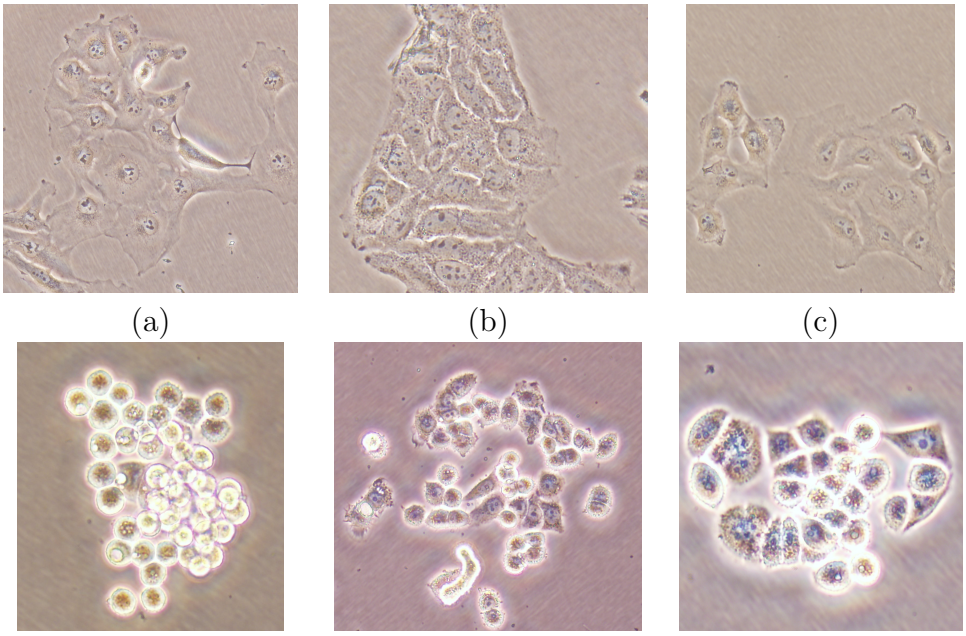


Figure 2.2: Example phase contrast images from human liver and breast cancer cell lines used in our experiments: (a) FOCUS, (b) Huh7, (c) SkHep1, (d) MDA-MB-453, (e) MDA-MB-468, and (f) CAMA-1 cell lines.

techniques, including color normalization and image denoising to handle this issue [14, 15, 16]. Afterwards, segmentation continues with identifying cells on the binary mask.

Segmentation of cells is quite straightforward when the cells appear isolated on an image; each connected component on the binary mask corresponds to a cell. On the other hand, it becomes challenging to segment cells in cell clumps, in which the cells appear touching or overlapping on the image. In this case, a connected component should be split into multiple cells. There are many cell segmentation studies targeting on cell clumps. These studies can be categorized according to the main concept utilized for the segmentation purpose.

Shape-based methods split one component into multiple cells using the fact that a typical cell is nearly circular and convex. For that, a group of these algorithms have used Gaussian mixtures [17] and physical deformable models [18] to decompose clustered cell nuclei based on their roundness. The marker-controlled

watershed is another technique that the previous studies have commonly used to segment clustered cells. They define a set of markers on an image and obtain cell nucleus regions growing them only from these predefined markers. A group of the previous studies apply morphological operations (e.g., erosion) on the binary mask [19, 20] to obtain markers for the watershed algorithm. Yet another group of marker controlled watershed algorithms take regional minima found on distance transform maps as the markers [21, 22, 23] and/or grow the identified markers on the distance transforms [24, 25, 26, 27]. However, defining regional minima as markers is very sensitive to noise, and hence, may lead to defining spurious markers. To alleviate this problem, these algorithms typically apply the h-minima transform, which suppresses all minima under a value of h, before finding the regional minima [28, 22, 23]. Another group of shape based methods employ concavity detection algorithms to find concave points on the mask and split the cells from these points [29, 30, 31, 32]. It has also been proposed to split the mask by identifying circular shapes by the Hough transform [33] and ellipse fitting techniques [34]. The shape-based methods usually yield promising results when the degree of overlapping is relatively low so that there is not so much deviation in cell appearance from its assumed circular shape. Additionally, when the overlapping degree impedes finding a sufficient amount of background pixels adjacent to the boundary of a cell, the distance transform may give misleading results and concavity detection may not work.

Gradient-based methods identify individual cells in a cluster relying on the fact that cell contours have high contrast differences. The voting based techniques define kernels to obtain the gradient information and get image pixels voted along the directions specified by these kernels, imposing cells on having radially symmetrical shape. They then identify regions with larger votes as nucleus centers [35, 36, 37, 38, 39]. It is also possible to use pixel gradients to detect the markers of a marker-controlled watershed algorithm [4, 11, 40, 41]. Level set algorithms commonly employ the pixel gradients to refine the nucleus boundaries found by the shape-based methods. These algorithms define their energy functions on the gradients and converge the final boundaries by minimizing these energy functions [42, 43]. Yet another group of gradient based approaches include

superpixel based algorithms. A superpixel is a group of neighboring pixels having similar features. It reduces the complexity of the algorithm since, instead of dealing with a great deal of pixels, just a few superpixels are taken into account within the scope of this approach. In [44, 45, 46], superpixels are classified as one of the three major classes for cell detection: nucleus, cytoplasm, and background. There is another group of superpixel based algorithms which over-segment images into superpixels and then merges the ones belonging to the same cells [47, 48]. However, for all these gradient based methods, the use of the gradient information may not always be adequate to correctly split the overlapping nuclei due to the imperfections in pixel values due to the following reasons, which are also explained in Section 1.1. These are (1) inhomogeneities of pixel intensities in a cell which may cause to define spurious boundaries resulting in falsely defined cells and (2) insufficient pixel intensity differences at the border of overlapping cells which may cause to miss some true boundary pixels resulting in missed or under-segmented cells.

Graph-based algorithms utilize graph structure to better represent cells. One popular approach is to segment cells by constructing a tree structure which stores maximally stable extremal regions (MSER) [50, 51, 52, 53, 54, 55]. Based on region properties, the algorithm detects the hot spots which have higher probability of being a cell. The method given in [50] employs eccentricity features of the extremal regions. Those explained in [51, 52, 53, 54, 55] exploit color and shape properties of the regions together. In [51, 52, 54], detection problem is converted to an optimization problem, where the aim is to select non-overlapping regions that are maximizing previously defined score metric. Moreover, the method in [52] shows the generality of the MSER tree structure, having experiments on various independent data sets. Even though it performs well on these data sets, what might limit its efficiency is the use of hand-crafted *low-level* features. Other graph-based approaches define each image as a graph, where nodes correspond to pixels and edges to their neighborhood information [56, 57]. In [56], cell nuclei are segmented using two-step graph-cut methods. In [57], the authors incorporate *blob-like* shape information into a graph-cut algorithm to segment cells. In [46], instead of pixels, superpixels are defined as nodes, which increases efficiency of

the algorithm significantly. One of the previous studies developed by our research group also utilizes graph structure, called attributed relational graph, to segment nuclei in fluorescent microscopy images [58]. First, it defines four different *edge primitives* with respect to orientation information. Then, it constructs an attributed relational graph by defining nodes and edges based on the *edge primitives* and their spatial properties according to each other. It localizes nuclei by searching specific structural patterns on the graph.

Lastly but not least importantly, a group of machine learning-based algorithms have been proposed as a segmentation model. Several of these algorithms use machine learning as a preprocessing step to enhance images [59, 60] and/or a post-processing step to eliminate false positives [59]. Besides, the majority of these studies employ hand-crafted features, which limit their extendability to other domains [48, 61, 62, 63, 64, 65]. More interestingly, there exist a group of studies focusing on deep learning approaches, which have the ability to learn features from input images without requiring any human interference. These deeply learned features might have high level information and can be used, unlike the hand-designed ones, to tackle problems such as detection and segmentation. These studies can be categorized by some aspects such as: whether the problem is formulated as a classification or a regression task and whether the model is trained with a sliding window approach or with an end-to-end technique. The majority of these studies pose cell detection as a pixel-wise classification problem in which a deep classifier is trained to differentiate between cell pixels and the background. Then, for a given image, they may obtain a binary mask by estimating the class labels of the image pixels using a trained classifier and then use this mask as an input to shape-based methods [68, 69]. Alternatively, they may use the class posteriors of the pixels and identify cell locations on this posterior map by either thresholding [66, 70] or clustering [67] but mostly finding regional maxima [71, 72, 73, 74, 75]. Since their focus is cell pixel classification, these studies treat the pixels taken from the annotated cells the same way, regardless of their relative positions within the cell during the training step. On the other hand, the position of a pixel relative to a cell center (or to a cell boundary) may bring

about additional information. There exists some studies that take this information into account by constructing a regression model that learns a continuous distance output for each pixel [76, 77, 78, 79]. All these approaches compute the output map with regards to the distance between the pixel and its closest cell center that is annotated. In [77, 78, 80], the output is the Euclidean distance from the pixel to its closest cell center annotated. This definition, however, imposes a circular and one-sized shape on cells, which may not be true for all cell types. In both classification and regression, earlier studies have used a patch-based (sliding window) approach [68, 69, 66, 70, 67, 71, 72, 74] to estimate the output (i.e., class labels posteriors in classification and distance maps in regression). However, this technique is not efficient due to the redundant computations required for the overlapping patches extracted around each pixel. To overcome this issue, end-to-end methods, especially Fully Convolutional Networks (FCN) [81, 82] and U-net model [83], have become more popular for cell segmentation purposes [79, 84, 80]. The method proposed in [79] utilizes pre-trained off-the-shelf models (DeepLab model from [85]). Those explained in [84, 80] modify FCN and U-net architectures by adding/dropping some network layers, by changing the number of features, and by adding residual connections/blocks.

Chapter 3

Object Oriented Segmentation of Cell Nuclei in Fluorescence Microscopy Images

This study introduces a new cell segmentation method that relies on using gradient information not at the pixel-level but at the object-level. To this end, it proposes to decompose an image into smaller homogeneous subregions, define *edge-objects* at four different orientations to encode the gradient information at the object-level, and devise an effective algorithm that segments nuclei by merging the smaller subregions using the *edge-objects*. In this merging algorithm, the *edge-objects* vote for subregion pairs along the direction specified by their edge types and the subregion pairs are iteratively merged provided that they get sufficient votes from multiple directions. The main contributions of this object oriented method are the introduction of representing a fluorescence microscopy image in terms of subregions and *edge-objects* of different types and the implementation of a new merging algorithm that effectively uses this high-level representation to segment nuclei. As it works on a high-level representation and employs object-level gradients, the proposed segmentation method is expected to be less vulnerable to the aforementioned pixel-level imperfections compared to the existing studies that directly work on the pixel intensities/gradients [35, 36, 37, 38, 39]. Note

that the study developed by our research group also uses the *edge-objects* for nucleus segmentation [58]. However, this use is completely different than the one proposed in this current study. Our previous study constructs a graph on the *edge-objects* and achieves segmentation by searching predefined patterns on the constructed graph. It does not define any subregions, and thus, obviously, it does not use them in any merging algorithm in conjunction with the *edge-objects*.

In the literature, there exist studies that also partition an image into subregions and then form nuclei by merging them. All these studies extract features from the subregions and select the subregions to be merged by solving an optimization problem on the extracted features [47, 48, 49, 52, 54]. Different than our proposed method, these previous studies do not define any kind of high-level objects encoding the gradient information and they do not employ such high-level objects to merge their subregions. Working on 2661 nuclei, our experiments show that this high-level object-based representation together with the proposed merging algorithm yield better results compared to its pixel-based counterparts.

3.1 Methodology

The proposed object oriented method relies on first dividing an image into over-segmented subregions and then merging them with the help of the edge-objects to segment nuclei. The motivation behind this first-divide-then-merge approach is as follows: In principle, one can locate a single subregion for every nucleus. However, due to non-ideal conditions in real life, this may not actually happen for many images and the located subregions are commonly over- or under- segmented. Inhomogeneities inside a nucleus may cause to define multiple over-segmented subregions corresponding to this nucleus whereas insufficient contrast differences at the boundary of two overlapping nuclei may result in representing the two nuclei with the same under-segmented subregion. Hence, we propose to divide the image into homogeneous subregions that are usually smaller than the average nucleus and merge them afterwards.

The merging process employs the *edge-objects* of four different types that are defined at four different orientations. These *edge-objects* correspond to the left, right, top, and bottom nucleus boundaries according to the orientation they are defined. In the ideal case, for each nucleus, one can define exactly one left *edge-object* for its left boundary, one right *edge-object* for its right boundary, one top *edge-object* for its top boundary, and one bottom *edge-object* for its bottom boundary and these *edge-objects* form a closed curve (this hypothetical case is illustrated in Figure 3.1a). In this case, the merging process would be quite simple; one could easily form a nucleus by merging the subregions surrounded by the edge-objects of this nucleus. On the other hand, there may exist the following deviations from this ideal case:

1. The *edge-objects* belonging to the same nucleus may not cover all of its boundaries, and thus, they may not form a closed curve (Figure 3.1b).
2. More than one nucleus may share the same *edge-object* of the same type (Figure 3.1c).
3. Multiple *edge-objects* of the same type may correspond to a single nucleus (Figure 3.1d).
4. The *edge-object* of at least one type may be missing (Figure 3.1e).
5. There may exist spurious *edge-objects* inside a nucleus (Figure 3.1f).

We draw illustrations for each of these deviations in Figure 3.1b-Figure 3.1f. In order to address these non-ideal conditions, the proposed method devises an iterative merging algorithm, in which two subregions are merged provided that they share an *edge-object* for a sufficient number of the edge types. The pseudo-code of the proposed object oriented algorithm is given in Algorithm 1 and its details are explained in the following subsections. The source codes of its implementation are available at `http://www.cs.bilkent.edu.tr/~gunduz/downloads/ObjectOrientedCellSegm/`.

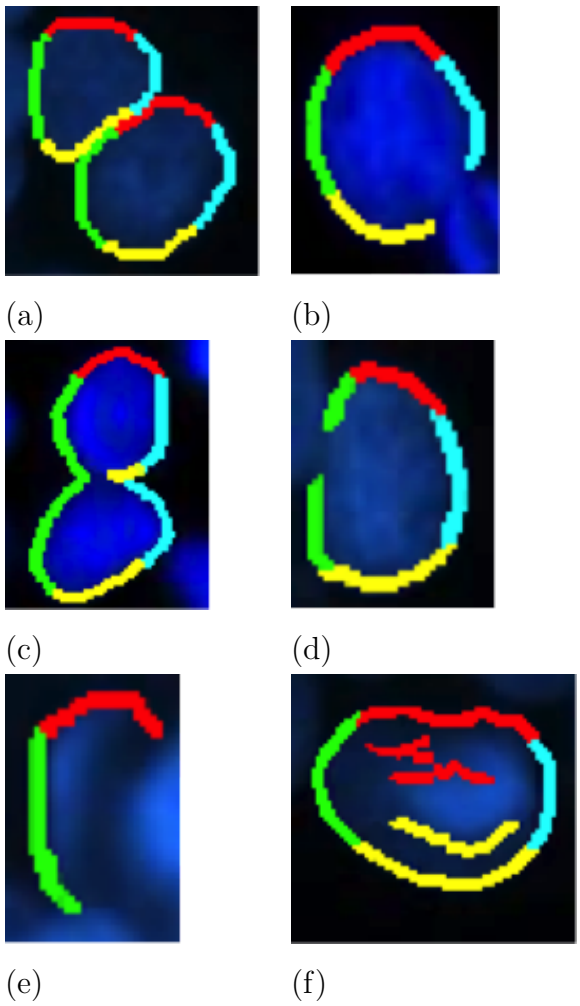


Figure 3.1: Illustrations of (a) the hypothetical case and (b)-(f) the non-ideal conditions for the *edge-object* definition. (b) The *edge-objects* do not form a closed curve. (c) The same left *edge-object* is partially shared by two nuclei. (d) Two different left *edge-objects* are defined for the same nucleus. (e) There is no right and no bottom *edge-object* defined. (f) Spurious top and bottom *edge-objects* are defined. Here left, right, top, and bottom *edge-objects* are shown with green, cyan, red, and yellow, respectively.

Algorithm 1 *NucleusSegmentation*: Overall framework of the proposed object oriented algorithm.

Input: image I, superpixel image P, object size threshold t_{size}, maximum distance d_{max}, voting threshold t_{vote}, area threshold t_{area}
Output: segmented nuclei S

$B \leftarrow OtsuGlobalThresholding(I)$

for all $perc \in \{0.5, 1.0, 1.5, 2.0\}$ **do**
  $O_{perc} \leftarrow EdgeObjectDefinition(I, B, t_{size}, perc)$
end for

$S \leftarrow SubregionPartitioning(P, O_{0.5}, B)$

for all $perc \in \{0.5, 1.0, 1.5, 2.0\}$ **do**
  $S \leftarrow SubregionMerging(S, O_{perc}, d_{max}, t_{vote})$
end for

$S \leftarrow SmallNucleusElimination(S, t_{area})$

3.1.1 Edge-Object Definition

The proposed algorithm defines four different types of the *edge-objects*: left, right, top, and bottom. It derives the *edge-objects* of each type using a gradient map and a binary mask, both of which are obtained on the L channel of an image. For each type, the gradient map is obtained by convolving the L channel with one of the following Sobel operators. In this work, we prefer using the La*b* color space for both *edge-object* definition and subregion partitioning since it was designed to be perceptually uniform with respect to human color vision. Thus, as the first step, an RGB image is converted to its equivalent in the La*b* color space and the remaining steps use this converted image.

The same binary mask B is used for all of the edge types and it is obtained by thresholding the L channel with the value automatically calculated by the Otsu's method [86]. It is usually sufficient for our method to use a rough binary mask as long as this mask does not miss too many true nucleus pixels. The post-processing step will correct false pixels up to a certain degree; it will eliminate

small regions of false nucleus pixels by small area elimination and fill gaps on false background pixels by majority filtering. Thus, the proposed method uses a quite simple thresholding technique for binarization. However, one may consider to obtain such a mask by employing more advanced methods such as supervised techniques. The investigation of using such techniques could be considered as a future work.

$$S_{left} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \; S_{right} = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

$$S_{top} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \; S_{bottom} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

The *edge-object* definition step defines the left *edge-objects* as follows. Let G_{left} be the gradient map obtained by convolving the L channel with the Sobel operator S_{left}. First, G_{left} is compared against a threshold t_{left} and pixels with high enough gradients are identified. These identified pixels are masked with the binary mask B and spurious edges on the image background are eliminated. Then, a binary edge map is defined on the remaining pixels. Finally, the *m-leftmost*[1] pixels of the binary edge map are taken and the connected components of these *m-leftmost* pixels are considered as the left *edge-objects* provided that their heights are larger than the size threshold t_{size}. Here, we take the *m-leftmost* pixels instead of just taking the leftmost pixels since discontinuities may exist in boundaries due to the pixel-based representation of a digital image. In this work, we select $m = 3$ considering the pixel resolution of the images that are used in our experiments. The steps of the left *edge-object* definition are illustrated in Figure 3.2.

[1]The *m-leftmost* pixels of a connected component are defined as follows. For each row of the component, the white pixels which have background pixel in their left neighborhood are identified. Afterwards, these pixels and the consecutive $m - 1$ white pixels which are located in their right adjacency are defined as *m-leftmost* pixels for this component.

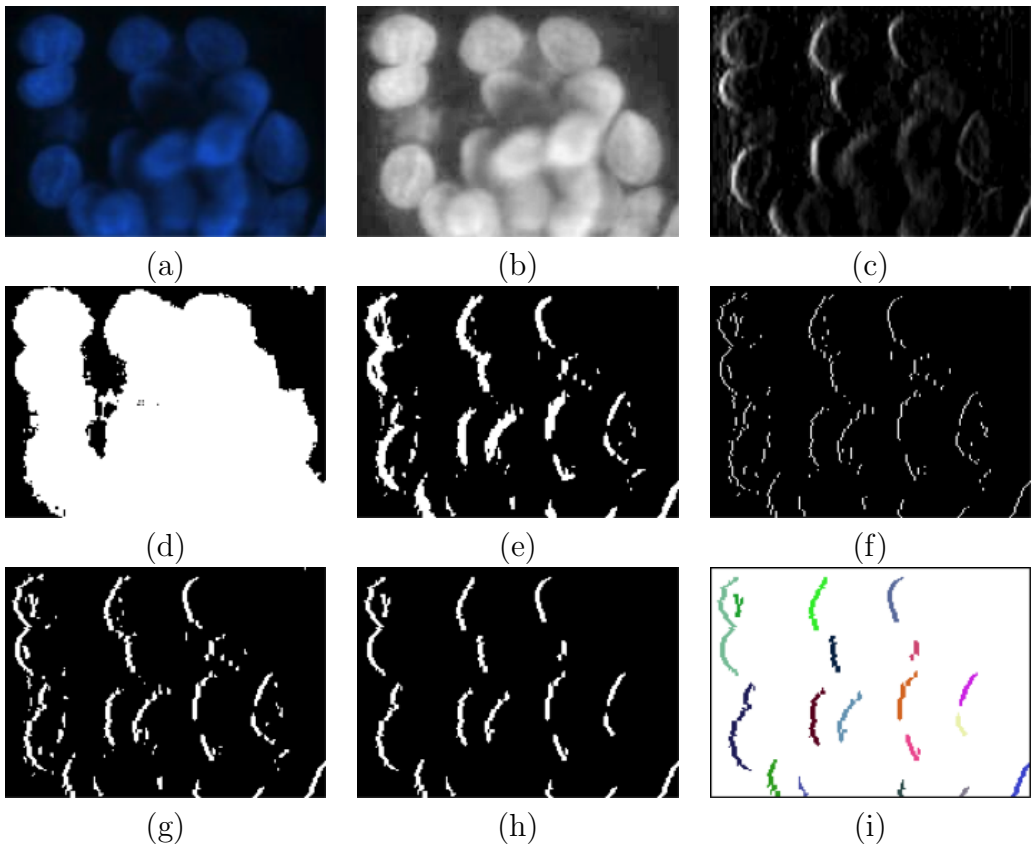


Figure 3.2: Left edge-object definition on an example subimage: (a) original subimage obtained from the HepG2 liver cancer cell line, (b) its L channel, (c) gradient map G_{left} obtained by convolution, (d) binary mask B, (e) binary edge map obtained after thresholding G_{left} and masking the result with B, (f) leftmost pixels of the binary edge map, (g) *m-leftmost* pixels of the same binary edge map, (h) remaining connected components after eliminating the shorter ones, and (i) left *edge-objects* defined for the subimage.

Here it is critical to determine a threshold value t_{left} that can identify *edge-objects* correctly. However, this is not always straightforward and a single threshold may not always work over an entire image especially when the image exhibits variance on the gradient distribution in its different parts. Threshold values smaller than necessary may lead to defining spurious edges whereas too large values may cause to miss some nucleus boundaries. Thus, we propose to use a set of multiple threshold values, for which the subregion merging step is consecutively called one after another (lines 6-7 of Algorithm 1). In particular, a threshold τ is automatically calculated on the gradient map G_{left} by the Otsu's

method, and then the values starting from the half of this threshold to its double are used. In this work, four sets of the left *edge objects* O_{left} are defined using $t_{left} \in \{0.5\tau, 1.0\tau, 1.5\tau, 2.0\tau\}$. We decide to use the threshold percentage *perc* constants of $0.5, 1.0, 1.5$, and 2.0 because of the following reasons: Small values for the minimum *perc* cause to define too much spurious *edge-objects*. On the other hand, large values for the maximum *perc* result in defining almost no useful *edge-objects*. The use of small intervals in between the consecutive *perc* values increases the computation time without adding too much extra information. Thus, considering all these issues, we select $perc \in \{0.5, 1.0, 1.5, 2.0\}$.

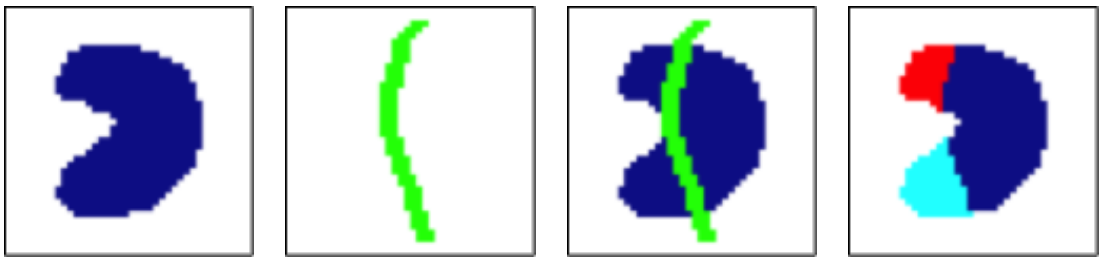
This step defines the objects of the other edge types similarly with the difference that *edge-objects* shorter than t_{size} are eliminated for the left and right types whereas those narrower than t_{size} are eliminated for the top and bottom types. At the end of this step, we obtain four sets of the *edge-objects* $O_{perc} = \{O_{left}, O_{right}, O_{top}, O_{bottom}\}$, each of which is calculated using a different threshold percentage constant $perc \in \{0.5, 1.0, 1.5, 2.0\}$ (lines 2-4 of Algorithm 1).

3.1.2 Subregion Partitioning

The proposed algorithm first partitions an image into homogeneous subregions, which are usually smaller than a typical nucleus, and then merges them with the help of the *edge-objects*. The first step of this partitioning runs the SLIC (Simple Linear Iterative Clustering) superpixel algorithm [2] on the image. The SLIC algorithm clusters image pixels according to their L, a, and b values in the La*b* color space together with their x and y coordinates, and defines a superpixel for each of these clusters. After obtaining superpixels by the SLIC algorithm, the second step of this subregion partitioning takes one of the following three actions for each superpixel p.

1. If p is entirely outside the binary mask B, which is also used in the *edge-object* definition step, it discards this superpixel.

2. Otherwise, if p entirely or partially overlaps with the mask, the overlapping



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

Figure 3.3: Illustration of splitting a superpixel into multiple subregions: (a) a superpixel before splitting, (b) a left *edge-object* that will split the superpixel, (c) the left object superimposed on the superpixel, and (d) three subregions obtained after splitting.

part of p is taken. Then, if any *edge-object* cuts this overlapping part into multiple components, the second step further splits p using this *edge-object* and defines multiple subregions corresponding to the superpixel, as illustrated in Figure 3.3. It repeats this split operation for all such *edge-objects*. Note that this second step considers the *edge-objects* in the set of $O_{0.5}$, which is defined using the lowest Otsu threshold, for further partitioning of the superpixels (line 5 of Algorithm 1).

3. Otherwise, it considers p as one subregion.

At the end of this step, an image is represented with the subregions and the *edge-objects* of four different types. The next step will merge the subregions using the *edge-objects* to form nuclei and the last step will postprocess these nuclei to obtain the final segmentation (Figure 3.4).

3.1.3 Subregion Merging

The merging algorithm involves an iterative procedure, each of whose iterations starts with assigning the present subregions to the *edge-objects* within a distance d. Afterwards, based on these assignments, pairs of the adjacent subregions that share a sufficient number of the *edge-objects* are determined and their merging scores are calculated. Starting from the best one, such pairs are merged with

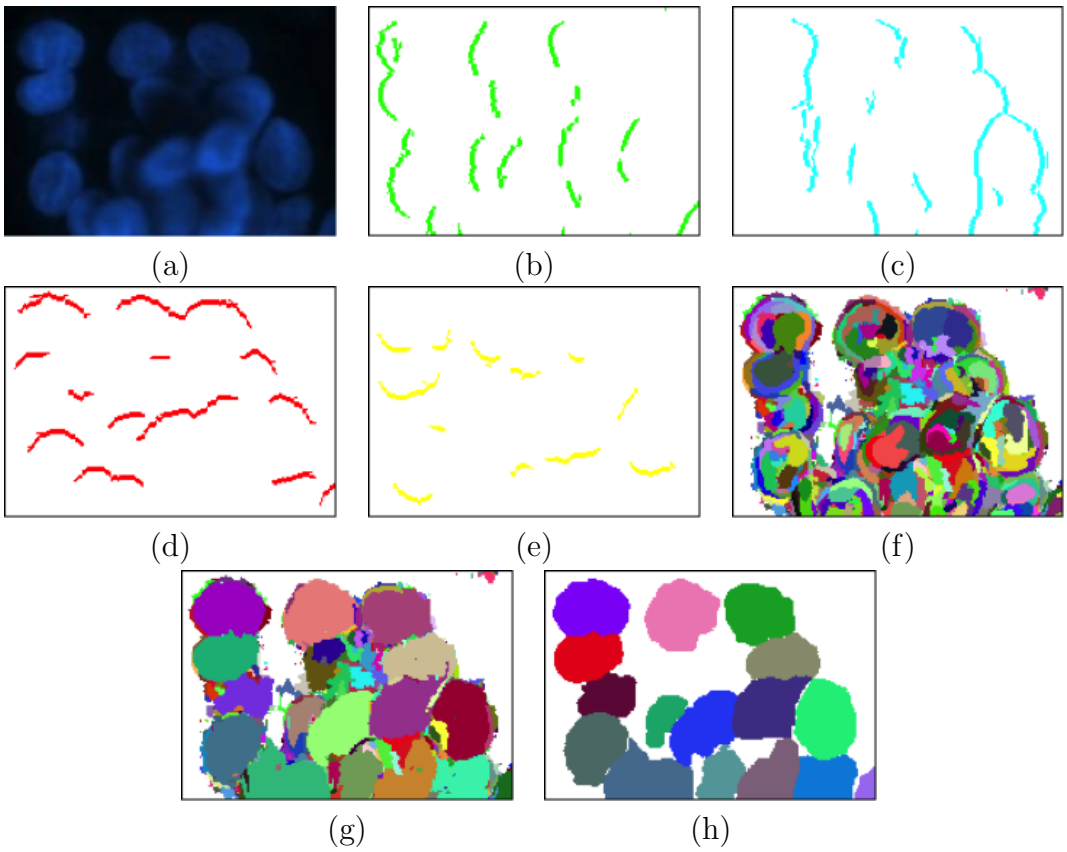


Figure 3.4: Illustration of the *edge-objects* and the subregions before and after merging: (a) original subimage obtained from the HepG2 liver cancer cell line, (b) left *edge-objects*, (c) right *edge-objects*, (d) top *edge-objects*, (e) bottom *edge-objects*, (f) subregions at the end of the subregion partitioning step, (g) nuclei obtained by merging the subregions, and (h) final segmentation after postprocessing.

respect to their scores and the pairs are updated also considering the newly emerged subregions. Each iteration continues until there remains no subregion pair to be merged. The next iteration increments the value of d by one and repeats the same steps. This procedure continues its iterations from $d = 1$ to d_{max}. The pseudo-code of this procedure is given in Algorithm 2; the details of its steps are explained below.

In the first step of each iteration, the subregions are assigned to the *edge-objects* (lines 2-7 of Algorithm 2). For a subregion, this assignment is done separately for each edge type. To assign a subregion s_i to a left *edge-object*, the vote $v(s_i, o_j)$

Algorithm 2 *SubregionMerging*: Procedure that iteratively merges subregion pairs using the *edge-objects*.

Input: subregions S, edge-objects O, maximum distance d_{max}, voting threshold t_{vote}

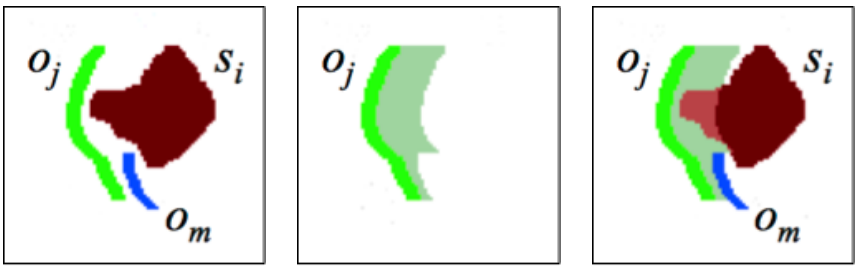
Output: merged subregions S

for $d \leftarrow 1$ **to** d_{max} **do**
  for all subregions s_i of S **do**
    $left(s_i) \leftarrow LeftAssignment(O, s_i, d)$
    $right(s_i) \leftarrow RightAssignment(O, s_i, d)$
    $top(s_i) \leftarrow TopAssignment(O, s_i, d)$
    $bottom(s_i) \leftarrow BottomAssignment(O, s_i, d)$
  end for

  $\phi = \emptyset$
  for all adjacent subregions s_i and s_k of S **do**
    if $\big[left(s_i) = left(s_k)$ **and** $\sigma_{left}(s_{ik}) \geq t_{vote}$ **or**
    $right(s_i) = right(s_k)$ **and** $\sigma_{right}(s_{ik}) \geq t_{vote}\big]$
    and $\big[top(s_i) = top(s_k)$ **and** $\sigma_{top}(s_{ik}) \geq t_{vote}$ **or**
    $bottom(s_i) = bottom(s_k)$ **and** $\sigma_{bottom}(s_{ik}) \geq t_{vote}\big]$ **then**
      $\sigma(s_{ik}) \leftarrow \sigma_{left}(s_{ik}) + \sigma_{right}(s_{ik}) + \sigma_{top}(s_{ik}) + \sigma_{bottom}(s_{ik})$
      $\phi = \phi \cup \{\langle s_{ik}, \sigma(s_{ik})\rangle\}$
    end if
  end for

  for all subregion pairs of ϕ **do**
    $s_{ik} \leftarrow$ select the pair with the highest $\sigma(s_{ik})$
    merge the subregions s_i and s_k
    update the set ϕ
  end for

end for



(a) (b) (c)

Figure 3.5: Illustration of calculating the vote $v(s_i, o_j)$ that the left *edge-object* o_j gives the subregion s_i. (a) The subregion s_i and the two *edge-objects* o_j and o_m are shown in maroon, green, and blue, respectively. (b) The area scanned for the rows of o_j, which is shown with light green color. For the upper rows of o_j, the scan continues until it reaches the distance d. For its lower rows, the scan stops earlier since it hits the blue object o_m. (c) The part of the subregion s_i that overlaps the scanning area, which is shown with light maroon color. The vote $v(s_i, o_j)$ is incremented by one for each row of this overlapping part.

that this subregion takes from each left *edge-object* o_j is calculated and the one with the maximum vote is selected. If $v(s_i, o_j) = 0$ for all o_j, there is no left *edge-object* assignment for the subregion s_i. To calculate $v(s_i, o_j)$, for each row of o_j, the image is scanned towards right starting from the leftmost pixel of o_j in this row and this vote is incremented by one if the scan meets a pixel of s_i along this row. The scan continues until it hits another object or reaches the distance d. This voting is illustrated in Figure 3.5. For the other edge types, the assignment is done in a similar way with the following differences. The image is scanned row-wise for the left and right types and column-wise for the top and bottom types. Additionally, the image is scanned towards the opposite direction of the specified edge type.

The next step determines the subregion pairs to be merged (lines 8-14 of Algorithm 2). For each edge type, every pair of adjacent subregions that share an *edge-object* is identified and a score that quantifies the degree of this sharing is calculated. If they do not share an *edge-object*, this score is set to 0. The subregions s_i and s_k are said to share an *edge-object* if they are assigned to the same object o_j. In this case, the object o_j will vote for the boundary between the subregions s_i and s_k. This vote calculation is very similar to the aforementioned

one, except that each row/column of the object will increment the vote by one if the corresponding scan meets a boundary pixel (instead of a pixel of a subregion) and the scan stops only if it hits another object (instead of also reaching the distance d). After calculating the votes (e.g., the vote $v_{left(s_{ik})}$ that the shared left *edge-object* gives to the merge of the subregions s_i and s_k), the merging scores are defined as

$$\sigma_{left}(s_{ik}) = mean\{\frac{v_{left}(s_{ik})}{height(s_i)}, \frac{v_{left}(s_{ik})}{height(s_k)}\}$$

$$\sigma_{right}(s_{ik}) = mean\{\frac{v_{right}(s_{ik})}{height(s_i)}, \frac{v_{right}(s_{ik})}{height(s_k)}\}$$

$$\sigma_{top}(s_{ik}) = mean\{\frac{v_{top}(s_{ik})}{width(s_i)}, \frac{v_{top}(s_{ik})}{width(s_k)}\}$$

$$\sigma_{bottom}(s_{ik}) = mean\{\frac{v_{bottom}(s_{ik})}{width(s_i)}, \frac{v_{bottom}(s_{ik})}{width(s_k)}\}$$

where each vote is normalized with the size of the subregions. Here we use normalization not to create any bias towards merging larger subregions, for which the number of boundary pixels is expected to be higher. If the pair of s_i and s_k gets sufficient vote from at least one vertical edge type (left or right) and at least one horizontal edge type (top or bottom), the total merging score $\sigma(s_{ik})$ is calculated as the sum of all of its scores and the pair is added to the merge set ϕ. In other words, if $\sigma_{left(s_{ik})} \geq t_{vote}$ or $\sigma_{right(s_{ik})} \geq t_{vote}$ and $\sigma_{top(s_{ik})} \geq t_{vote}$ or $\sigma_{bottom(s_{ik})} \geq t_{vote}$, the merging score $\sigma(s_{ik}) = \sigma_{left(s_{ik})} + \sigma_{right(s_{ik})} + \sigma_{top(s_{ik})} + \sigma_{bottom(s_{ik})}$. Otherwise, this pair will not be qualified for merging.

As the last step, all pairs in ϕ are iteratively merged with respect to their total merging scores (lines 15-19 of Algorithm 2). After merging a pair of the subregions s_i and s_k, the newly emerged subregion s_{ik} is reassigned to the *edge-objects*, the merging scores between this new subregion and its neighbors are recalculated and the merge set ϕ is updated accordingly. Each iteration continues until there is no subregion pair left for merging.

As explained in Section 3.1.1, the *SubregionMerging* procedure is called for different sets of the *edge-objects* O_{perc}, each of which is calculated using a different Otsu threshold percentage constant $perc \in \{0.5, 1.0, 1.5, 2.0\}$ (see lines 6-8 of

Algorithm 1). As a final step, subregions smaller than an area threshold t_{area} will be eliminated to obtain the final segmented nuclei. Here, the majority filter is applied afterwards to obtain smoother boundaries. Note that this majority filter only slightly affects the segmentation performance but it yields smoother boundaries. In this work, we select the radius of this filter as 3 considering the pixel resolution of the images used in our experiments.

3.2 Experiments

3.2.1 Dataset

We test our object oriented algorithm on 2661 cell nuclei of 37 fluorescence microscopy images. The cells were taken from the Huh7 and HepG2 liver cancer cell lines and stained with nuclear Hoechst 33258. The images were taken under a Zeiss Axioscope fluorescent microscope with a Carl Zeiss AxioCam MRm monochrome camera with a 20× Carl Zeiss objective lens. For Hoechst 33258 florescent dye which a bisbenzimide DNA intercalator can be observed in the blue region upon UV region excitation. Hoechst 33258 dye was excited with $365nm$ and the emitted blue light ($420nm$) was acquired, the beam splitter was 395nm. During the image acquisition, binning was set to 1×1, the gain and the offset were set to default 0, and the integration time was 10-40ms. The images were saved in the jpg image format and their pixel resolution was set to 768×1024.

We use 785 nuclei of ten randomly selected images (five Huh7 and five HepG2 cell line images) in the training set, on which the model parameters are estimated. The nuclei in the remaining 27 images are used for testing. HepG2 cells tend to grow in more overlayers than Huh7 cells. This leads to more overlapping nuclei in the images of the HepG2 cell line. Thus, we separately test our algorithm for these cell lines. The Huh7 cell line test set includes 891 nuclei of 11 images and the HepG2 cell line test set includes 985 nuclei of 16 images. The nuclei in these images were manually annotated by our biologist collaborator. The image sets

and their annotations are publicly available at `http://www.cs.bilkent.edu.tr/~gunduz/downloads/NucleusSegData/`.

3.2.2 Evaluation

Each algorithm is quantitatively evaluated by calculating the precision, recall, and F-score metrics at both the nucleus- and pixel-levels. The nucleus-level calculation finds true positive nuclei as follows: it matches a nucleus N segmented by the algorithm with an annotated nucleus A if at least half of the N's pixels overlap with those of A. Similarly, it matches each annotated nucleus with a segmented one. It then considers a segmented nucleus as true positive if there exists one-to-one match between this segmented nucleus and an annotated one. Afterwards, considering the correctly identified pixels of only the true positive nuclei as true positive pixels, the pixel-level precision, recall, and F-score metrics are calculated. Note that in this work, we used the same nucleus- and pixel-level quantitative evaluation with the studies proposed by our research group [3, 58], the results of which will be provided for comparison.

3.2.3 Parameter Selection

Table 3.1 lists the external parameters of the proposed method. We select the values of these parameters on the training set; in this selection, we do not use the test sets at all. For that, we consider a set of values for each parameter, also given in Table 3.1, take the results for all possible combinations of different parameters, and select the combination that yields the highest F-score for the training set. The selected values are $t_{size} = 5$, $d_{max} = 20$, $t_{vote} = 0.1$, and $t_{area} = 400$.

Table 3.1: A list of the external model parameters together with their values considered in parameter selection. The selected values are indicated as bold.

Parameter	Explanation	Values considered
t_{size}	Minimum height/width for a component to be an edge-object	$\mathbf{5}, 10, 15$
d_{max}	Maximum distance within which an edge-object can vote for a subregion	$15, \mathbf{20}, 25$
t_{vote}	Minimum score that a subregion pair should take from at least one vertical (left or right) and at least one horizontal (top or bottom) edge type to be qualified for merging	$\mathbf{0.1}, 0.2, 0.3$
t_{area}	Minimum area for a subregion to be a nucleus	$200, 300, \mathbf{400}, 500$

3.2.4 Results

Quantitative segmentation results of the proposed object oriented method are given in Tables 3.2 and 3.3 for the Huh7 and HepG2 cell line test sets, respectively. These tables show that the object oriented algorithm improves segmentation results at both the nucleus- and pixel- levels. This improvement is higher for the HepG2 cell line test set, which includes more overlapping nuclei. When the results are visually examined, it is observed that the proposed method is able to determine nucleus locations with high success for both less and more overlapping nuclei (some examples are given in Figure 3.6). This is consistent with the nucleus-level evaluation results.

In order to understand its effectiveness, we compare our object oriented method with two of our previous methods [3, 58] and the other three proposed by other research groups [22, 35, 36]. The quantitative and visual results of these comparison methods are also provided in Tables 3.2 and 3.3 and Figure 3.6. These methods could be grouped into three. The first group includes the adaptive h-minima [22] and iterative h-minima [3] methods, which are marker-controlled watersheds. Both of these methods apply h-minima transform to a distance/gradient map to suppress its noise and then identify the regional minima on the noise-suppressed map as their markers. The former one determines and uses a single h value to identify its markers and adaptively changes it to refine the shapes of the

identified markers [22]. On the other hand, the latter method iteratively identifies its markers using a set of multiple h values [3]. Nucleus-level evaluation given in Table 3.2 shows that using a single h value [22] is less efficient to correctly identify many markers, each of which corresponds to a nucleus in the result. When multiple h values are used, more correct markers are found, and as a result, the latter comparison method [3] yields nucleus-level evaluation comparable with our method (it gives 89.29 and 83.22 percent F-scores for the Huh7 and HepG2 cell line test sets, respectively, whereas our method gives just 90.75 and 84.21 percents). On the other hand, pixel-level evaluation given in Table 3.3 reveals that the proposed object oriented method gives much more successful results than both of these comparison methods to delineate the nucleus' boundaries, as also observed in the visual results. For pixel-level evaluation, the proposed method increases the F-score of iterative h-minima from 78.46 to 83.98 percent for the Huh7 cell line test set and from 71.77 to 76.45 percent for the HepG2 cell line test set. This may be attributed to the following reason. After identifying its markers, a marker-controlled watershed algorithm grows these markers pixel-by-pixel usually with respect to pixel gradients and/or distance transforms. This pixel-by-pixel growing is, however, more susceptible to pixel-level noise and imperfections. On the other hand, the proposed object oriented method relies on subregion-level merging with the help of the edge-objects, which are defined to encode gradients at the object-level. This object-level processing results in delineating the nucleus' boundaries more successfully.

The second group of comparison algorithms relies on using pixel-level gradients [35, 36]. The iterative voting method defines a series of oriented kernels to obtain the gradient information and determines nucleus centers by getting image pixels iteratively voted along the directions specified by these kernels [35]. The single-pass voting method improves the nucleus seed detection algorithm by defining a voting area on the eroded binary mask of an image. This method considers only the boundary regions of this binary mask instead of traversing the entire image [36]. Tables 3.2 and 3.3 show that our proposed method leads to higher F-scores compared to these two voting based methods. This indicates the effectiveness of using the gradient information at the object-level instead of using

Table 3.2: Comparison of the algorithms in terms of nucleus-level evaluation on (a) the Huh7 cell line and (b) the HepG2 cell line test sets.

Huh7			
	Precision	**Recall**	**F-score**
Object-oriented [1]	92.33	89.23	90.75
Adaptive h-minima [22]	88.27	83.61	85.87
Iterative h-minima [3]	89.24	89.34	89.29
Iterative voting [35]	81.28	80.92	81.10
Single-pass voting [36]	87.82	85.75	86.77
ARGraphs [58]	88.14	88.44	88.29

(a)

HepG2			
	Precision	**Recall**	**F-score**
Object-oriented [1]	87.00	81.00	84.21
Adaptive h-minima [22]	80.37	69.44	74.50
Iterative h-minima [3]	86.35	80.3	83.22
Iterative voting [35]	75.89	73.19	74.52
Single-pass voting [36]	77.04	72.89	74.91
ARGraphs [58]	81.41	79.19	80.28

(b)

pixel-level gradients. Table 3.2 also shows that both of these comparison methods yield lower nucleus-level precision and recall values. These lower values indicate the detection of less true positives (correctly located nuclei), but also lower precisions are the indicators of more false positives (incorrectly located nuclei) and lower recalls are those of more false negatives (missing nuclei). The increase in the number of false positives and false negatives might be the result of noise and imperfections in pixel values, which will directly affect the computation of the gradients at the pixel-level. Intensity inhomogeneities in a nucleus may lead to defining spurious edges, which increases the number of false positives, whereas insufficient pixel intensity differences at the nucleus' boundaries may cause not to identify existing nuclei, which increases the number of false negatives. The use of the gradient at the object-level alleviates the negative effects of these imperfections, which might be the reason of obtaining higher nucleus-level precision and recall values.

The last comparison group includes the ARGraphs method [58], which was

Table 3.3: Comparison of the algorithms in terms of pixel-level evaluation on (a) the Huh7 cell line and (b) the HepG2 cell line test sets.

Huh7			
	Precision	**Recall**	**F-score**
Object-oriented [1]	80.25	88.08	83.98
Adaptive h-minima [22]	82.57	79.47	80.99
Iterative h-minima [3]	83.58	73.93	78.46
Iterative voting [35]	81.26	68.48	74.33
Single-pass voting [36]	83.61	71.03	76.81
ARGraphs [58]	78.28	85.51	81.74

(a)

HepG2			
	Precision	**Recall**	**F-score**
Object-oriented [1]	71.96	81.53	76.45
Adaptive h-minima [22]	67.16	66.33	66.74
Iterative h-minima [3]	80.09	65.02	71.77
Iterative voting [35]	70.67	61.12	65.55
Single-pass voting [36]	71.70	59.12	64.80
ARGraphs [58]	65.75	75.37	70.24

(b)

implemented by our research group. This method constructs an attributional relational graph on the *edge-objects* and identifies nucleus centers by searching predefined patterns on this graph. This previous method also uses the *edge-objects*, but this use is completely different than the one proposed in this current work. ARGraphs does not define any subregions, and thus, obviously, it does not use these subregions in conjunction with the *edge-objects*. Moreover, it does not make use of any first-divide-then-merge approach to form nuclei from the subregions and the *edge-objects*. We use this comparison method to understand the effects of developing such kind of approach in segmenting the nuclei. Tables 3.2 and 3.3 demonstrate that this newly proposed approach improves the F-scores both at the nucleus- and pixel-levels. This improvement is higher for the HepG2 cell line test set, in which cells tend to grow in overlayers. This reveals the effectiveness of our first-divide-then-merge approach, which first divides an image into subregions and then merges them by the *edge-objects*, to more correctly identify overlapping nuclei in more overlayered cell clumps. This is also consistent with the visual results given in Figure 3.6.

Table 3.4: Comparison of the algorithms in terms of computational times on the Huh7 and HepG2 cell line test sets.

Computational Time (sec)		
	Huh7	**HepG2**
Object-oriented [1]	14.82 ± 10.38	14.11 ± 11.84
Adaptive h-minima [22]	1.81 ± 1.29	1.75 ± 1.36
Iterative h-minima [3]	2.02 ± 1.44	1.38 ± 1.99
Iterative voting [35]	10.03 ± 7.96	9.42 ± 6.11
Single-pass voting [36]	6.52 ± 5.85	4.85 ± 4.47
ARGraphs [58]	5.86 ± 2.08	5.84 ± 2.69

Table 3.4 provides the average computational time to segment nuclei in a given image and its standard deviation. These computational times are obtained on a computer with a 2.9 GHz Intel Core i5 processor and 16 GB of RAM. We implement our object oriented method mostly in Matlab but when faster computations are needed, we write the code in C++ and compile it by the MEX compiler of Matlab. Its average computational time is approximately 15 seconds, which is higher than those of the other comparison methods. The most expensive part of our method is the iterative subregion merging procedure (lines 15-19 of Algorithm 2). At each iteration, this procedure selects one subregion pair from the candidate set, merges them, and updates the assignments and the voting scores of the remaining candidates. This part takes longer time especially when the number of subregions is high. Although it is implemented in C++, it is still possible to make this part faster by more effectively coding it. It is also possible to obtain further speedups by implementing the entire algorithm in C++. This is considered as a future work of this implementation.

3.3 Discussion

This chapter presents a new object oriented method for segmenting cell nuclei in fluorescence microscopy images. This method relies on the use of gradient information at the object-level, instead of directly using pixel-level gradients. To

this end, it proposes to partition an image into smaller subregions, define *edge-objects* at four different orientations for encoding the gradient information at the object-level, and devise an effective merging algorithm that forms nuclei from the subregions with the help of the *edge-objects*. In this subregion-level merging, the *edge-objects* vote for subregion pairs along the direction specified by their edge types and the subregion pairs are iteratively merged provided that they get sufficient votes from multiple directions. This high-level representation together with this high-level merging are expected to be less susceptible to pixel-level noise and imperfections compared to the methods that directly work on pixel values. Our experiments on fluorescence microscopy images are consistent with this expectation. They demonstrate that the proposed object oriented method leads to better segmentation results compared to pixel-level cell nucleus segmentation algorithms.

The proposed method defines four different types for the *edge-objects* but does not define any type for the subregions. It is also possible to assign types to subregions, based on their characteristics, and incorporate them into the merging algorithm. This could be considered as a future work of this study. In this work, we focus on the fluorescence microscopy images. As another future work, one may consider to extend this object oriented method to other types of microscopy images. For instance, it can be extended to 3D nucleus segmentation by defining 3D *edge-objects* and 3D subregions. For that, the third axis (depth) can be employed to identify the edge-objects of six different types (left, right, top, bottom, front, and back object types) and supervoxels can be used to define the subregions instead of superpixels. For obtaining the supervoxels, one can use the option provided by the SLIC algorithm [2]. After defining them, the merging step can be modified to consider object assignments for the new types and to use the votes of the *edge-objects* from six directions. This may be considered as another future work of the proposed study.

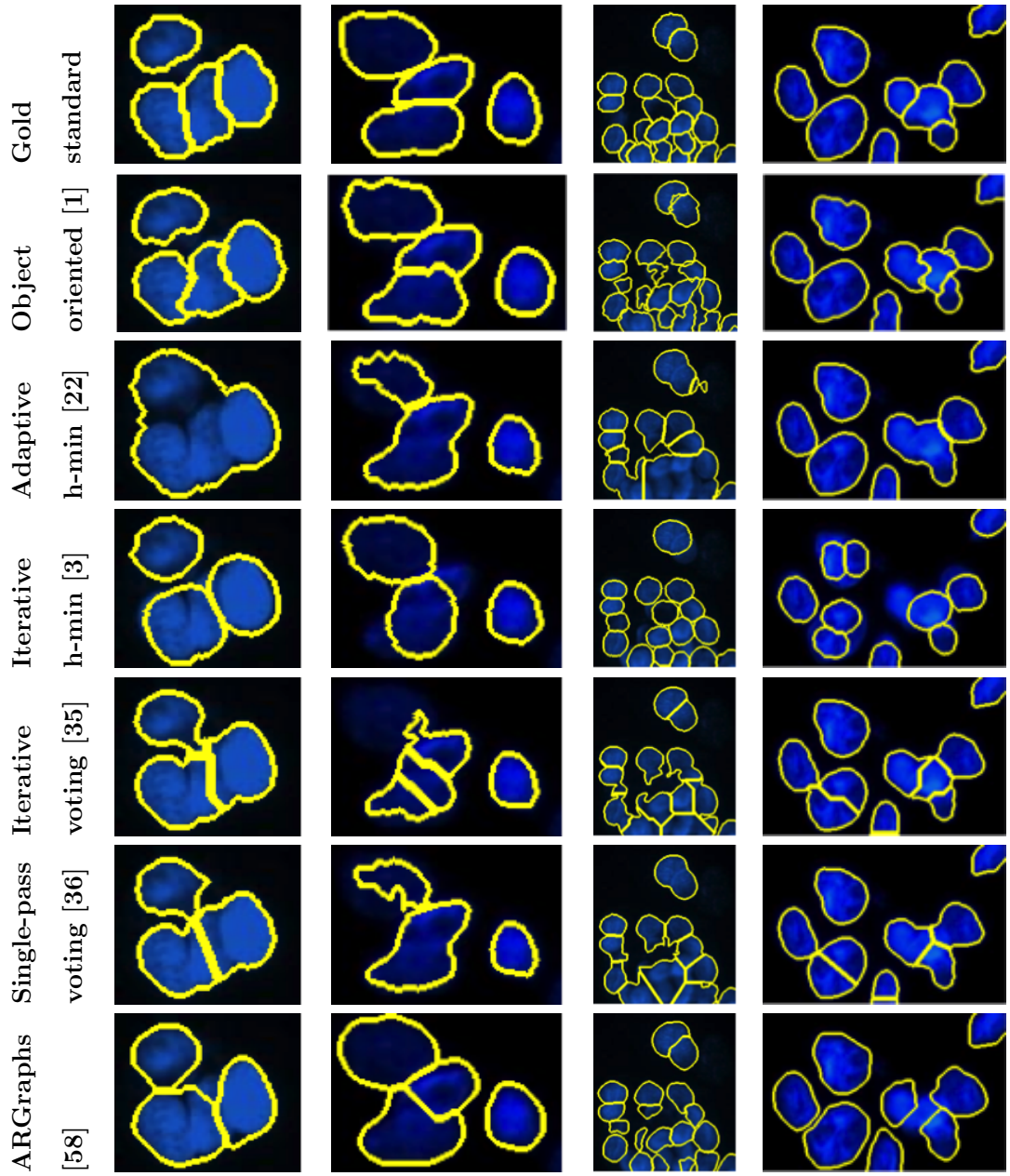


Figure 3.6: Visual results obtained by the algorithms for various subimages. The subimage sizes have been scaled for better visualization.

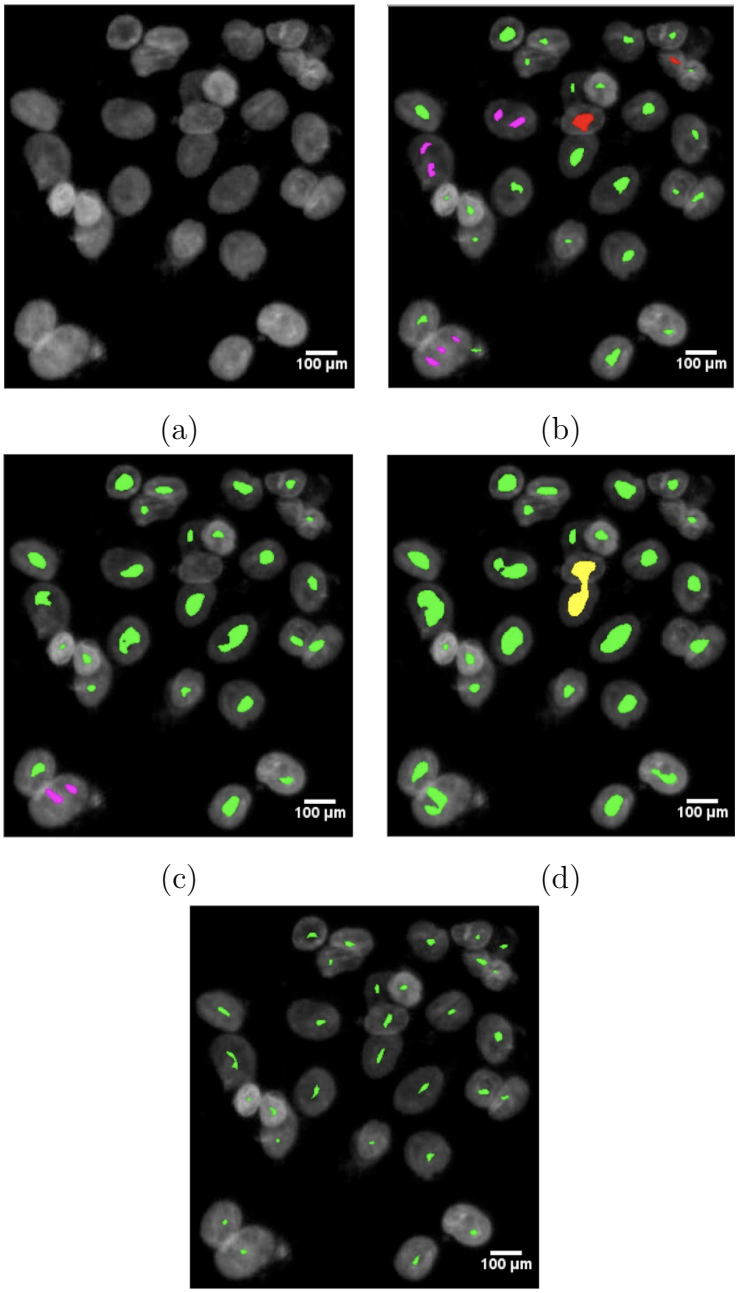
Chapter 4

Iterative H-minima Based Marker-Controlled Watershed for Cell Nucleus Segmentation

The marker-controlled watershed is a commonly used technique that segments clustered cell nuclei. It defines a set of markers on an image and obtains cell nucleus regions growing them only from these predefined markers. In this technique, it is crucial to correctly identify the markers since a nucleus cannot be segmented if a marker is not defined for it. The majority of the previous algorithms take regional minima/maxima found on the intensity/gradient [4] and/or the distance transform [21] maps as the markers. However, this is very sensitive to noise, and hence, may lead to defining spurious markers. To alleviate this problem, these algorithms typically apply the h-minima/h-maxima transform, which suppresses all minima/maxima under a value of h, before finding the regional minima/maxima [28, 22, 23]. The selection of the h value directly affects the defined markers. Smaller h values do not sufficiently suppress the noise, which might result in defining false and over-segmented markers. On the other hand, larger h values suppress too many pixels such that minima/maxima become connected to each other or to the background; this might yield missing and under-segmented markers.

The previous algorithms typically use the same h value for an entire image or for each connected component of the binary mask of the image, which corresponds to a nucleus cluster. They select this h value experimentally [28, 87] or by optimizing a criterion function [23]. Once it is selected, this value is used for the entire image or the corresponding connected component. On the other hand, the same image/component may require using different h values for more accurately identifying the markers. For instance, Figure 4.1 shows the markers found on an example image using three different h values. The cell nuclei illustrated as red markers in Figure 4.1b can only be identified using a smaller h value. However, the same h value yields many over-segmented cell nuclei, markers of which are shown in magenta in Figures 4.1b and 4.1c. Increasing the h value may overcome the over-segmentation problem, but this time, it may cause under segmentations, as illustrated with a yellow marker in Figure 4.1d, and missing nuclei.

In this study, we propose a new marker-controlled watershed algorithm to address this issue. To this end, the proposed algorithm iteratively identifies its markers, considering a set of different h values. In each iteration, it defines a set of candidates using a particular h value and selects the markers from those candidates provided that they fulfill the size requirement. In the literature, there also exist h-minima based methods that make use of iterative approaches to identify their markers [22, 23]. After identifying the initial markers using a selected h value, the shape of these markers are refined by increasing the selected h value iteratively, until the point just before the initial markers start to merge with each other [22]. The method given in [23] determines the h value that optimizes an evaluation function in an iterative algorithm. However, once they fix the h values, these algorithms use them for the entire image/component. Our proposed algorithm differs from these algorithms in the sense that it identifies its markers using multiple h values for the same image/component. By doing so, it alleviates the over and under-segmentation problems due to the use of the same h value for the entire image/component. Our experiments on widefield fluorescence microscopy images demonstrate that this use of multiple h values improves the segmentation performance for nuclei of both isolated and confluent cells.



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

Figure 4.1: Markers found on an example subimage: (a) original subimage, (b) markers when $h = 1$, (c) markers when $h = 2$, and (d) markers when $h = 3$. Here, magenta and yellow markers indicate oversegmentations and undersegmentations, respectively. The markers that cannot be identified with larger h values are shown with red in (b). The markers identified by our proposed algorithm are shown in (e).

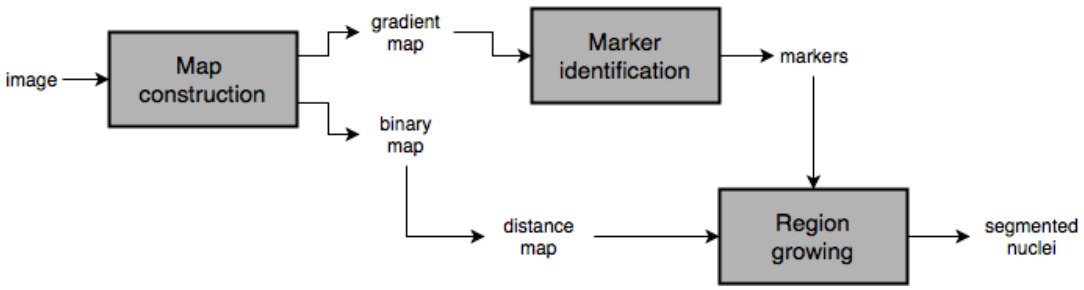


Figure 4.2: Schematic overview of our proposed algorithm.

4.1 Methodology

Our proposed algorithm relies on using multiple h values to identify the markers of a connected component, which corresponds to a nucleus clump in an image. The motivation behind this use is the fact that there exists no best h value that can be used to identify all markers of the same connected component, due to the possible variations in the nuclei's sizes, shapes, and intensities within the same nucleus clump. Our algorithm has three main steps: *map construction*, *marker identification*, and *region growing*. We implement the map construction and marker identification steps in Matlab, using its built-in function for h-minima transform. We implement the region growing step in C. The source codes of our implementation are available at `http://www.cs.bilkent.edu.tr/~gunduz/downloads/IterativeHMin`. The schematic overview of the algorithm is given in Figure 4.2. The details of its steps are given in the following subsections.

4.1.1 Map Construction

In this step, we construct two maps on which initial markers are identified and grown. These are the gradient map G_{map}, which we use to model the intensity deviations along the nucleus boundaries, and the distance transform map D_{map}, which we use to model the size and shape of nuclei. For an image I, we obtain the gradient map G_{map} by applying the Sobel operators on its grayscale. Here

we smooth both the grayscale image and the Sobel responses to reduce intensity variations and noise within nuclei. In particular, before applying the Sobel operators, we smooth the grayscale image by morphological opening that uses a disk structuring element with a radius of d_{size}. Then, after obtaining them, we smooth the Sobel responses using the average filter also with a half size of d_{size}. Note that we select the diameter (radius) of the disk structuring element and the filter size (its half size) the same to reduce the number of free model parameters in our algorithm.

We calculate D_{map} by taking the distance transform for the pixels of a binary mask B, which is obtained by thresholding the grayscale of the image I. In our algorithm, we use a global threshold value calculated by the Otsu's method [86]. However, we use its half to ensure that the mask covers most of the nuclear regions.

4.1.2 Iterative Marker Identification

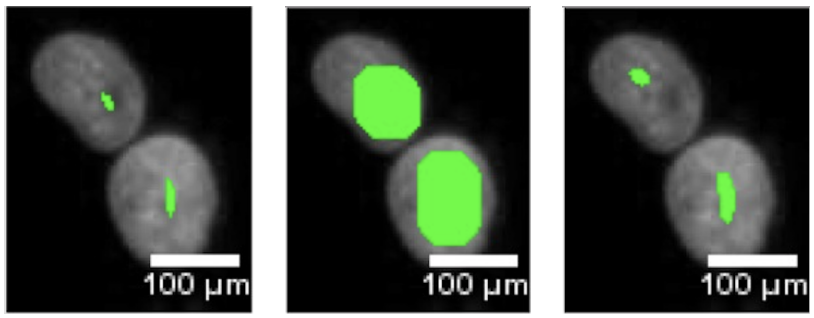
Watershed-based nucleus segmentation algorithms commonly define their markers on nucleus centroids. For that, they typically find regional maxima on a distance transform map, to reflect a fact that nucleus centroids are the locations farthest from boundaries, and/or regional minima on a gradient map, to reflect a fact that the centroids typically show smallest intensity deviations. In this work, we use the gradient map G_{map} to iteratively identify the markers. In each iteration of this process, we first suppress noise on G_{map} using the h-minima transform, with a different h value, and then find the regional minima on the noise-suppressed map. The motivation behind using different h values in different iterations is that the selection of the h value is not straightforward since a single fixed h value would not be enough to suppress all noise at a desired level, and thus, different h values work with different levels of success to identify the markers corresponding to different types of nuclei. Smaller h values work better to identify the correct markers for nuclei containing a fair amount of noise inside, but may yield over-segmented markers for those with a high amount of noise. On the other hand,

larger h values address the over-segmentation problem, but this time, they may lead to under-segmented or missing markers for the former type of nuclei. Thus, in order to address this problem, we proposed to use multiple h values in an iterative algorithm (Figure 4.1).

In this algorithm, we start iterations from $h = 1$ and increment its value by one until no new markers are defined. In each iteration, we suppress noise on G_{map} using the h-minima transform and identify the regional minima on the noise-suppressed map as marker candidates. Then, in order to reduce the number of over-segmented markers, whose areas are typically small especially when a small h value is used, we eliminate the candidates that are smaller than an area threshold t_{area}. We eliminate such small candidates to prevent defining a noisy region as a marker. Note that if such a region corresponds to a true marker, next iterations are expected to locate it since larger h values typically yield larger candidates (regional minima).

At the end, we add the candidates to the marker set provided that they do not overlap with the markers defined in the previous iterations. Here instead of considering the previous markers as they are, we dilate them with a disk structuring element, whose radius is also d_{size}, and determine the overlaps accordingly. The rationality of this dilation is that consecutive h values may yield overlapping markers or those that are not overlapping but very close to each other and the dilation prevents over-segmentation arising from such close markers (Figure 4.3).

We provide the pseudocode of this marker identification in Algorithm 3. The algorithm takes three inputs. The first one is the gradient map G_{map}, on which markers are identified. The next one is the area threshold t_{area}, which is used to eliminate small marker candidates. The last one is the radius d_{size} of a disk structuring element, which is used to dilate the previous markers for determining the overlaps. The iterative marker identification algorithm outputs the marker set M. Figure 4.4 illustrates an example output of this algorithm, each iteration of which uses a different h value. Each image shown in this figure corresponds to a different iteration and illustrates the markers added to the marker set in the current iteration in red and those found in the previous iterations in green.



(a) (b) (c)

Figure 4.3: (a) Previously identified markers before dilation, (b) previously identified markers after dilation, and (c) currently identified markers. There is no overlap between the top marker of (a) and the top marker of (c) before dilation. However, after dilation, these two become overlapping and the top marker of (c) will not be included into the marker set, which prevents oversegmentation for the top nucleus.

4.1.3 Region Growing

After identifying the markers, we grow the dilated markers on the foreground pixels of the binary mask B by a marker-controlled watershed algorithm and delineate the nucleus boundaries. We use the distance transform map D_{map} as the marking function in the flooding process of the watershed. For each foreground pixel, D_{map} keeps the closest distance from this pixel to its closest marker. In a standard watershed algorithm, the flooding process grows the identified markers on all foreground nucleus pixels until the grown markers meet. However, this may cause a problem when markers are not correctly identified for all adjacent nuclei. Figure 4.5 illustrates this problem on two subimages, each of which contains three nuclei. In each subimage, the markers are correctly identified for the two nuclei but no marker is found for the other nucleus (Figure 4.5a). The standard flooding process grows these markers on the nucleus pixels, whose boundaries are given in Figure 4.5b. Thus, it yields incorrect nucleus boundaries, as shown in Figure 4.5c, since some of these pixels belong to the nucleus with an unidentified marker.

To prevent flooding into pixels that belong to a nucleus with an unidentified marker, we modify the flooding process such that it grows a marker on a foreground pixel unless it meets the stopping condition for this pixel, which is defined

Algorithm 3 *IterativeMarkerIdentification*: Procedure that iteratively identifies markers.

Input: gradient map G_{map}, area threshold t_{area}, disk size d_{size}
Output: markers M

1: $M = \emptyset$
2: $h = 1$
3: **repeat**
4: $\quad h_{map} = HMinima(G_{map}, h)$
5: $\quad M_{curr} = RegionalMinima(h_{map})$
6: $\quad M_{curr} = EliminateSmall(M_{curr}, t_{area})$
7: $\quad M_{curr} = EliminateOverlapping(M, M_{curr}, d_{size})$
8: $\quad M = M \cup M_{curr}$
9: $\quad h = h + 1$
10: **until** $M_{curr} = \emptyset$

considering other pixels found in its symmetric location. Particularly, to grow a marker M on a foreground pixel P, we check all pixels found on a circular arc, whose midpoint is symmetric to P with respect to the M's centroid. The start and end angles of the arc are $-\alpha$ and $+\alpha$ degrees with respect to the line passing through this midpoint and the M's centroid (see Figure 4.6). We allow growing only if none of the pixels on this arc belong to the background or have previously been assigned to another marker. At the end, when none of the markers can be grown further, we allow them to grow on the foreground at most p more pixels without considering the stopping condition. For the subimages given in Figure 4.5a, the boundaries obtained by our modified flooding process are shown in Figure 4.5d. Note that since this flooding process considers pixels on an arc, instead of an entire circle, it locates non-circular nuclei better, as illustrated in Figure 4.7.

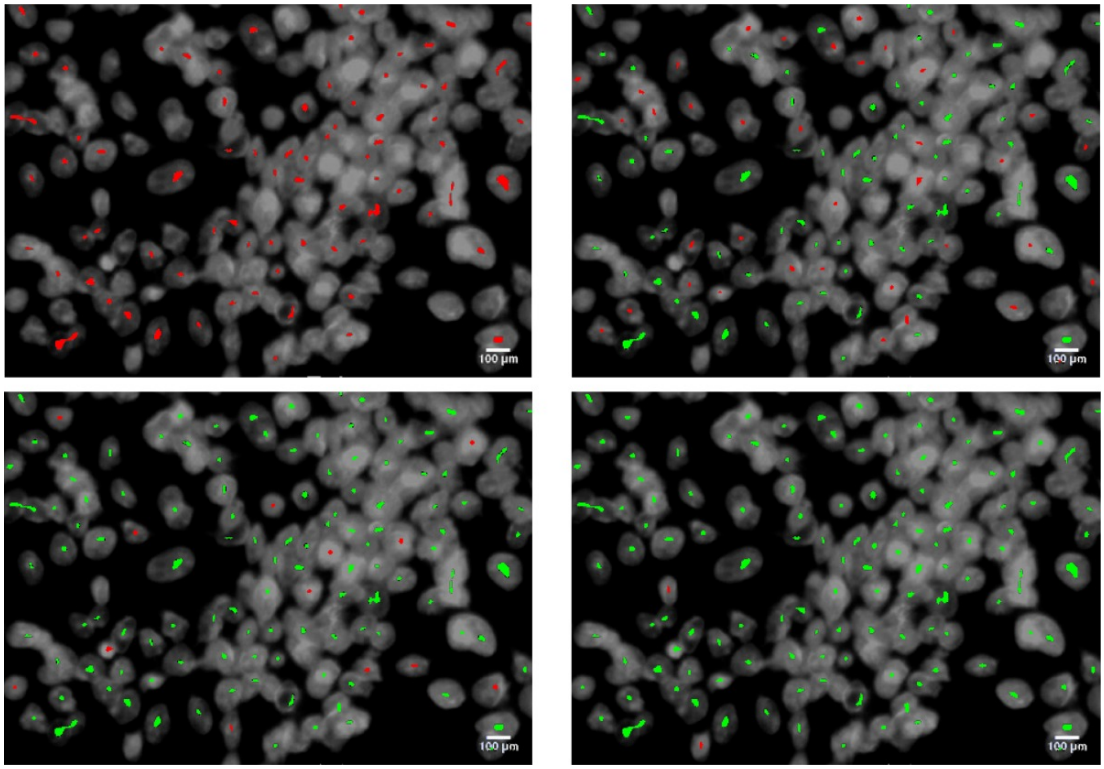
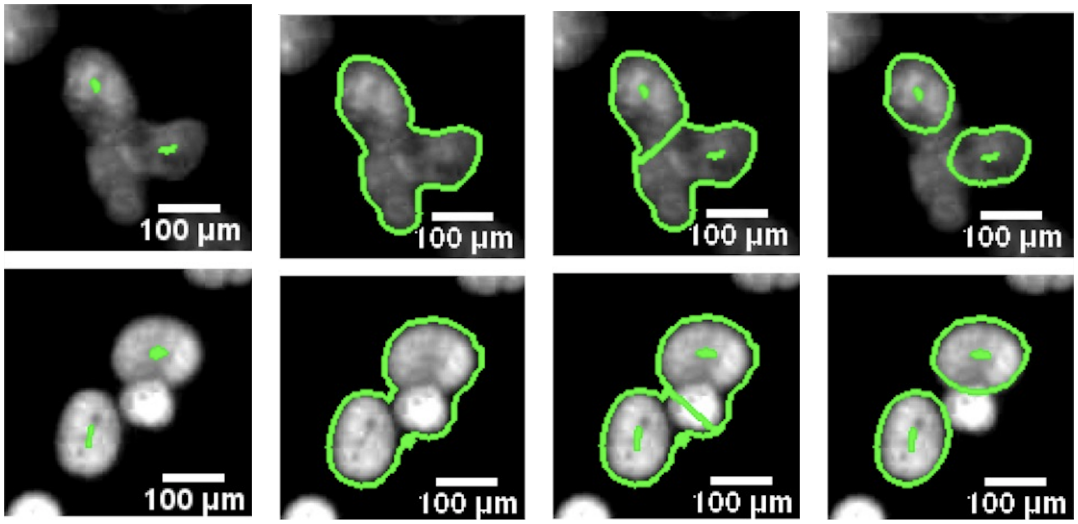


Figure 4.4: Outputs of four different iterations, each of which uses a different h value, in the marker identification step. In each image, the red markers are the ones that are added to the marker set in the current iteration and the green markers are those that were found in the previous iterations.

4.2 Experiments

4.2.1 Dataset

In our experiments, we use fluorescence microscopy images of human hepatocellular carcinoma (Huh7 and HepG2) cell lines that were cultured in the Molecular Biology and Genetics Department at Bilkent University. The cells were stained with Hoechst 33258 nuclear staining and their images were taken under a Zeiss Axioscope fluorescent microscope with an AxioCam MRm monochrome camera. The objective lens is 20× and the image size is 768×1024. The cell nuclei in these images were annotated by our biologist collaborators.



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

Figure 4.5: Flooding process of the watershed algorithm for two example subimages. (a) Markers from which flooding starts, (b) boundaries of nucleus pixels, (c) nucleus boundaries obtained using the standard flooding process, and (d) nucleus boundaries obtained using our flooding process.

First, we conduct experiments on the dataset containing the fluorescence microscopy images of Huh7 and HepG2 cell lines. In this dataset, 785 nuclei are used as training instances, on which the parameters of the algorithms are selected. These nuclei are taken from 10 randomly selected images; five of them are selected from the Huh7 cell line and the other five from the HepG2 cell line. The rest of the images are used as test instances. Since cells are grown in more overlayers in the HepG2 cell line and since we want to explore the effectiveness of the algorithms on different confluency levels, there are two test sets. The first one contains 891 nuclei taken from 11 images of the Huh7 cell line. The second one contains 985 nuclei taken from 16 images of the HepG2 cell line. In addition to these test sets, we form another one that contains more confluent cells. This test set contains 1065 nuclei taken from 4 images of the HepG2 cell line. We will refer them as the *Huh7 test set*, the *HepG2 test set*, and the *dense HepG2 test set*, respectively.

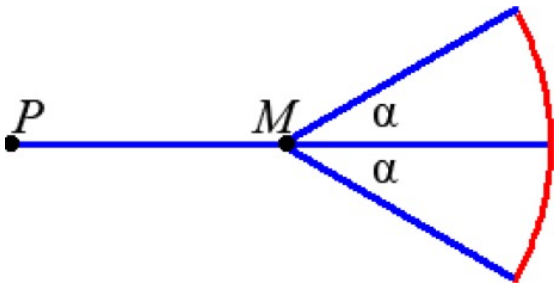
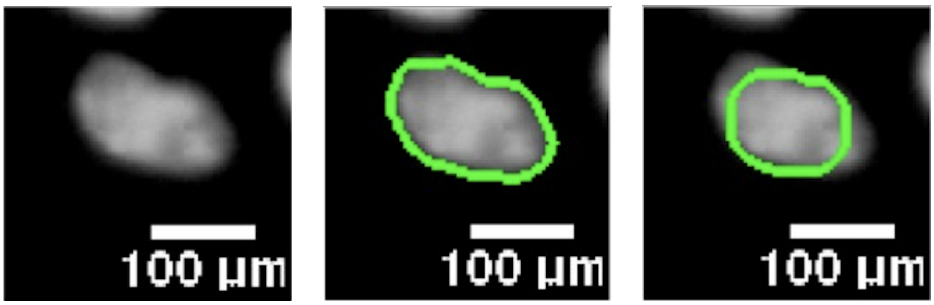


Figure 4.6: Illustration of defining the stopping condition in region growing. To grow a marker M on a pixel P, this condition checks all pixels (red pixels in this figure) on a circular arc, whose mid-point is symmetric to P with respect to the M's centroid.



(a) (b) (c)

Figure 4.7: Effects of defining the stopping condition by considering the pixels of an arc instead of an entire circle. (a) An example image of nucleus, (b) boundaries obtained when the pixels of an arc are considered, and (c) boundaries obtained when the pixels of an entire circle are considered.

4.2.2 Evaluation

We evaluate our proposed algorithm and the comparison methods, both visually and quantitatively. For quantitative evaluation, we use the precision, recall, and F-score metrics. First, we calculate these metrics on nuclei to quantify how successful an algorithm is in the correct identification of nuclei. Then, we calculate them on pixels by considering the correctly segmented pixels of only the correctly identified nuclei as correct segmentation.

Note that we use the same approach to evaluate our method which we explained in Chapter 3. To explain the evaluation method again: We match each nucleus N that an algorithm segments with an annotated nucleus A in the gold standard

if at least half of N's segmented pixels overlap with those of A. Likewise, we match each annotated nucleus with a segmented nucleus. Then, N is considered as correctly identified if there is a one-to-one match between N and an annotated nucleus. Otherwise; 1) N is a false detection if it does not match with any annotated nuclei, 2) A is a miss if it does not match with any segmented nuclei, 3) A is over-segmented if more than one segmented nucleus match with A, and 4) annotated nuclei that match with the same segmented nucleus are under-segmented.

4.2.3 Parameter Selection

The proposed algorithm has four external parameters. The first one is the area threshold t_{area}, which is used to eliminate smaller markers in the marker identification step. The second parameter d_{size} is used in two different steps: map construction and marker identification. In the map construction step, it determines the size of the disk structuring element and the average filter, both of which are used for smoothing operation. In the marker identification step, this parameter also determines the size of the disk structuring element, which is used to dilate the previous markers for eliminating the overlapping markers. Note that although it is possible to use different values, we set the radius of the disk structuring elements and the half size of the average filter to the same d_{size} value to reduce the number of the external parameters of our algorithm. The last two parameters are used in the region growing step. The angle α is used to define the start and endpoints of an arc, whose pixels are used to define the stopping condition of the flooding process. The offset p is the maximum number of pixels that a marker grows at the end without considering the stopping condition. In our experiments, we consider any combination of the following values $t_{area} = \{5, 10, 20, 30\}, d_{size} = \{5, 7, 10, 13\}, \alpha = \{0, 15, 30, 45\}$, and $p = \{0, 2, 4\}$, and select the one that maximizes the F-score metric on the training set. The selected parameter values are $t_{area} = 20, d_{size} = 10, \alpha = 15$, and $p = 2$. In this selection, none of the test set images are used. In addition to these external parameters, we have an internal choice, which is the decrease ratio of the Otsu

threshold to obtain the binary mask B in the map construction step. In this step, we decrease the Otsu threshold to its half (i.e., use the 0.5 ratio) to ensure that B covers most of the nucleus pixels. We will analyze the effects of this selection to the segmentation performance in Section 4.3.1.

4.2.4 Comparisons

We compare our proposed algorithm with four nucleus segmentation methods: adaptive h-minima [22], conditional erosion [19], iterative voting [35], and ARGraphs [58]. The first two are marker-controlled watersheds. The adaptive h-minima method [22] identifies markers by finding regional minima on the inverse distance map, which is also explained in Chapter 3. The conditional erosion method [19] finds its markers by iteratively eroding the binary mask of an image using two different structuring elements. The iterative voting [35] and the ARGraphs [58] methods are also explained in Chapter 3 since they are compared with our iterative h-minima method. You can find the details of the three algorithms, namely adaptive h-minima [22], iterative voting [35], and ARGraphs [58], in Chapter 3. Note that we select the parameters of these four comparison methods also on the training set images.

4.3 Results

We provide the quantitative results of our algorithm and the comparison methods in Figure 4.8 and report their nucleus-based F-score metrics in Tables 4.1, 4.2, and 4.3. The figure and the tables show that the proposed algorithm improves the segmentation performance of the other methods. This improvement is more evident in more confluent cells, as seen in the results obtained on the *dense HepG2* test set (Table 4.3). These quantitative results are also consistent with the visual ones given in Figures 4.10, 4.11, and 4.12. Figure 4.10 contains subimages taken from the *Huh7* test set, which typically have nuclei of isolated and less confluent cells. All algorithms give good segmentation results for almost all of such nuclei.

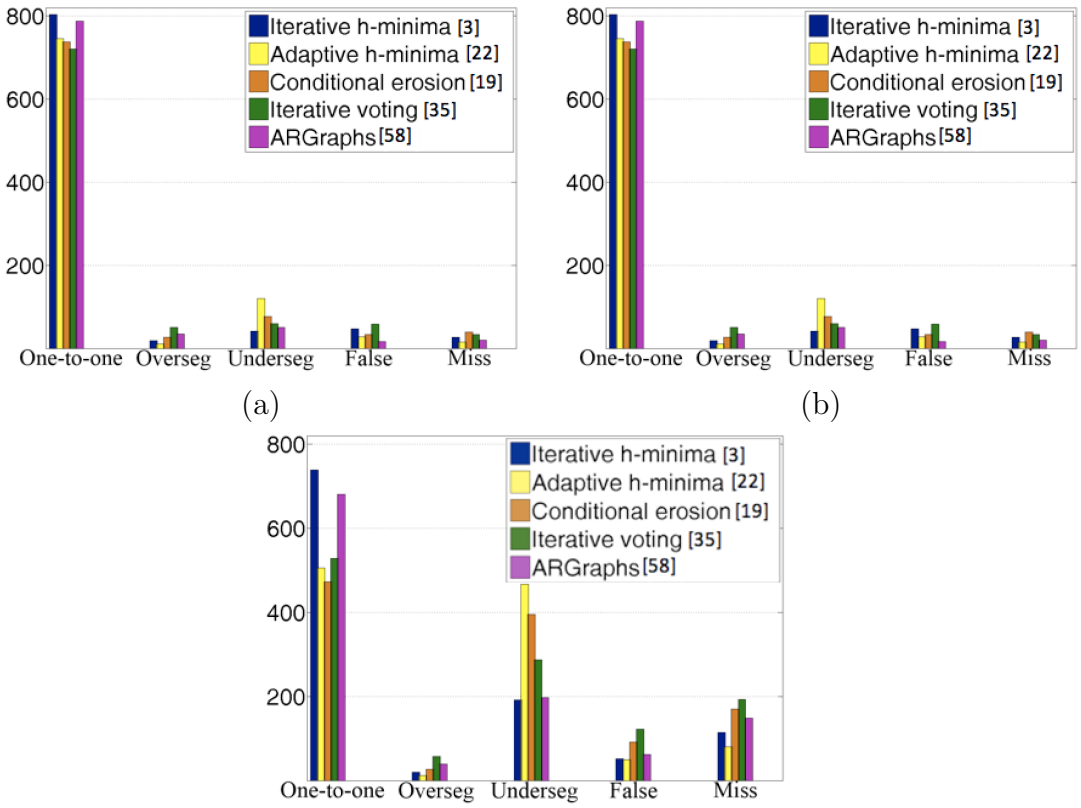


Figure 4.8: Comparison of the algorithms in terms of segmented-annotated nucleus matches on the (a) *Huh7*, (b) *HepG2*, and (c) *dense HepG2* test sets.

Figure 4.11 contains subimages from the *HepG2* test set and Figure 4.12 contains subimages from the *dense HepG2* test set. These visual results show that as the confluency degree increases, the performance of the comparison methods decreases more compared to our proposed algorithm.

Table 4.1: Comparison of the algorithms in terms of nucleus- and pixel-based precision, recall, and F-score measures (a) and segmented-annotated nucleus matches (b) on the *Huh7 test set*.

	Nucleus-based			Pixel-based		
	Precision	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
Iterative H-minima [3]	89.24	89.34	89.29	83.58	73.93	78.46
Adaptive H-minima [22]	88.27	83.61	85.87	82.57	79.47	80.99
Conditional erosion [19]	85.21	82.82	84.01	86.11	72.61	78.78
Iterative voting [35]	81.28	80.92	81.10	81.26	68.48	74.33
ARGraphs [58]	88.14	88.44	88.29	78.28	85.51	81.74

(a)

	One-to-one	**Overseg**	**Underseg**	**False**	**Miss**
Iterative H-minima [3]	796	31	22	17	42
Adaptive H-minima [22]	745	11	120	28	15
Conditional erosion [19]	738	27	77	34	49
Iterative voting [35]	721	51	60	59	34
ARGraphs [58]	788	35	51	17	20

(b)

Table 4.2: Comparison of the algorithms in terms of nucleus- and pixel-based precision, recall, and F-score measures (a) and segmented-annotated nucleus matches (b) on the *HepG2 test set*.

	Nucleus-based			Pixel-based		
	Precision	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
Iterative H-minima [3]	86.35	80.30	83.22	80.09	65.02	71.77
Adaptive H-minima [22]	80.37	69.44	74.50	67.16	66.33	66.74
Conditional erosion [19]	73.89	62.63	67.80	64.24	55.86	59.76
Iterative voting [35]	75.89	73.19	74.52	70.67	61.12	65.55
ARGraphs [58]	81.41	79.19	80.28	65.75	75.37	70.24

(a)

	One-to-one	**Overseg**	**Underseg**	**False**	**Miss**
Iterative H-minima [3]	791	31	48	37	115
Adaptive H-minima [22]	684	4	280	43	17
Conditional erosion [19]	617	15	297	57	56
Iterative voting [35]	721	58	131	75	45
ARGraphs [58]	780	37	116	44	52

(b)

Table 4.3: Comparison of the algorithms in terms of nucleus- and pixel-based precision, recall, and F-score measures (a) and segmented-annotated nucleus matches (b) on the *dense HepG2 test set*.

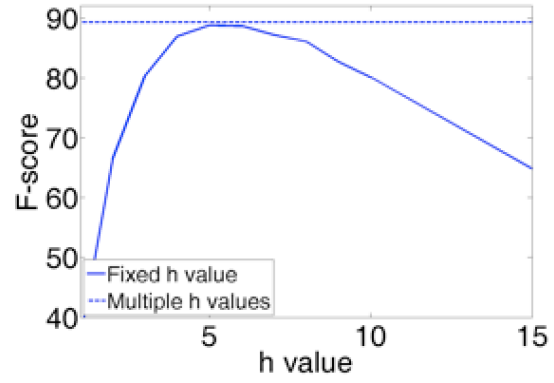
	Nucleus-based			**Pixel-based**		
	Precision	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
Iterative H-minima [3]	84.48	70.05	76.59	74.94	59.89	66.58
Adaptive H-minima [22]	71.37	47.51	57.05	48.89	44.08	46.36
Conditional erosion [19]	61.67	44.41	51.64	49.22	39.82	44.02
Iterative voting [35]	58.41	49.58	53.63	52.52	42.32	46.87
ARGraphs [58]	74.34	63.94	68.75	59.69	62.87	61.24

(a)

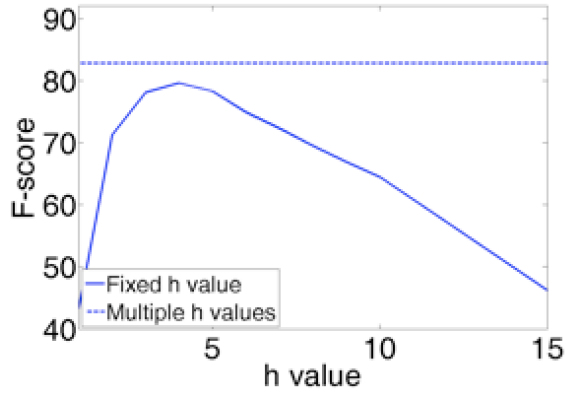
	One-to-one	**Overseg**	**Underseg**	**False**	**Miss**
Iterative H-minima [3]	746	22	126	32	171
Adaptive H-minima [22]	506	12	467	49	80
Conditional erosion [19]	473	27	395	91	170
Iterative voting [35]	528	57	287	122	193
ARGraphs [58]	681	39	197	62	148

(b)

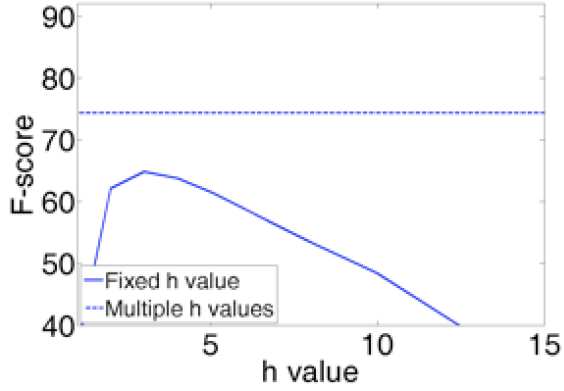
The comparison between our proposed algorithm and the adaptive h-minima method also reveals that using multiple h values to identify the markers for the same connected component leads to better segmentation results. To investigate whether this is indeed a result of using multiple values or improper selection of the fixed h value, we conduct another experiment. For that, we have modified our algorithm such that it uses a single fix h value; the other parts of the algorithm remain exactly the same. For the *Huh7*, *HepG2*, and *dense HepG2* test sets, Figure 4.9 shows the nucleus based F-score metric as a function of h values. For each test set, it also plots the nucleus based F-score metric obtained by our algorithm, which iteratively uses multiple h values. This figure shows that it is possible to obtain a similar F-score metric when the optimal h value is used for the *Huh7* test set, in which cell nuclei are isolated or less confluent (Figure 4.9a). On the other hand, the gap between the F-scores obtained by the proposed algorithm and the optimal h value increases for the *HepG2* and *dense HepG2* test sets, in which cell nuclei are more confluent. This indicates the effectiveness of using multiple h values, especially when cell nuclei form denser clusters.



(a)



(b)



(c)

Figure 4.9: Nucleus based F-score metrics obtained when a fixed h value is used (solid lines) and when multiple h values are iteratively used by our proposed algorithm (dashed lines). The F-score metrics are obtained for the (a) *Huh7*, (b) *HepG2*, and (c) *dense HepG2* test sets.

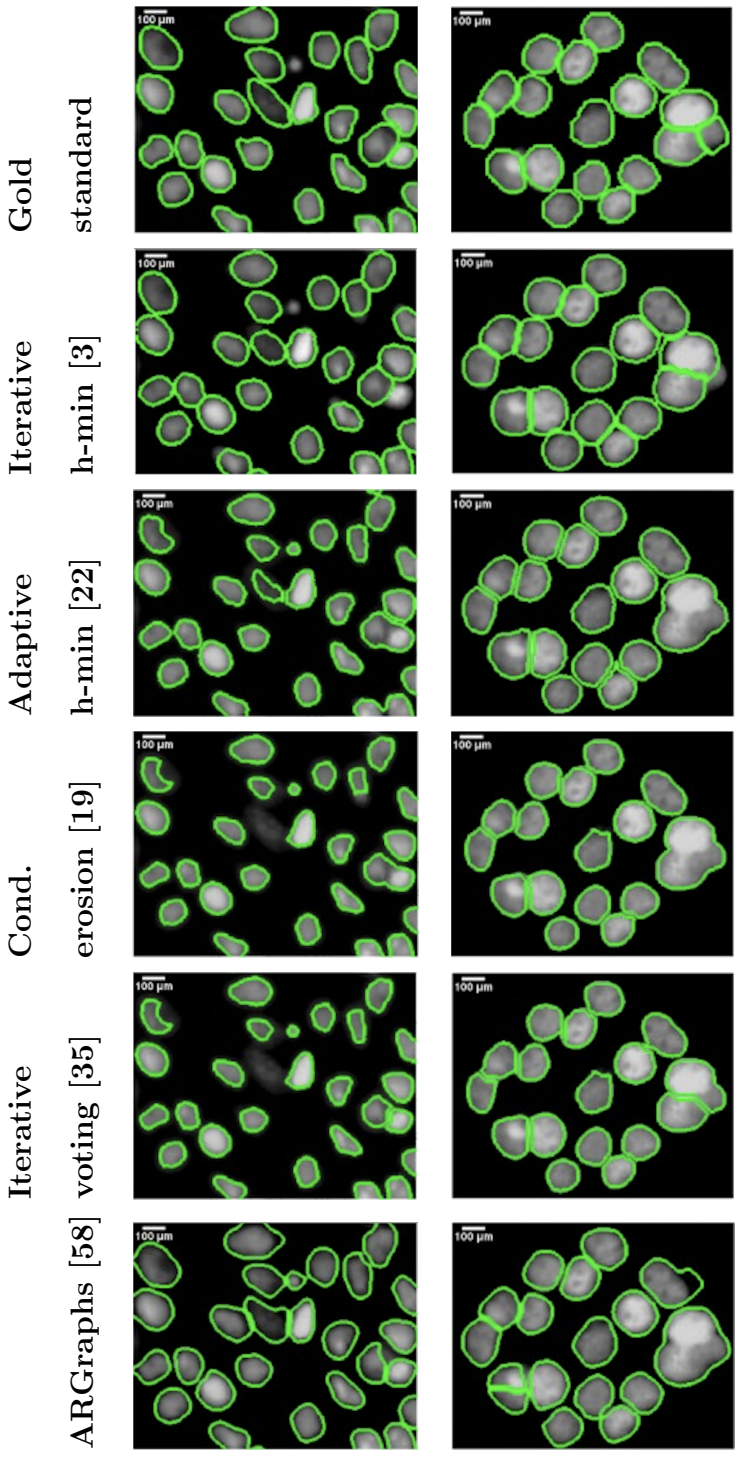


Figure 4.10: Visual results obtained by the algorithms for various subimages. The subimages are from the *Huh7* test set. Note that the subimage sizes have been scaled for better visualization.

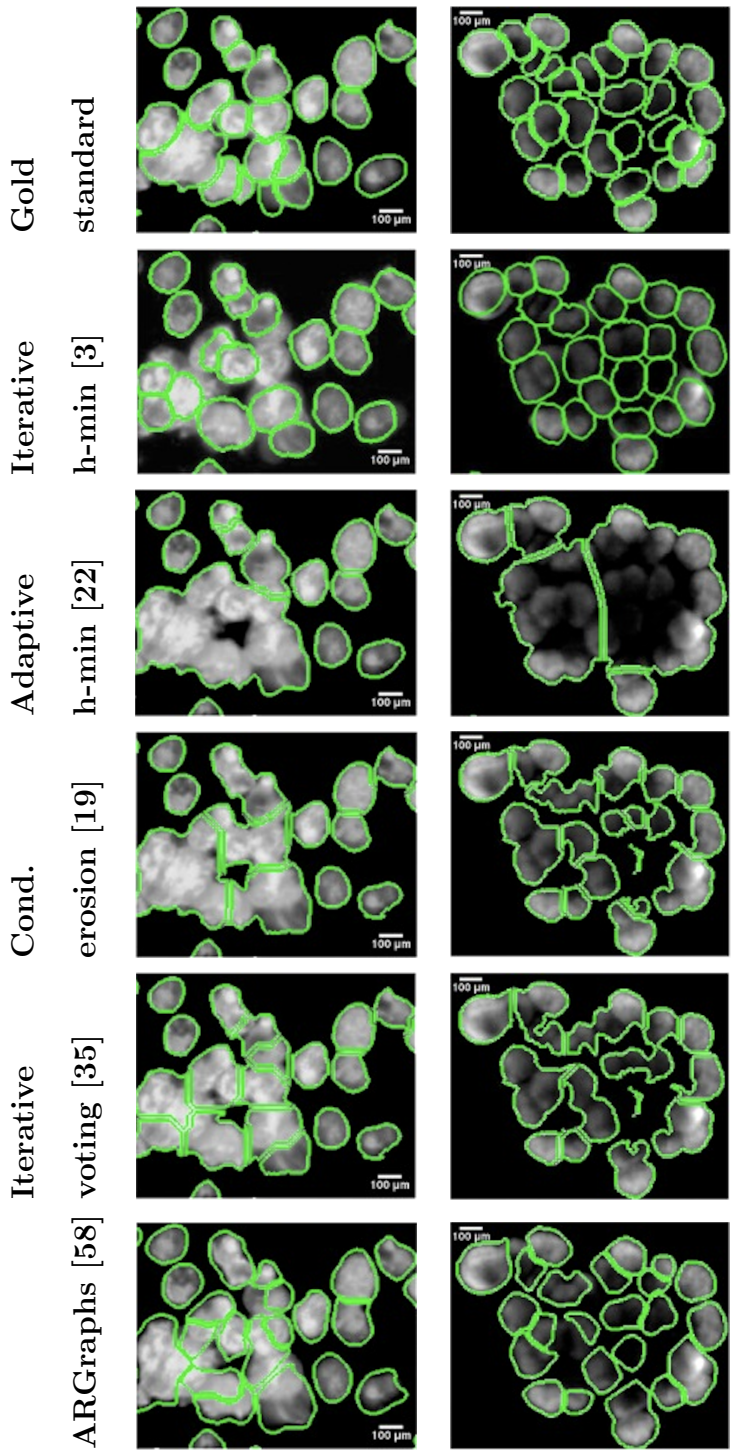


Figure 4.11: Visual results obtained by the algorithms for various subimages. The subimages are from the *HepG2* test set. Note that the subimage sizes have been scaled for better visualization.

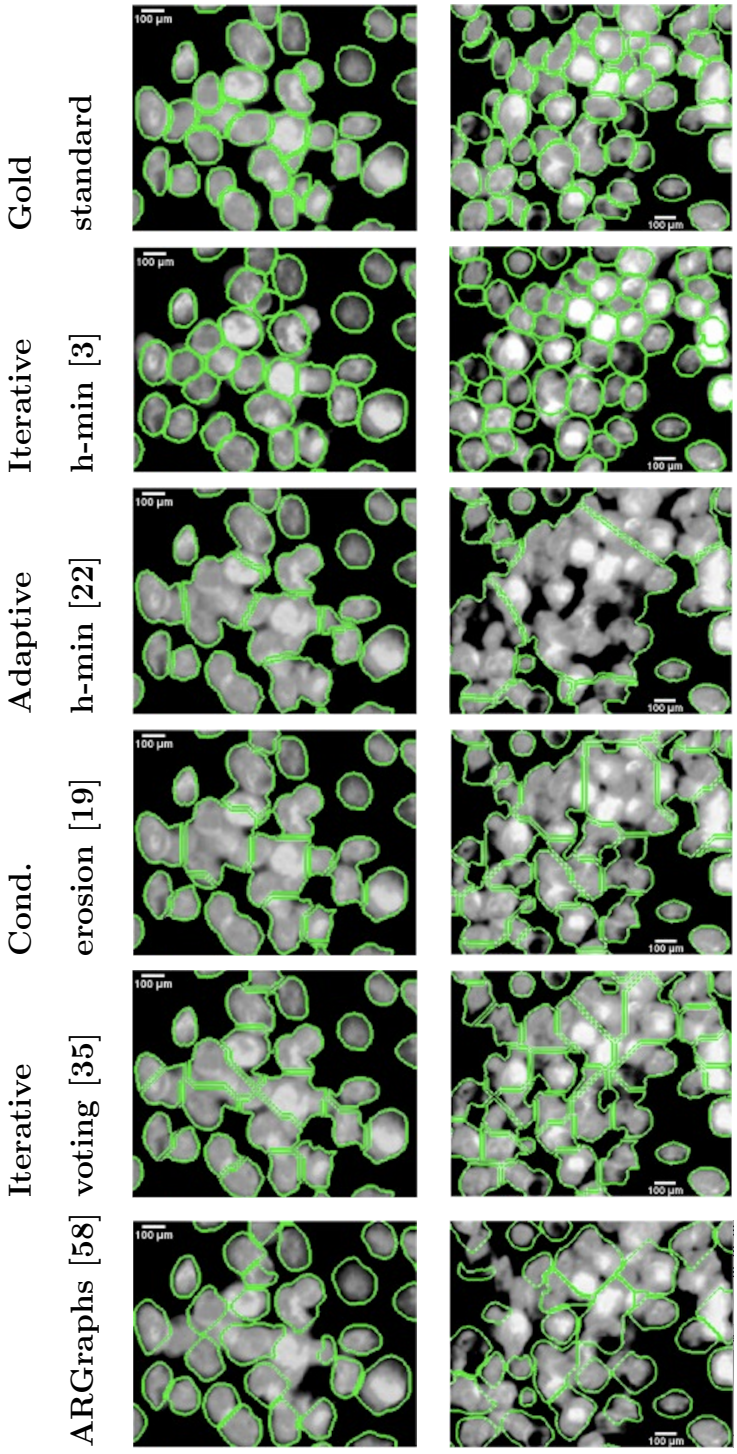


Figure 4.12: Visual results obtained by the algorithms for various subimages. The subimages are from the *dense HepG2* test set. Note that the subimage sizes have been scaled for better visualization.

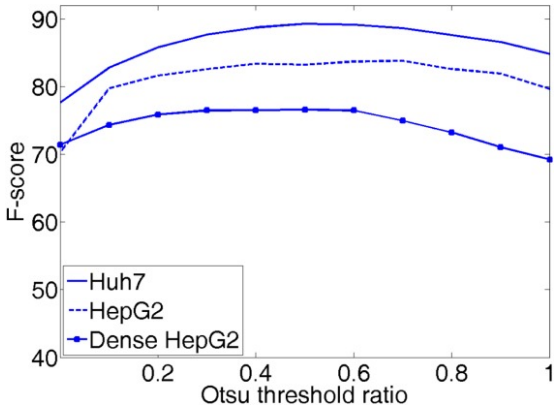


Figure 4.13: For the *Huh7*, *HepG2*, and *dense HepG2* test sets, the nucleus-based F-score metrics as a function of the Otsu threshold ratio used to obtain the binary mask.

4.3.1 Analyses

Our proposed algorithm has four external parameters: the area threshold t_{area}, the size d_{size} of the disk structuring elements and the average filter, the angle α, and the offset p. As explained in Section 4.2.3, we select the values of these parameters on the training set, without using any test set images at all. Besides, the algorithm has an internal choice, which is the Otsu threshold ratio. Although this ratio could also be considered as an external parameter and its value could also be selected on the training set, we fix it to 0.5 for reducing the number of free model parameters in our algorithm. In this section, we will first analyze the effect of this choice to the segmentation results.

To identify the foreground pixels, we obtain a binary mask B by thresholding the gray scale image. Here we calculate the threshold value by the Otsu's method and decrease this value to its half to ensure that the mask covers most of the nucleus pixels. However, instead of decreasing the value to its half (i.e., using the 0.5 ratio), it is also possible to use other decrease ratios. In Figure 4.13, we analyze the effects of using different Otsu threshold ratios to the F-score metrics for the three test sets used in our experiments. This figure indicates that ratios in the range of 0.4 and 0.8 give similar results and the segmentation performance does not very much depend on a specific value of this ratio.

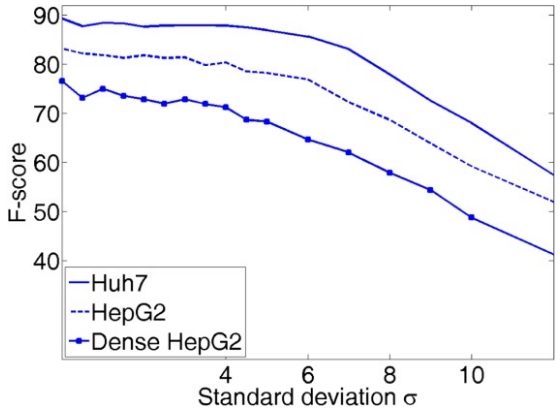


Figure 4.14: Effects of image quality degradation to segmentation results. For the *Huh7*, *HepG2* and *dense HepG2* test sets, the nucleus-based F-score metrics as a function of the standard deviation σ of a Gaussian filter, with which images are blurred. Note that Poisson noise is also added to each blurred image.

Next, we analyze the effects of image quality degradation to segmentation results. To this end, we degrade the quality of images by blurring them with a Gaussian filter and added Poisson noise to the blurred image. Figure 4.14 shows the F-score metric as a function of the standard deviation σ of the Gaussian filter, which controls the degradation degree. This figure shows that our proposed algorithm is robust to image quality degradation to a certain extent. However, as expected, when the image quality drops below a certain point (when the standard deviation σ too much increases), there is a substantial decrease in the segmentation performance.

4.3.2 Experiments on Tissue Section Images

In our experiments, we test our proposed algorithm on the images of cultured human hepatocellular carcinoma (Huh7 and HepG2) cell lines. To understand its applicability on different image types, we extend the application of our algorithm on images of tissue sections from mouse liver, which were stained with 4?,6-diamidino-2-phenylindole (DAPI) nuclear stain. The images of these tissue sections were taken under a fluorescent microscope with a 20$\times$ objective lens. The image size is 480×640. Our biologist collaborators annotated these images

Table 4.4: For the experiments on tissue section images, comparison of the algorithms in terms of the precision, recall, and F-score measures. The results are obtained on the test images.

	Huh7	**HepG2**	**Dense HepG2**
Iterative h-minima [3]	84.78	87.96	86.34
Adaptive h-minima [22]	84.30	73.71	78.65
Conditional erosion [19]	86.77	75.50	80.75
Iterative voting [35]	81.60	75.61	78.49
ARGraphs [58]	85.46	77.88	81.49

by marking the cell nuclei without drawing their boundaries. Because these annotations do not include the nucleus boundaries but a marker for each nucleus, we consider a segmented nucleus as a one-to-one match if this nucleus contains only a single marker, which indicates a gold standard nucleus, inside. For quantitative evaluation, we compute the precision, recall, and F-score metrics on these one-to-one matches.

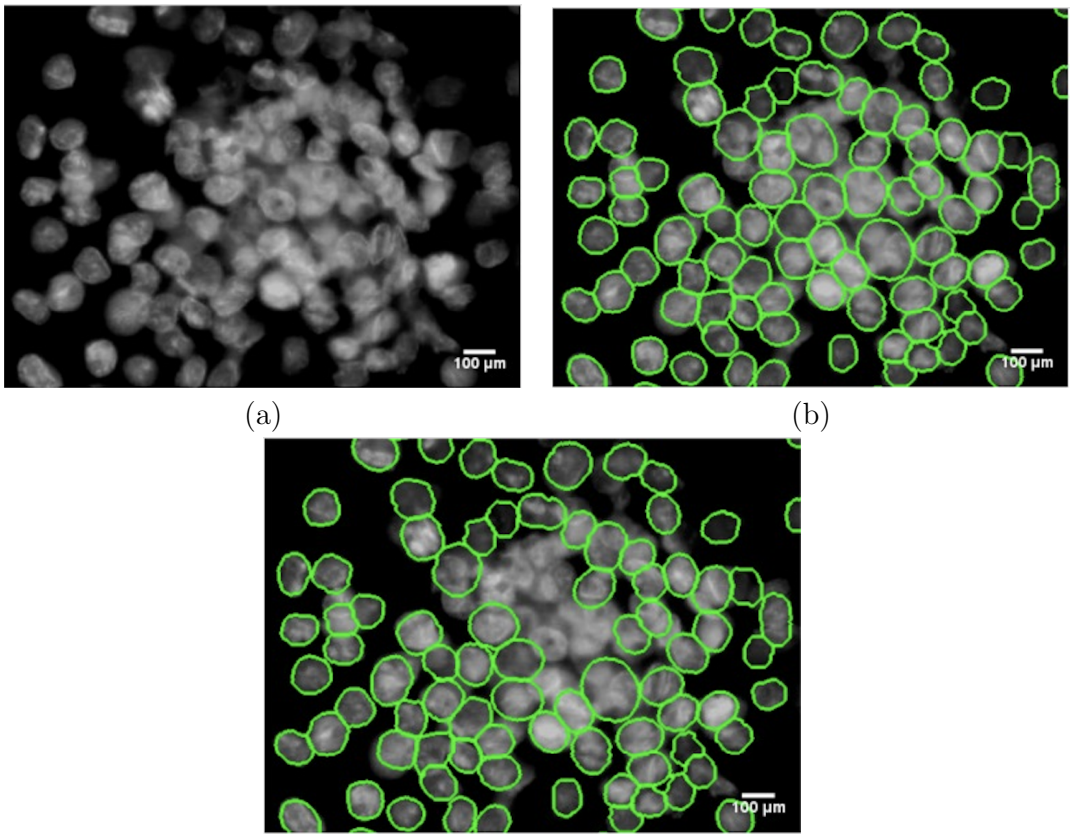
In this tissue section dataset, there are a total of 13 images containing 2,660 cell nuclei. Because these images may show characteristics different than those of cultured human hepatocellular carcinoma cell lines, we randomly separate them into the training set (766 nuclei from four images) and the test set (1,894 nuclei from the remaining nine images) and select the model parameters again on the training nuclei. In this selection, we consider any combination of the following parameter values $t_{area} = \{5, 10, 15, 20, 30\}, d_{size} = \{3, 5, 7, 10, 13\}, \alpha = \{0, 15, 30, 45\}$, and $p = \{0, 2, 4\}$, and select the one that maximizes the F-score metric on the training nuclei. The selected parameter values are $t_{area} = 15, d_{size} = 5, \alpha = 30$, and $p = 2$. Likewise, we select the parameters of the comparison methods again, considering the training set of these tissue sections. On the test set nuclei, our proposed algorithm gives 86.34 percent F-score metric, leading to the highest F-score compared to the other methods. The test set F-scores are 78.65 percent for the adaptive h-minima method [22], 80.75 percent for the conditional erosion method [19], 78.49 percent for the iterative voting method [35], and 81.49 percent for the ARGraphs method [58]. We also present the visual results obtained on three example subimages in Figure 4.16. These preliminary results indicate that the proposed algorithm has a potential to be applied on other image types as

well. One could consider the detailed investigation of this application as a future research direction of the proposed segmentation algorithm.

4.3.3 Tight Nucleus Cluster Detection

Some images may contain tight clusters of nuclei, which cannot accurately be analyzed even manually. To identify such kind of clusters, we develop a simple detection algorithm, which determines markers whose likelihood of corresponding to nuclei in a tight cluster is high and eliminates these markers before region growing takes place. To this end, for each identified marker M, we calculate the minimum distance from its centroid to the background and the distance to the closest marker's centroid. We eliminate the marker M if both of these distances are greater than the distance threshold. The motivation behind using this method is the following. For a tight cluster that contains indiscernible nucleus boundaries, the gradient map is not too much informative. As a result, only a few correct markers can be found within this tight cluster. Additionally, since such a cluster is typically large in size, these markers are usually far from the background.

In our experiments, we select the distance threshold as 30 considering the average radii of cell nuclei in the training images. As expected, the proposed tight nucleus cluster detection method does not eliminate any markers from the *Huh7* test set since this set contains relatively less confluent cells. On the other hand, it eliminates one marker from the *HepG2* and six markers from the *dense HepG2* test sets, which contain more cells grown in overlayers. For an example subimage, taken from the *dense HepG2* test set, the segmentation results obtained with and without using this detection method are given in Figure 4.15. As seen in this figure, no nuclei are found within the tight cluster of this subimage since the corresponding markers have been eliminated by the proposed detection method. Please note that the use of this method slightly changes the F-score metrics for the *HepG2* sets; it changes the F-scores from 83.22 to 83.16 percent for the *HepG2* test set, and from 76.59 to 76.31 percent for the *dense HepG2* test set.



(a) (b)

(c)

Figure 4.15: Visual segmentation results obtained when the tight nucleus cluster detection method is used. (a) Original subimage from the *dense HepG2* test set, (b) nucleus boundaries obtained when the detection method is not used, and (c) nucleus boundaries obtained when the detection method is used.

4.4 Discussion

This study presents a new marker-controlled watershed algorithm for cell nucleus segmentation in fluorescence microscopy images. In this algorithm, we propose to define the markers iteratively, using a different h value in each iteration. The use of different h values suppresses noise at different levels, allowing us to define better markers for nuclei showing different characteristics. Our experiments on widefield fluorescence microscopy images demonstrate that this algorithm gives better markers for nuclei of both isolated and confluent cells, leading to better segmentation results.

In this work, we develop an algorithm for segmenting nuclei of both isolated and confluent cell taken from conventional widefield fluorescent microscopes, which are highly available and affordable for various laboratories. They are found in every molecular biology laboratory as well as they are routinely used for morphological analysis of cells in pathology diagnostics laboratories. However, we do not focus on confocal microscopy, which produces cell images with higher magnification and resolution for detailed visualization of subcellular distribution of fluorescent-labeled proteins. Although our algorithm can also be used for confocal microscopy images, simpler segmentation techniques would also be adequate for their segmentation since these images have only a few cells that are of higher magnification and resolution and that are mostly isolated (nonconfluent). However, the confocal microscopes may not be affordable for every research laboratory. Moreover, the interest may be the confluent cells if a researcher aims to see the aggregation of cells (e.g., cancer stem cell mammosphere formation). In such cases, our proposed algorithm can be used for cell nuclei segmentation.

We conduct our experiments on the images of cultured human hepatocellular carcinoma (Huh7 and HepG2) cell lines. To understand the applicability of our proposed algorithm on different image types, we also extend the application of our algorithm on images of tissue sections from mouse liver and obtain the preliminary results. The application of our algorithm on other image types could be considered as a future work.

In this work, we mainly focus on finding better markers. We use a relatively simple region growing algorithm to delineate nucleus boundaries. As another future work, we plan to work on designing better techniques for marker growing. Here one could consider designing iterative methods also in the region growing process. Another possibility is to explore the use of other types of maps, on which the growing takes place.

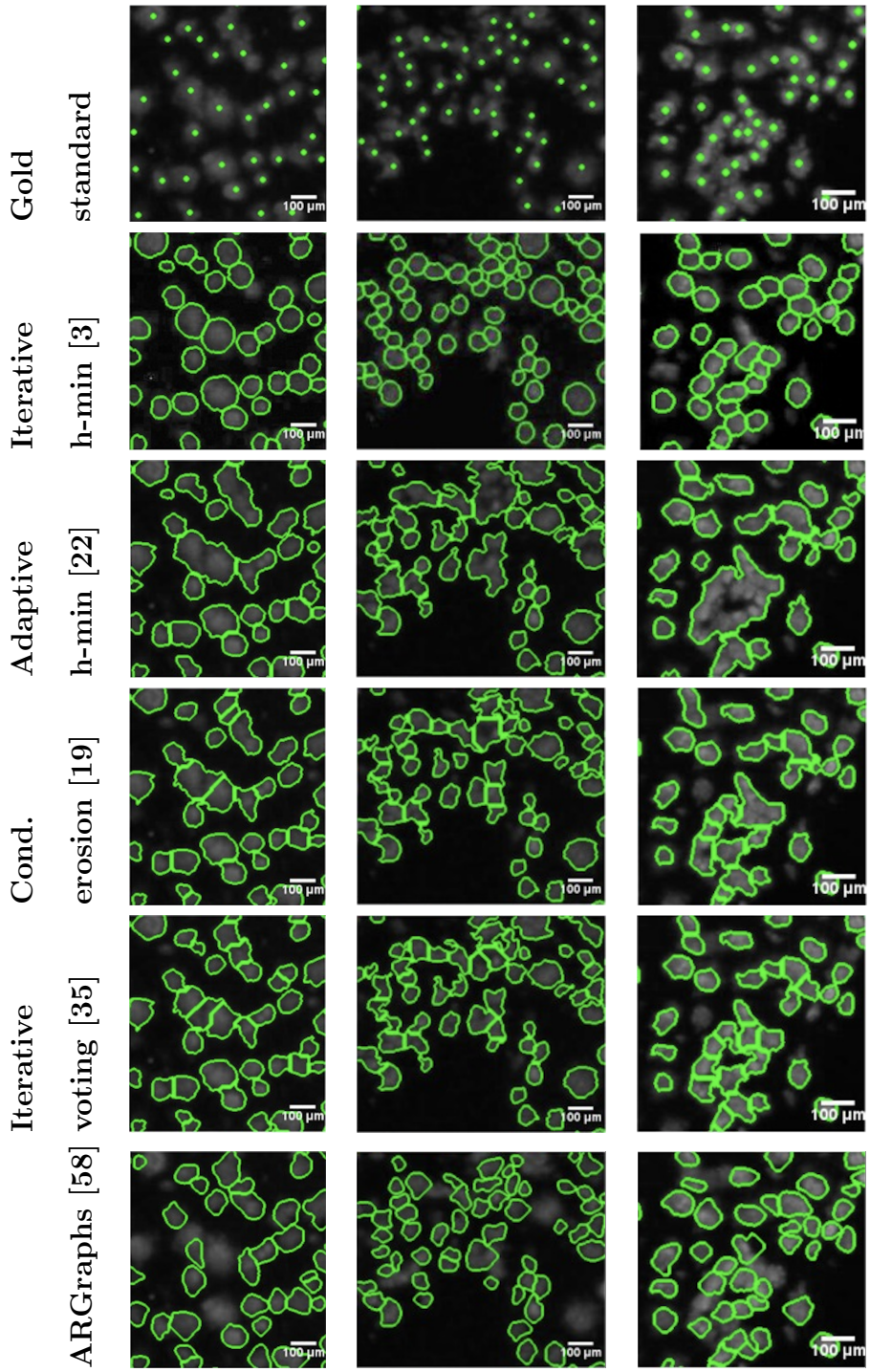


Figure 4.16: Visual results obtained by the algorithms for various tissue section subimages. Note that the subimage sizes have been scaled for better visualization.

Chapter 5

Deep Learning for Cell Detection in Live Cell Images (*DeepDistance*)

This study presents a new deep model that learns high-level features directly from image data. In particular, this model proposes to pose cell detection as a regression problem, in which the normalized distance from each pixel to the closest background is learned by training a deep convolutional neural network (CNN) on small patches cropped around the pixels. Then, for a given image, cell detection is achieved by generating a normalized distance map of the image pixels with the trained CNN and finding regional maxima of the generated map. Since the proposed model automatically learns features from the image data, it is not necessary to redesign any of its steps for new cell types, even when they show different visual characteristics. For a new cell type, our model uses the same CNN architecture and the same cell detection steps; it only needs a couple of annotated training images of the new type, on which the CNN is to be fine-tuned. Our experiments on six different cell line types, examples of which are shown in Figure 5.1, reveal that the proposed model successfully identifies the cell locations, improving the results of both the traditional approach and the previous deep learning based methods.

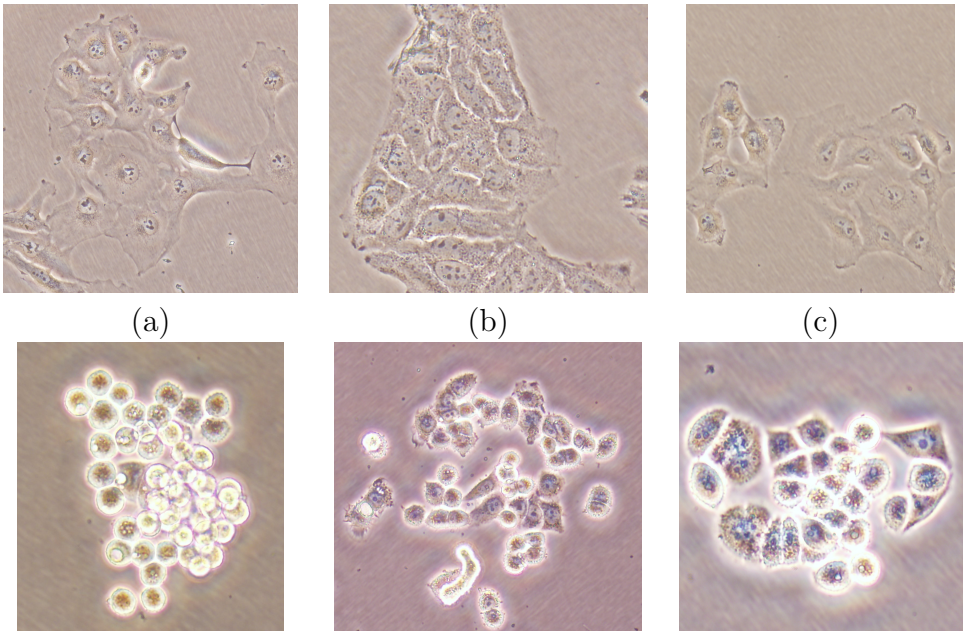


Figure 5.1: Example images from human liver and breast cancer cell lines used in our experiments: (a) FOCUS, (b) Huh7, (c) SkHep1, (d) MDA-MB-453, (e) MDA-MB-468, and (f) CAMA-1 cell lines.

The proposed *DeepDistance* model is different than the previous cell detection models in the following aspects. As opposed to the traditional approach where the features are handcrafted and the model is designed considering the visual cell characteristics, *DeepDistance* provides a generic cell detection model where the features are automatically learned from the image data. Our experiments show that this feature learning allows us to use exactly the same model for different cell types and improves the results of the previous studies. In the literature, there also exist studies that use deep learning for cell detection. Our *DeepDistance* model is also different than these studies. First, different than our model, the majority of these studies pose cell detection as a classification problem in which a deep classifier is trained on small patches to differentiate cell pixels from those of the background. Then, for a given image, they may obtain a binary mask by estimating the class labels of the image pixels with the trained classifier and use this mask as an input to shape-based methods [68, 69]. Alternatively, they may use the class posteriors of the pixels and identify cell locations on this posterior map by either thresholding [66, 70] or clustering [67] but mostly finding regional

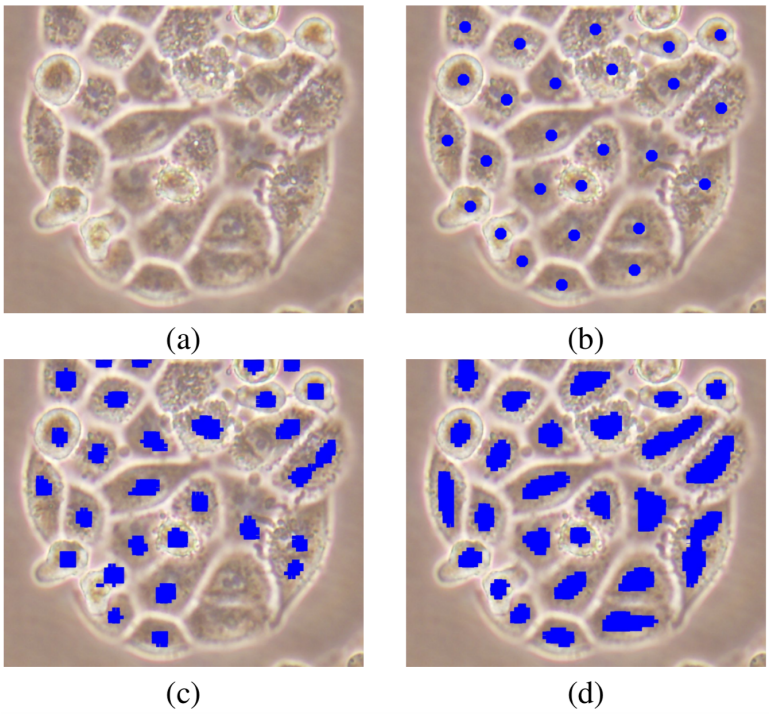


Figure 5.2: Detection results on an exemplary subimage. (a) Original subimage, (b) annotated subimage where cells are marked with blue dots, (c) cell locations detected by a model that uses the distance from each cell pixel to the closest *cell center annotation* as the output, and (d) cell locations detected by the proposed *DeepDistance* model, which uses the normalized distance from each cell pixel to the closest *cell boundary annotation* as the output. Since the *DeepDistance* model does not impose any shape on cells, its results better preserve the cells' shapes, improving the detection results.

maxima [71, 72, 77, 74, 75]. Since their focus is the classification of cell pixels, these studies treat the pixels taken from the annotated cells in the same way, regardless of their relative positions within the cell, while training their classifier. On the other hand, the position of a pixel relative to a cell center (or to a cell boundary) may bring about additional information.

There exist studies that take this information into account by constructing a regression model that learns a continuous distance output for each pixel. To learn this output, these studies construct a training set by taking small patches of pixels as the input and calculating an output distance for the pixels. In [73, 78], this output is the Euclidean distance from the pixel to its closest annotated cell

center. In [82], it is the response of a Gaussian which is located around the closest annotated cell center of the pixel. These definitions, however, impose a circular shape on cells, which may not be true for all cell types. As opposed to these previous studies, the proposed *DeepDistance* model does not have such imposition. It uses the boundary annotations (not the annotated centers) and estimates the normalized distance from each cell pixel to the closest cell boundary designing a deep regression model (it estimates zero for background pixels). Our experiments demonstrate that this distance estimation better preserves shape characteristics of the cells, and as a result, improves the success of cell detection (see Figure 5.2 for exemplary results, more can be found in Section 5.2).

5.1 Methodology

The proposed *DeepDistance* model relies on defining a distance metric that better preserves morphological characteristics of cells, constructing a deep convolutional neural network (CNN) to estimate this distance metric for image pixels, and using the estimated distances for the purpose of cell detection. The following subsections give the details.

5.1.1 Distance Definition

The premise of our model is that it will be quite straightforward to successfully detect cells in an image when an ideal distance map can be obtained for image pixels. In such an ideal map, the distance for pixels close to a cell center will be the largest and it will gradually decrease to zero towards cell boundaries. Moreover, the definition of this ideal distance should not impose any assumption on the shape and size of the cells if the aim is to detect cells of different visual characteristics observed within and across various image samples.

In response to this premise, the *DeepDistance* model defines its distance metric

as follows when cell (or nucleus[1]) annotations are given. Let $\mathcal{P}(A_k) = \{p_i\}$ be the set of pixels belonging to an annotated cell A_k, $\mathcal{B}(A_k) = \{b_i\}$ be the set of its boundary pixels, and $maxD(A_k)$ be the Euclidean distance from the centroid of A_k to its furthest boundary pixel $b_{max} \in \mathcal{B}(A_k)$. The distance d_q for pixel q is defined as

$$d_q = \begin{cases} \dfrac{\min\limits_{b_i \in \mathcal{B}(A_k)} \|q - b_i\|^2}{maxD(A_k)} & \text{if } q \in \mathcal{P}(A_k) \\ 0 & \text{if } q \in \text{background.} \end{cases} \tag{5.1}$$

Note that the calculation of this distance requires the cell annotations, which are, of course, not available for images whose cells are supposed to be automatically detected. Thus, our model proposes to estimate this distance by a deep CNN that will be trained on the pixels of annotated images. The details of this CNN and its training are given in Section 5.1.2.

For an example subimage given in Figures 5.3a and 5.3b shows the distance map calculated using Equation 5.1. As shown in the figure, this distance definition preserves the morphological characteristics of cells. This is attributed to the following properties of this definition: First, the distance is calculated with a reference to a cell boundary instead of a cell center. If the center was used as a reference point, the distance decrease from the center to the cell boundaries would be the same for all directions, which would impose a circular shape on the cells (Figure 5.3c). Second, this definition normalizes the distance with respect to the size (maximum radius) of the cell that a pixel belongs to. This normalization is effective to obtain similar distances for cells of different sizes, which will drive the CNN classifier to make better generalizations regardless of the cell size. As a result, this classifier is able to yield accurate estimated distances (Figure 5.3d), on which the cell detection steps will be run (Section 5.1.3). Note that the subimage used in this figure is not a part of a training image used in our experiments, and thus, its pixels are not used to train the CNN classifier.

[1]Since the aim of cell detection is to identify the locations of cells (but not to segment their exact boundaries), detecting the cells and detecting the nuclei of these cells will be equivalent. Therefore, the annotations of cell nucleus boundaries will be used when delineating the exact cell boundaries (especially for irregular shaped cells) is difficult.

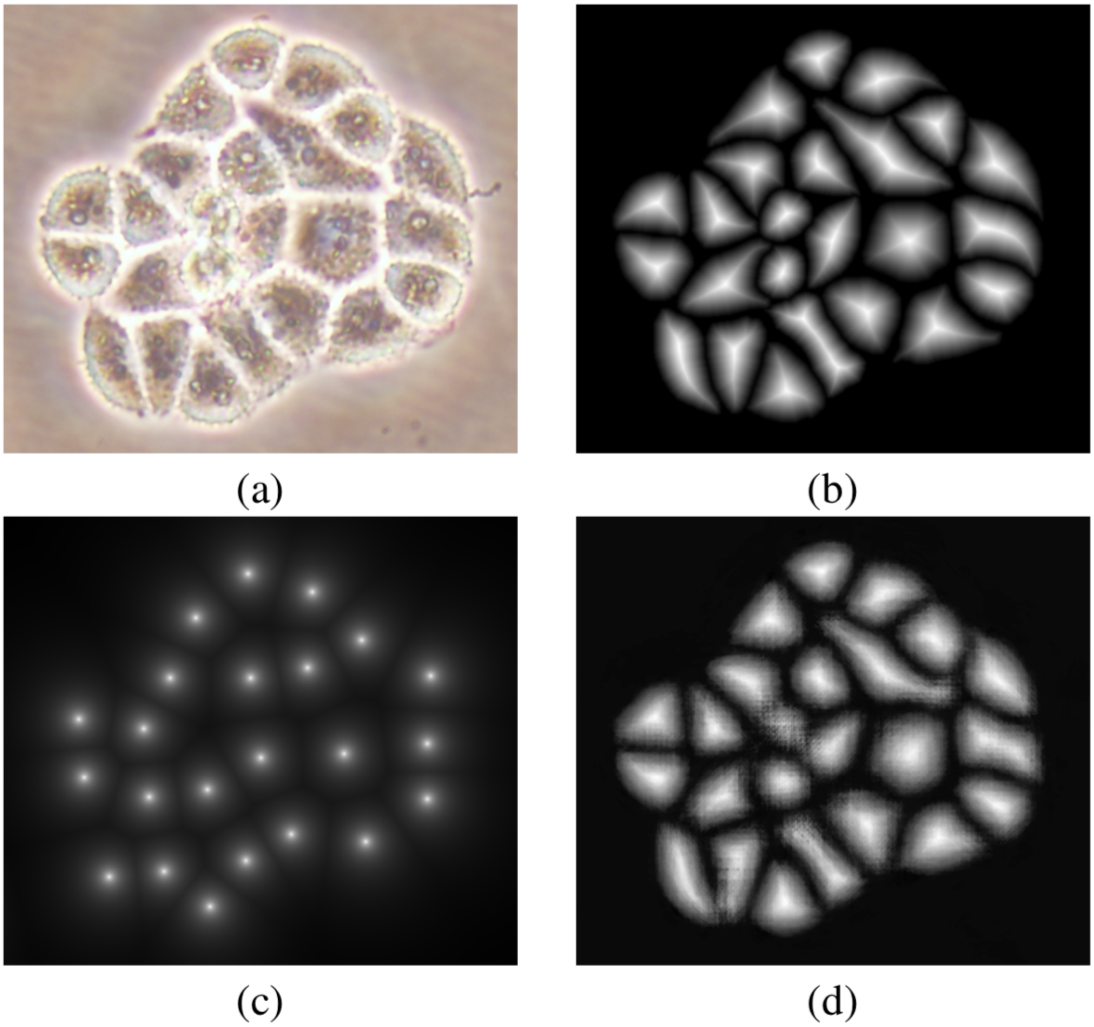


Figure 5.3: (a) Original subimage, (b) normalized distance map obtained by the proposed distance metric, which calculates distances from cell pixels to their closest boundary annotations, (c) normalized distance map obtained when the distance is calculated from cell pixels to their closest cell centers, and (d) distance map estimated by our CNN classifier. Note that the subimage used in this figure is not a part of a training image used in our experiments, and thus, its pixels are not used to train our CNN classifier.

5.1.2 Distance Learning by CNN

This step trains a deep CNN to estimate the distance (given in Equation 5.1) for a pixel in an unannotated image. This CNN takes the pixels of the patch located at the center of a pixel as an input and outputs the distance estimated for this pixel. Note that this work uses the L channel of the pixels as the input, after converting the given RGB image to its La*b* equivalent.

For learning, it makes use of the 16-layer network architecture proposed by the VGG team in the ILSVRC-2014 competition [88]. The original VGG-16 network was trained on the ImageNet dataset, where the inputs are 224×224 images and the output is one of the 1000 classes. In particular, this step uses the convolutional, ReLu, and pooling layers of the VGG-16 network as they are and makes the following changes in its input, output, and fully connected layers. The first change is in the input size. Our model calculates the average cell size over the training images and uses this average as the patch (input) size. The size of a cell is calculated as the average of the width and the height of its bounding box. The motivation behind selecting the input size on the training images of a given cell type, instead of using a hard-coded one, is the following. One goal of this study is to build a cell detection algorithm that can work on different cell types without redesigning the model, which includes redesigning the CNN architecture and the cell detection steps. Thus, we want the CNN classifier to be less susceptible to variations in image resolution and magnification. By using this automatic input size selection, the same network architecture can easily be used for different cell types, of course after fine-tuning its weights on the given training set. The second change is in the output size and type. Since the estimated distance is a scalar continuous value, the problem is to regress a single output. Thus, the model reduces the number of units in the output layer to one and does not use the softmax function. Due to this reduction, the number of hidden units in the fully connected layer is reduced from 4096 to 500 (Figure 5.4).

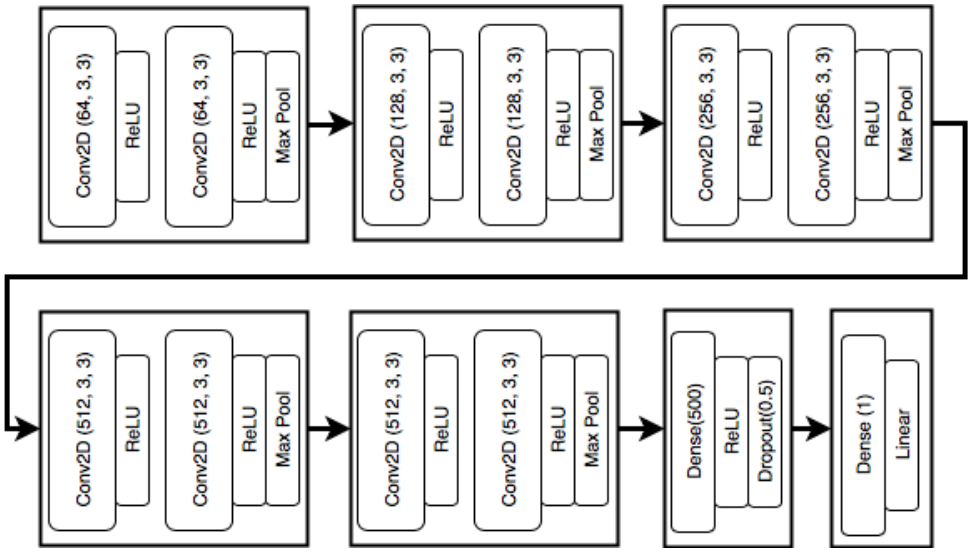
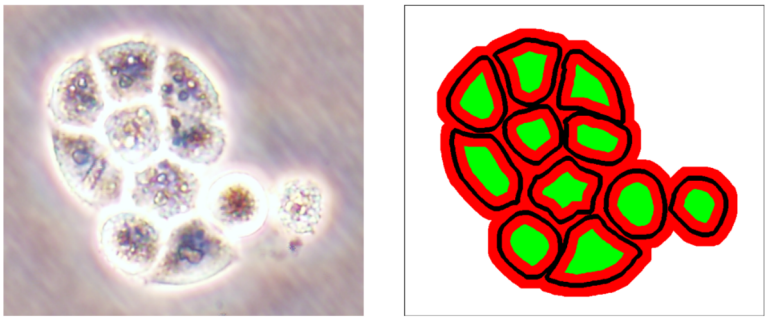


Figure 5.4: Updated version of VGG-16 network used in this study.

To learn the weights of this CNN classifier, a training set of patches is formed by cropping small windows around the selected pixels of the training images. Note that here the pixel selection is not purely random to prevent any bias that would occur upon algorithm's over-fitting to any specific part of an image during its training. Instead, each training image is divided into three parts using its annotation and then pixels are uniformly selected from each part. These parts correspond to the regions close to cell centers, regions close to boundaries, and image background. A training image is divided into three by using the morphological erosion and dilation operators (Figure 5.5). For that, annotated cell pixels are first eroded with a disk structuring element of 51×51 and the remaining annotated pixels are considered as "center". Then, the original annotated cell pixels are this time dilated with the same structuring element and the background pixels of the dilated image are considered as "background". The other pixels are considered as "boundary". Once again note that these center, boundary, and background parts are just used to form the training sets for regression where the inputs are patches and the outputs are continuous distance values.



(a) (b)

Figure 5.5: (a) Original subimage and (b) its division into three parts for forming a training set of patches. In (b), "center", "boundary", and "background" pixels are shown with green, red, and white respectively. Here black lines are used to show boundary annotations. These black lines do not represent an additional part but are drawn just for the illustration purpose. Note that they are considered as "boundary" pixels when forming the set of training patches.

5.1.3 Distance Estimation for Cell Detection

The last step estimates a distance map for the pixels of an unannotated image and identifies cell locations on this estimated map. For a pixel, the patch cropped around it is fed to the trained CNN as the input and the CNN's output is used as the estimated distance. In this map, pixels with higher estimated values are expected to belong to cellular regions far from a boundary, and thus, to be part of a cell center. Thus, the proposed *DeepDistance* model finds regional maxima on the distance map and then identifies the regions larger than the area threshold t_{area} as the cell centers. In order to suppress possible noise in the estimated distance map, the model applies the h-maxima transform beforehand and suppresses the maxima whose height is less than the value of h.

In this step, the distance estimation may require extensive computation for images with high resolution. In order to reduce the computational time for such images, one may consider to estimate the distance every k pixels using the trained

CNN and calculate the approximate distance for the other pixels. This approximation yields almost k^2 fold speed-up in the computational time. In this work, for a pixel for which the CNN is not directly called, we use the estimated distance of its closest pixel. Considering the resolution of the images, we select $k = 10$, which yields a considerable amount of speed-up and still produces accurate detection results.

5.2 Experiments

5.2.1 Datasets

We test our *DeepDistance* model on six different datasets, each of which consists of live cell images of a different cell line. The first three are FOCUS, Huh7, and SkHep1 human liver cancer cell lines and the other three are MDA-MB-453, MDA-MB-468, CAMA-1 human breast cancer cell lines. The images in all these datasets were acquired at $20\times$ magnification and 3096×4140 pixel resolution. An example from each dataset is shown in Figure 5.6. As seen in this figure, visual characteristics may change from one cell type to another. Additionally, in the first three cell lines, cells have irregular shapes and heterogeneous intensities. Small low-intensity dots exist as a part of cells' nuclei but there also exist similar small dark subregions especially towards cells' boundaries. For such cells, it would be quite challenging to define handcrafted intensity, gradient, or shape-based features. In the last three cell lines, cells' shapes are less irregular. However, this time, they appear in different looks. For instance, in Figure 5.6d, there are mostly near-circular cells, which sometimes contain mostly bright pixels but sometimes contain dark pixels inside and bright ones outside. On the other hand, in Figures 5.6e and 5.6f, there are near-circular as well as non-circular cells. For such images, it would not be easy to use a single model to detect cells of all these different looks.

For each cell line, three images are randomly selected and their cells are used

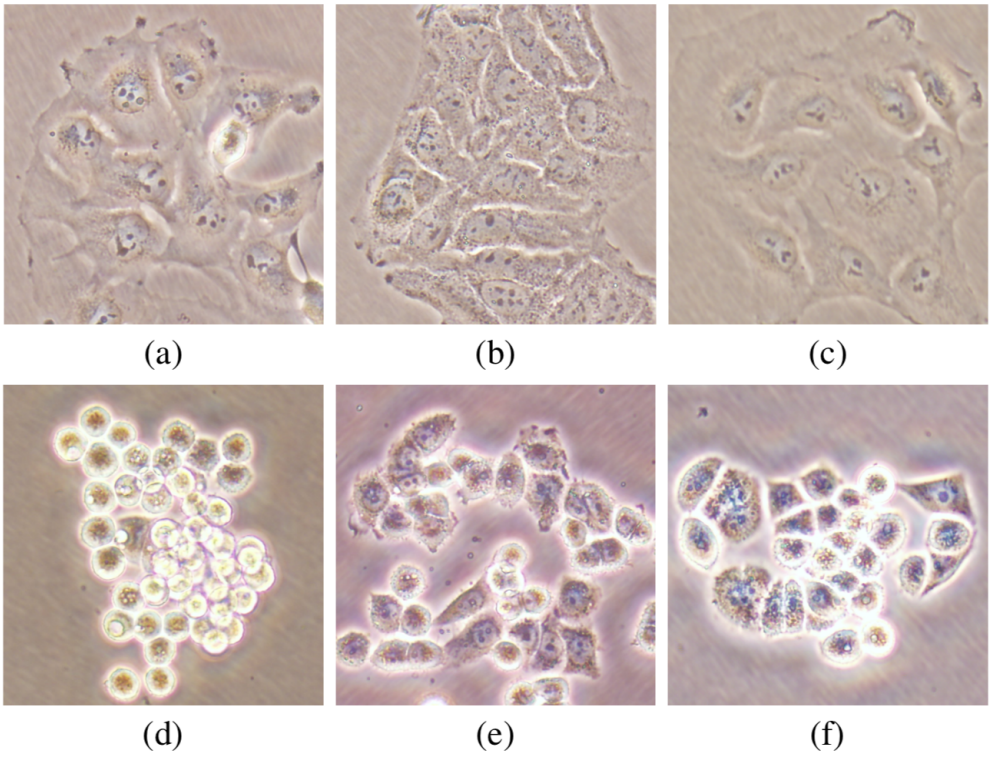


Figure 5.6: Example subimages from human liver and breast cancer cell lines used in our experiments: (a) FOCUS, (b) Huh7, (c) SkHep1, (d) MDA-MB-453, (e) MDA-MB-468, and (f) CAMA-1 cell lines.

for training. These training cells are used to learn the CNN regressors as well as to select the parameters of the cell detection step. The cells in the rest of the images are used for testing and none of them are used in any of the training steps including the parameter selection. For each cell line, the number of images and the number of cells in the training and test sets are presented in Table 5.1. The cells are annotated by putting markers to their approximate centers. These markers are used in the cell-level evaluation of our model as well as the comparison methods (Section 5.2.2). In addition to these markers, the training cells are also annotated by delineating their precise boundaries since the definition of the distance to be learned by the CNN requires knowing the cell boundaries.

Table 5.1: For each cell line, the number of images and the number of cells in its training and test sets.

	Training set		Test set	
	Image no	Cell no	Image no	Cell no
Liver cancer cell lines				
FOCUS	3	205	6	336
Huh7	3	438	5	948
SkHep1	3	309	5	596
Breast cancer cell lines				
MDA-MB-453	3	817	4	1254
MDA-MB-468	3	615	5	747
CAMA-1	3	786	6	866
Total	18	3170	31	4747

5.2.2 Evaluation

Each method is quantitatively evaluated on the test cells regarding the F-score metric. This metric is calculated at the cell-level, considering the number of one-to-one matches between the annotated markers and the detected cells (detected regional maxima). For that, each annotated marker is matched to every detected cell if the distance between this marker and the centroid of the detected cell is less than a distance threshold. Similarly, each detected cell is matched to every annotated marker if their distance is less than the same threshold. Afterwards, a detected cell C is considered as one-to-one match (true positive), if it matches with only one marker and this marker matches with only the cell C. Considering the resolution of the images and the cell sizes, this threshold is selected as 50. Then, the precision and recall metrics are obtained using on these one-to-one matches, and the F-score is calculated as the harmonic mean of these two metrics.

5.2.3 Parameter Selection

The *DeepDistance* model has two external parameters: The first one is the h value, which is used by the h-maxima transform to suppress possible noise in the estimated distance map. The second parameter is the area threshold t_{area},

Table 5.2: Selected parameter values for each cell line.

	h	t_{area}
FOCUS	0.5	750
Huh7	0.5	250
SkHep1	0.5	1000
MDA-MB-453	0.3	250
MDA-MB-468	0.4	750
CAMA-1	0.4	500

which is used to eliminate smaller maxima from the results. The values of these parameters are selected on the training sets. For that, we consider every combination of the values in the following sets and select the one that yields the highest F-score metric for the training cells. These sets are $h = \{0.1, 0.2, 0.3, ..., 1.0\}$ and $t_{area} = \{100, 250, 500, 750, 1000, 1500, 2000\}$. Here the parameters are selected for each cell line separately, by using the cells in the training set of this cell line, since different cell lines may show different intensity and morphological characteristics. For each cell line, the selected parameter values are given in Table 5.2. Note that the parameter values are selected similarly for the comparison methods.

5.2.4 Results

The quantitative test set results obtained by the proposed *DeepDistance* model are reported in the second column of Table 5.3. This table shows that our model leads to accurate detection results for different cell lines. As mentioned before, cells of the selected liver cancer cell lines have more irregular shapes and more heterogeneous intensities, which make their detection harder. This is also observed in the table, where our model gives more accurate results for cells of the selected breast cancer cell lines. Additionally, the second rows of Figures 5.7 and 5.8 present the visual results obtained on example subimages taken from different cell lines. Both the quantitative and visual results of our model indicate the effectiveness of using deep learning for cell detection.

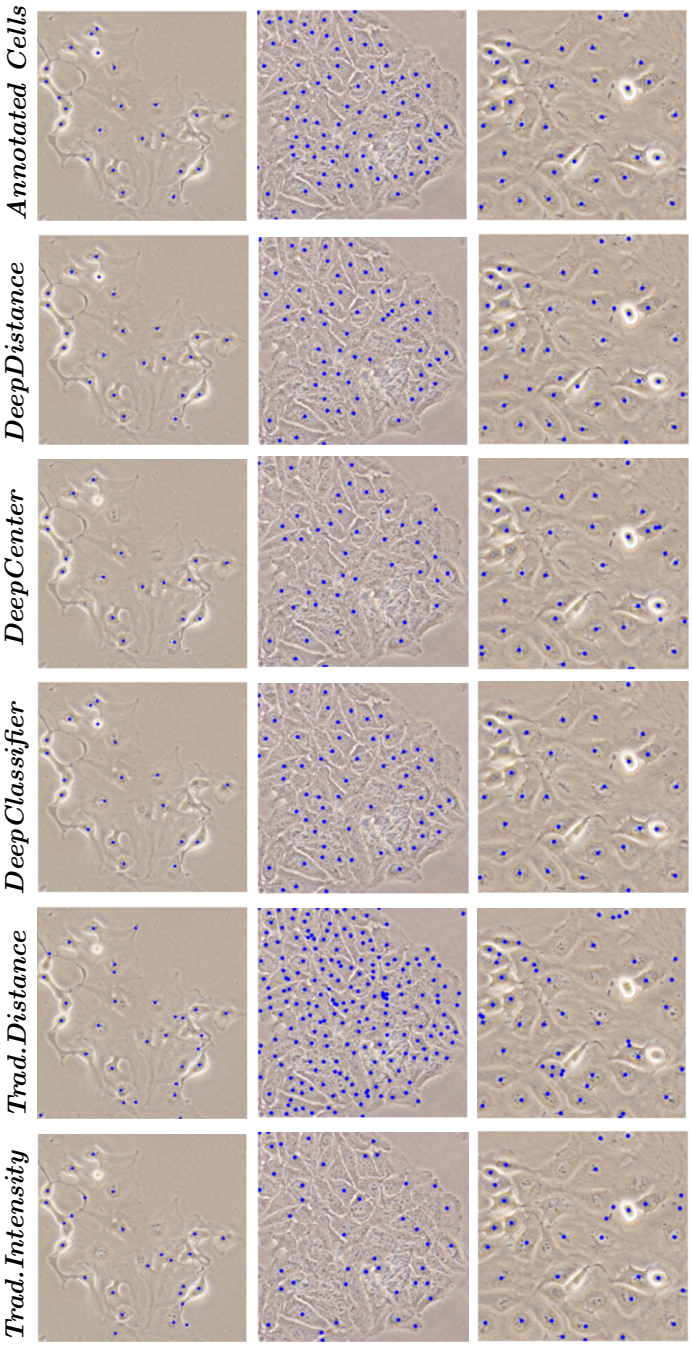


Figure 5.7: Visual results for various subimages taken from the test sets of each liver cancer cell line: (a) FOCUS, (b) Huh7, and (c) SkHep1. For each method, detected cells are indicated with blue dot markers. Note that the sizes of the subimages are scaled for better visualization.

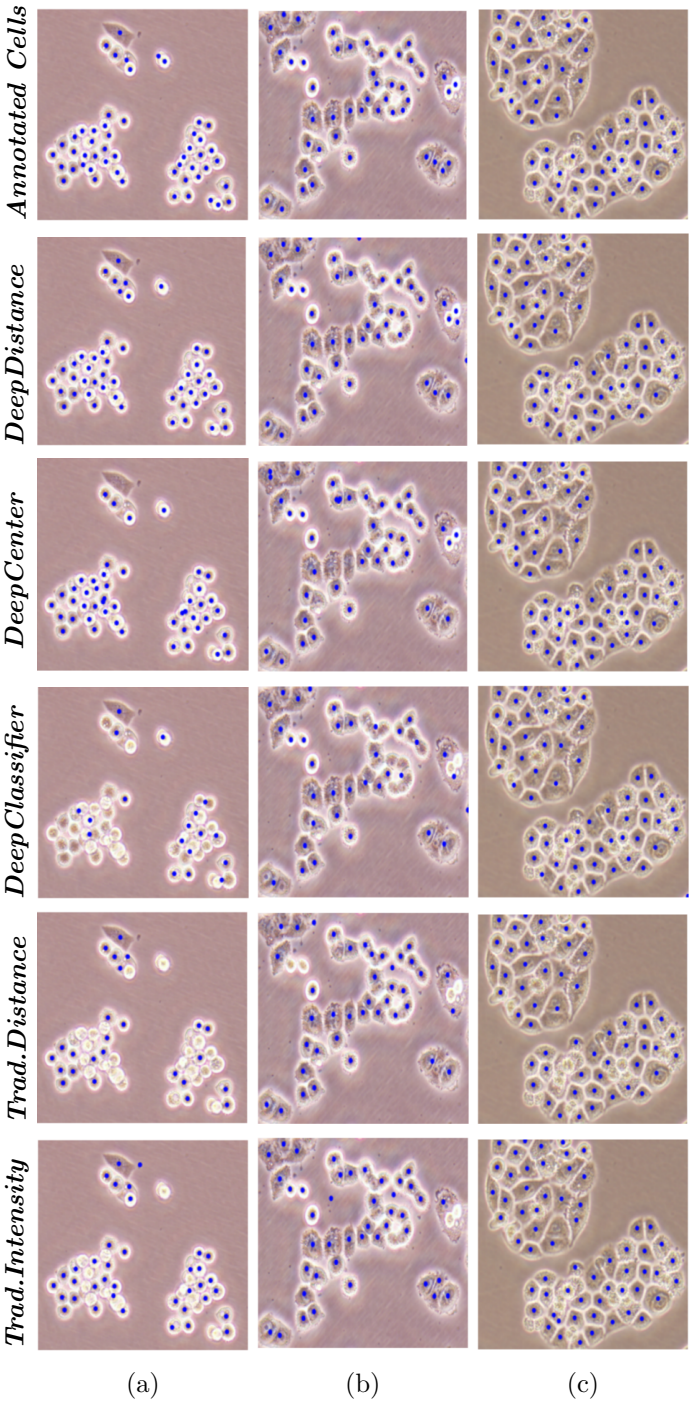


Figure 5.8: Visual results for various subimages taken from the test sets of each breast cancer cell line: (a) MDA-MB-453, (b) MDA-MB-468, and (c) CAMA-1. For each method, detected cells are indicated with blue dot markers. Note that the sizes of the subimages are scaled for better visualization.

Table 5.3: F-score metrics obtained on the test sets by the proposed *DeepDistance* model and the comparison methods.

	Deep Distance	*Deep Center*	*Deep Classifier*	*Traditional Distance*	*Traditional Intensity*
FOCUS	82.82	77.22	84.03	52.56	48.92
Huh7	78.76	54.53	76.38	39.10	28.32
SkHep1	86.54	77.75	84.11	69.88	59.59
MDA-MB-453	87.32	83.38	50.75	68.11	76.29
MDA-MB-468	89.73	83.77	81.18	77.97	79.05
CAMA-1	92.95	89.62	87.58	85.03	88.10
Overall	87.27	78.85	78.62	67.28	67.15

The *DeepDistance* model relies on estimating a distance map for image pixels by the trained CNN and identifying regional maxima on this map for cell detection. For this purpose, it proposes to use the distance definition given in Equation 5.1. In order to understand the effectiveness of this distance definition and estimation, we compare our model with different distance calculation/estimation approaches. The first approach is the *DeepCenter* method that includes exactly the same steps of our model except the distance definition. In particular, instead of using the definition given in Equation 5.1, the *DeepCenter* method defines the distance $\bar{d}_q$ for the pixel q as in Equation 5.2, where ϵ_q is the Euclidean distance from q to its closest annotated *cell center* and α is the decay ratio, which needs to be empirically selected.

$$\bar{d}_q = 1/(1 + \alpha\ \epsilon_q) \tag{5.2}$$

Note that this definition is similar to the one employed in [77]. We use this comparison method to investigate the effects of using boundary annotations, instead of cell center annotations, in distance definition. As aforementioned, the use of annotated cell centers results in imposing a circular shape on cells, which may not always be true for all cell types. Except this difference of distance definition, the *DeepCenter* method uses the CNN of the same architecture, trains this CNN on its selected patches in a similar way, and takes the same cell detection

steps of our model. These steps include applying the h-maxima transform, finding regional maxima, and eliminating smaller maxima. The test set results given in Table 5.3 and the third rows of Figures 5.7 and 5.8 show that the distance definition used in our model significantly improves the success of cell detection.

Next, we explore the use of a deep classification method that generates a posterior map on which regional maxima are found to detect the cell locations. Note that this is the most common approach followed by the previously reported deep learning based cell detection studies. For this, we implement this second comparison approach, which we call the *DeepClassifier* method. Instead of regressing a distance output, this method poses the problem as binary classification where it trains the CNN to classify image pixels as cell (foreground) or background. Afterwards, for an unannotated image, the *DeepClassifier* method obtains a map containing the cell class' posteriors of image pixels, suppresses noise on this posterior map applying the h-maxima transform, and identifies regional maxima larger than an area threshold as cell locations. The test set results obtained by this approach are given in Table 5.3 and the forth rows of Figures 5.7 and 5.8. Here it can be observed that except for the FOCUS cell line dataset, our proposed model leads to more accurate detections. Only for this dataset, the *DeepClassifier* method gives slightly better results. This may be attributed to the following. As this method estimates the cell class' posteriors of pixels regardless of their relative positions within the cell, it may produce similar posteriors for foreground pixels close to a cell center or close to a cell boundary. However, foreground pixels close to a cell boundary may have similar patches with background pixels that are also close to a cell boundary, especially when cells have overlaps and/or when there do not exist sufficient visually distinctive boundaries in between the overlapping cells. As a result, this may reduce the method's performance. This is also consistent with our experimental findings. We observe that the *DeepClassifier* method gives good results when cells are relatively far from each other and have less overlaps (e.g., FOCUS cell line dataset) and when there exist visually distinctive boundaries in between the nearby cells (e.g., CAMA-1 cell line dataset). On the other hand, its performance decreases when none of them are observed (e.g., MDA-MB-453 cell line dataset). On the other hand, the proposed

DeepDistance model leads to high accuracies for all these datasets. These results together with those of the *DeepCenter* method indicate the effectiveness of our distance definition as well as of its estimation constructing a deep CNN regressor.

The third comparison approach is the *TraditionalDistance* method. It obtains a binary image based on pixel intensities and gradients and calculates the distance transform on this binary image. This transform assigns a distance to every pixel, which is the Euclidean distance between the corresponding pixel and the nearest zero pixel in the binary image. After obtaining this distance map, the *TraditionalDistance* method takes the same steps of our model. That is, it applies the h-maxima transform, finds regional maxima, and eliminates smaller maxima. We use this method in our comparisons to understand the difference between defining a distance metric manually (by using a standard distance transform operation) and learning it directly on image data (by a deep learning approach). The quantitative test set results given in Table 5.3 show that the latter one significantly improves the detection success. From this table as well the visual results given in the fifth rows of Figures 5.7 and Figure 5.8, it can be seen that the *TraditionalDistance* method may give relatively good results for some datasets (e.g., for the CAMA-1 cell line). Indeed, the success of this method very closely relies on the accurateness of the binary image, on which the distance transform is calculated. In our experiments, we obtain a binary image using the algorithm previously developed in our research group [26] but take additional postprocessing steps to refine this binary image. The algorithm as well as the postprocessing steps make use of pixel intensities and gradients. However, here there are two main problems. First, the same set of operations may not give binarization results with the same accuracy level for entire parts of the image, especially when there exists heterogeneity in pixel intensities/gradients. Second, in our experiments, we need to manually select which additional postprocessing steps are to be used with respect to the given dataset (with respect to the given cell line type's characteristics). When compared this manual selection, our model has the benefit of using a deep learning approach, which learns distance estimation directly on image data. Such an approach makes it easier to use the same detection model, with the same set of computational steps, for different cell types, as also demonstrated by our

experiments.

The last comparison approach is the *TraditionalIntensity* method, which directly makes use of image intensities instead of applying a distance transform. Likewise, this method takes the same cell detection steps of our model. However, since the majority of cells have dark pixels inside, we apply the h-maxima transform on the inverse intensity map and find regional maxima on its result. Similar to the *TraditionalDistance* method, this one also yields inaccurate results when cells have heterogeneous pixels (e.g., the first three cell lines). Additionally, it fails to detect all cells in the same image when the cells appear in different looks. For example, in the MDA-MB-453 cell line, there may also exist cells that contain bright pixels inside. Of course, finding regional maxima on the inverse intensity map does not work for such cells. However, if the model was modified to detect these cells, this time, it would fail to detect the cells containing dark pixels inside. In this case, either one of these options should be selected at the expense of failing for the other cell type or a more complex model should be designed. However, as the cell characteristics may change from one cell type to another, this complex model design may still need to be modified for another cell type. On the other hand, it is less likely for our proposed model to make such modifications for different cell types, provided that a couple of annotated training images are given. This indeed demonstrates the strength of deep learning.

5.3 Improvement with U-Net

DeepDistance is a patch-based approach which takes a rectangular region surrounding pixel to be processed and then computes regressed value for that pixel. However, there are excessively redundant computations required because of the overlapping regions. Recently, end-to-end methods, such as fully convolutional networks (FCNs) and U-nets have become popular, which take a patch and then compute output value for each pixel in the patch at the same time. Since it calculates output value of all pixels in a patch at the same time, it boosts the efficiency

of the overall method dramatically. There are studies utilizing U-Net for cell segmentation problem [79, 80, 84]. The method explained in [79] makes use of a pre-trained off-the-shelf model, called *DeepLab* [85]. The methods given in [80, 84] modify FCN and U-net architectures by adding/dropping some network layers, by changing number of features, and by adding residual connections/blocks. On the other hand, they all consider the cell segmentation problem as a single-task problem where the image is the input and there only exists one output map to be learned, making it difficult to avoid local optima in the training process and to have a generalizable model. In response to these issues, we introduce a new cascaded multi-task learning framework instead of using a CNN for detection of cells. In this framework, two distance maps are learned end-to-end at the same time, sharing high-level feature representations at various layers (layers of the encoder path) in the context of multi-task learning. This multi-task network is cascaded to a segmentation network. In the literature, there exists a study that use cascaded network architecture for cell detection [89]. However as opposed to our framework, it is a single-task framework in which density regression is learned on the output mask of the network. There also exists a study [90] that learns two different maps and combined them for the final segmentation, in which such shared features are used to find the glands and nuclei in histopathology images. Not only this task is completely different than ours where our aim is to detect cells in microscopy images, but also, they neither used a distance (regression) approach nor a cascaded architecture. The output of this mentioned work is a combination of the tasks learned with a simple fusion function.

The contributions of our proposed framework are threefold: (1) As opposed to the previous studies that consider this problem as a single-task problem, we take advantage of the multi-task learning approach which is known to be successful for many domains, leveraging the contribution of different tasks to the feature representation learning process [91]. Learning complementary tasks with shared representations may increase the generalization ability and the performance of the method by avoiding false local optima. In our framework, the first task to be learned is the *inner distance*, which is calculated with respect to the cell centers,

similar to the ones used in the previous studies (Equation 5.2). This distance imposes a circular shape on the cells when it is used alone. However, as mentioned in the previous sections, there may exist different cells with different characteristics. (2) To address this issue, we propose to use the normalized *inner distance* metric, given in Equation 5.2[2], together with the newly proposed normalized outer distance metric, given in Equation 5.1. We claim that using these metrics together to learn shared feature representations improves performance for each task. (3) This multi-task network is cascaded to a segmentation network which constructs the *binary* mask from the two output regression maps. The motivation behind this cascade strategy is the simple fact that the distance outputs should contain all the information necessary to reconstruct the segmentation with minimum error. The segmentation network can be thought as a supervision mechanism which penalizes the short-comings of the regression outputs, such as deficient and/or perforated cell shapes and multiple local optima in one cell, by back-propagating the error through the entire framework in the end-to-end training process. Then, with these three contributions, for a given image, cell detection is achieved by generating *inner distance* outputs with the trained model and finding regional maxima of the generated map.

Our cascaded multi-task network includes three subnetworks, each of which has the same architecture (Figure 5.9). The first two subnetworks learn *outer distance* and *inner distance* from a grayscale image, simultaneously. Afterwards, outputs of the networks are conveyed to the last subnetwork. The last subnetwork learns the *binary* map from these two distance maps. Errors are back-propagated through the entire network to improve performance of all subnetworks. The entire network is depicted in Figure 5.10. Finally, regional maxima on the *inner* distance map are selected as the cell locations after we apply h-maxima transform to eliminate noise.

[2] α is the decay ratio which controls sharpness of the peaks on the cell locations. To keep the sharpness very small, we select α as 0.1.

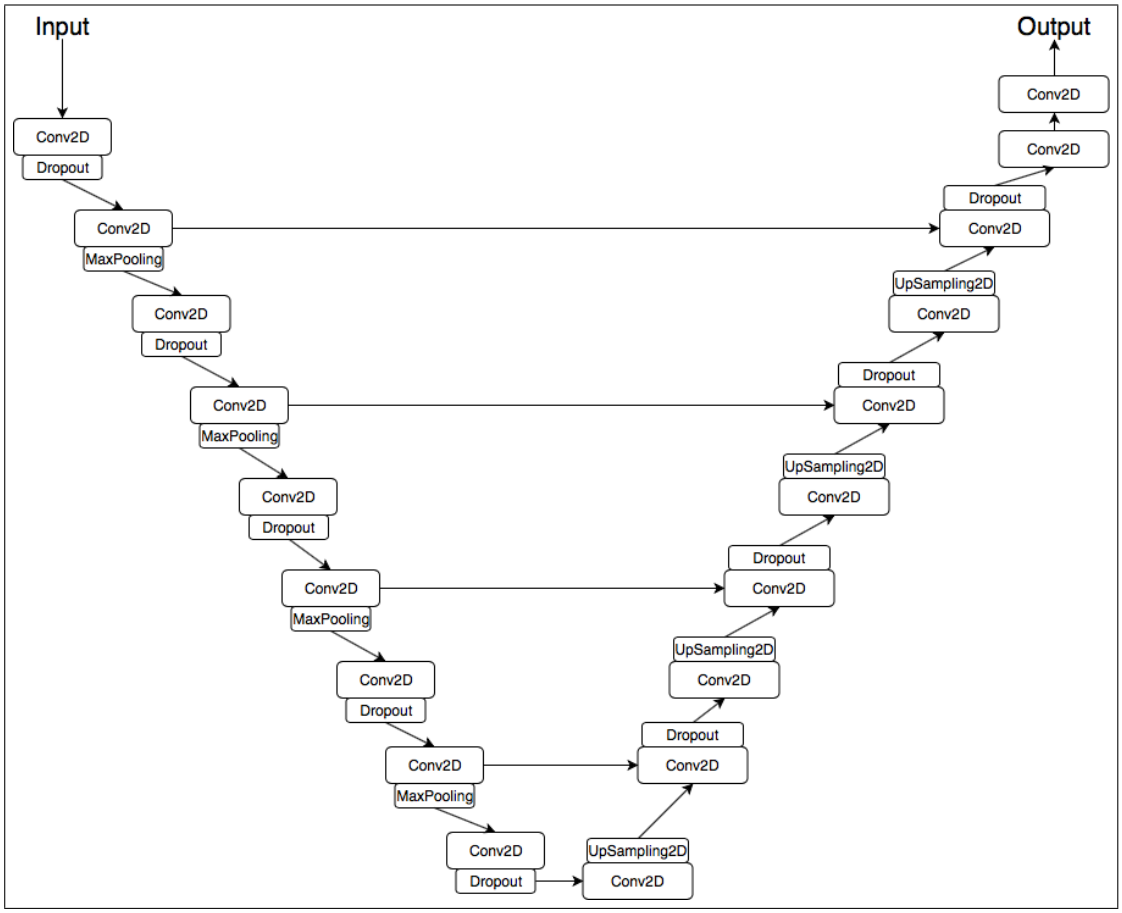


Figure 5.9: The U-Net structure utilized for the cascaded multi-task *DeepDistance*.

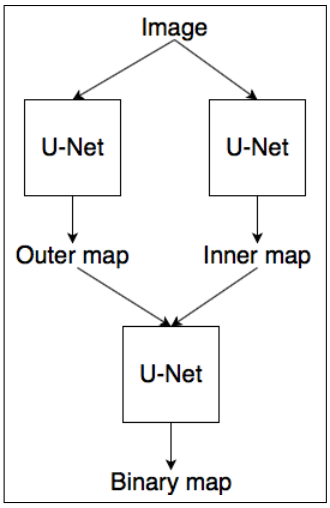


Figure 5.10: The overall pipeline. The pipeline takes a 2D image and learns three maps, namely *outer*, *inner*, and *binary* maps. The inner distance map is used for cell detection.

Table 5.4: Selected h value for each cell line.

	h
FOCUS	0.3
Huh7	0.3
SkHep1	0.3
MDA-MB-453	0.2
MDA-MB-468	0.2
CAMA-1	0.2

5.3.1 Experiments

The proposed model is evaluated on the same test sets and compared with the same algorithms explained in Section 5.2.4 in order to observe if we achieve any improvements on the proposed *DeepDistance*. To evaluate generalizability of the cascaded multi-task *DeepDistance* model, we exclude two cell line data sets (one from the liver cell lines, namely SkHep1, and one from the breast cell lines, namely MDA-MB-468) from the training step, merge the training images of the remaining cell lines (namely FOCUS, Huh7, MDA-MB-453, and CAMA-1) and train the model on the merged set. Here, we do not include any images of the SkHep1 and MDA-MB-468 cell lines in any steps of training. The aim of excluding two sets is to see how the proposed framework acts on previously unseen cell line types.

The *inner* distance map is used for cell detection purpose since it represents cell locations better. Before detecting cell locations, we eliminate undesired maxima by using h-maxima transform. However, even though the cascaded *DeepDistance* is one and common for all cell line types, the same h value in h-maxima transform is not sufficient for all cell line types because of variabilities in size and shape of the cells. Hard selection of h value in conventional h-maxima transform makes it limited to various cell types. Because of this reason, we select an h value for each cell line among values of the set $\{0.1, 0.2, 0.3, ..., 1.0\}$. The h value is selected for each cell line separately, using its own training images (Table 5.4). Here, we do not remove small regions because all regional maxima on the *inner* map have very small area, unnecessitating use of such elimination step.

Table 5.5: F-score metrics obtained on the test sets by the cascaded multi-task *DeepDistance* model and the comparison methods. Note that the proposed cascaded *DeepDistance* does not use any images of SkHep1 and MDA-MB-468 cell lines in the training step.

	Deep Distance	*Deep Center*	*Deep Classifier*	*Trad. Distance*	*Trad. Intensity*	*Cascaded Multi-task. DeepDist.*
FOCUS	82.82	77.22	84.03	52.56	48.92	88.58
Huh7	78.76	54.53	76.38	39.10	28.32	84.95
MDA-MB-453	87.32	83.38	50.75	68.11	76.29	85.20
CAMA-1	92.95	89.62	87.58	85.03	88.10	91.79
SkHep1	86.54	77.75	84.11	69.88	59.59	92.99
MDA-MB-468	89.73	83.77	81.18	77.97	79.05	86.80

We believe that our cascaded multi-task network can learn high-level features of cells in live microscopy images even though they appear in various shapes and intensities. Our results, even for the cell lines which were not included in training, reveal that the cascaded multi-task end-to-end model successfully identifies cell locations better, improving the results of the *DeepDistance* model. The cascaded multi-task *DeepDistance* has several advantages over the initial *DeepDistance*: (1) Instead of building one model for each cell line, which needs extra effort when a new data set comes in, we built one model which is common for all cell types, requiring no effort even for new cell lines. (2) Improved version of *DeepDistance* outperforms all comparison methods in terms of F-score values (Table 5.5). Besides, it outperforms the initial version of *DeepDistance* on datasets of all liver cell lines. For the breast cancer cell lines, F-score values of the cascaded multi-task *DeepDistance* model are slightly lower than the ones of the original *DeepDistance*. The reason might be that the initial approach trains a new model for each cell line. It is likely to learn features specific to one cell line, improving detection performance of the model. However, even though the model can detect cells of a specific cell type successfully, it cannot be as generalizable as the cascaded multi-task *DeepDistance*. (3) Since it does not have excessive redundant calculations as in patch-based approaches, which is also used by the initial *DeepDistance*, cell detection is performed more efficiently.

5.3.2 Further Improvement with Iterative H-maxima Transform

Cell detection is performed using the conventional h-maxima transform on the *inner distance* map. However, the conventional h-maxima method uses only one h value for the entire image, imposing cells on the image on having similar size and shape. Since the h-maxima disregards the fact that cells, especially living cells, have different morphological properties, it is likely to fail on cell detection. Besides, we need to define an h value for each cell line, which is contrary to the idea of our deep learning model. It limits us to build one common model working for all cell types. To make a further improvement in this study and to make the model generic for all cell types, we replace the conventional h-maxima transform with the iterative one, which constitutes the Chapter 4 of the thesis[3]. Since the iterative h-maxima transform does not require hard-selection of the h value, integrating the iterative h-maxima transform, instead of the conventional one, makes the overall framework independent from parameter fine-tuning and cell types.

The iterative h-maxima transform is performed on the *inner* distance map. However, even for the smallest value of h, which is 1, h-maxima transform would fail since the value would be too large for the *inner distance* map, which is in between $[0, 1]$. To solve this problem, we rescale the map into the range which is similar to the map used in the original iterative h-minima study. To do so, we multiply each and every value of it by a constant, which is 50, before giving it to the iterative h-maxima transform.

As seen in Table 5.6, the cascaded multi-task *DeepDistance* is further improved by integrating the iterative h-maxima transform, instead of the conventional one. We improve the framework so that it works different types of cells without requiring any parameter fine-tuning. Besides, we improve the F-score values for the

[3]The original study explained in Chapter 4 uses h-minima transform since regional minima on input image correspond to cell locations. However, here, since regional maxima on *inner* distance map correspond to cell locations we changed the h-minima transform to h-maxima transform.

Table 5.6: F-score metrics obtained on the test sets by the *DeepDistance* models. Note that the proposed cascaded *DeepDistance* models do not use any images of SkHep1 and MDA-MB-468 cell lines in the training step.

	Deep Distance	*Cascaded Multitask. DeepDist.*	*Cascaded Multitask. Iter. DeepDist.*
FOCUS	82.82	88.58	88.02
Huh7	78.76	84.95	85.02
MDA-MB-453	87.32	85.20	85.30
CAMA-1	92.95	91.79	93.23
SkHep1	86.54	92.99	92.26
MDA-MB-468	89.73	86.80	89.31

CAMA-1 and MDA-MB-468 cell lines and keep the F-score values for the other cell lines approximately same. Note that in this table, the results for the initial *DeepDistance* are those obtained by separately training a model for each cell line type.

5.4 Discussion

This study introduces a generic cell detection model, which we call *DeepDistance*, for microscopic images of live cells. The proposed *DeepDistance* model relies on defining an efficient metric, on which the detection steps take place, and estimating this metric for pixels of an unannotated image by a deep learning method. To this end, it defines its metric as to calculate a normalized distance from a pixel to its closest cell boundary, which is different than the previously reported deep learning based methods that use cell centers in their distance calculations. This definition of the *DeepDistance* model proves more useful in preserving the shape characteristics of the cells since the latter definition imposes a circular shape on the cells. In order to estimate this distance metric for unannotated pixels, our model proposes to train a CNN regressor on patches cropped around training pixels. Afterwards, it achieves cell detection by identifying regional maxima on the estimated distance map, which corresponds to finding a set of connected pixels whose estimated distances are locally maximum.

Our experiments on six different datasets, containing a total of 7917 cells, have revealed that the distance definition and estimation proposed by the *DeepDistance* model lead to better detection results compared to its counterparts. They have also demonstrated that the *DeepDistance* model provides a generic cell detection tool that can successfully be used for different datasets showing different visual characteristics, without making any algorithmic changes in its steps. On the other hand, it requires parameter fine-tuning when a new cell line comes in. We solve this problem by integrating a cascaded multi-task U-Net into *DeepDistance*. The cascaded network is trained once by using the training images of all cell lines together, excluding the SkHep1 and MDA-MB-468 cell lines. We do not need to train the model for each cell line, which contributes to making the tool much more generic. Experimental results show that even with one common model, improved *DeepDistance* performs as successful as the CNN based *DeepDistance*, which builds one model for each cell line. Moreover, since it eliminates redundant calculations for pixels in overlapping regions, performance of the whole framework is boosted. However, the improved *DeepDistance* model still needs a parameter fine-tuning step for each cell line at the step of h-maxima transform. The h value of the transform should be defined separately for each data set. Integrating iterative h-maxima provides us to have a fully independent cell detection framework which is independent from any cell types. The model is also tested on two different cell types not included in any steps of training. Table 5.6 shows that the cascaded multi-task iterative h-maxima based *DeepDistance* method can detect cells accurately independent from cell types as long as morphological properties of cells do no change dramatically from properties of the ones used in training.

This work mainly focuses on the cell detection problem, which corresponds to identifying cell locations in an unannotated image. It does not focus on delineating the precise boundaries of the cells. As a future work, it is possible to implement a seed-controlled region growing algorithm, in which the seeds are found by the *DeepDistance* model and are grown with respect to also the estimated distances. Our improved model has better performance on detecting different types of cells thanks to the cascaded subnetworks, which make performance of

each other better as if they are in a mutualist relationship. Here, regressed *inner* distance map is utilized for cell detection since it represents location of cells better. The *binary* map is another output of the network which illustrates the cellular regions as foreground (white) and others as background (black). Since the binary map represents these region nearly-optimal, cells' boundaries can be delineated accurately by making use of the *inner* distance map and the *binary* map together. To do that, cell locations can be detected on the *inner* distance map and then these locations can be grown on the distance transform of the *binary* map. This can be considered as one of the future research directions. As another future work, one may consider applying this algorithm to microscopic images of histopathology slides in order to detect and segment cells as well as tubular structures in these images.

Chapter 6

Conclusion

This thesis addresses the cell segmentation problem in two parts: (1) It focuses on finding a new representation for microscopy images so that imperfections on the image can be alleviated, and cells can be emphasized better, helping us for simplifying the cell segmentation problem. (2) It focuses on developing a more advanced cell segmentation method. Even though the representation method works well, it is almost impossible to obtain a perfect representation in practical life. There is still a need for developing more sophisticated cell segmentation techniques which overcome deficiencies on the representation. To this end, this thesis introduces three new cell segmentation models, two of which introduce a new cell representation technique as well.

Our first model is object oriented method for segmenting cells of fluorescence microscopy images. The model first represents a fluorescence microscopy image in terms of *edge-objects* and subregions. Then, cells are segmented by merging subregions sharing the same *edge-objects* in different orientations. We believe that utilizing *edge-objects* and subregions in a divide-and-then-merge manner have a great potential of segmenting cells accurately. Here, the reason for using edge information at the object level instead of the pixel level is that *edge-objects* have capability of handling pixel level noises and imperfections.

This high-level representation together with this high-level merging are expected to be more successful at segmentation of cells compared to the methods that directly work on low-level pixel properties. Our experiments on fluorescence microscopy images show that the proposed object oriented method leads to better segmentation results compared to pixel-level cell nucleus segmentation algorithms.

We believe that our object oriented approach has a capability of segmenting 3D cells as well. For that, *edge-objects* and subregions should be defined in 3D. Subregions in 3D can be extracted by the SLIC algorithm which has a support of defining supervoxels, as well. Additionally, there should be *edge-objects* defined in the third axis (depth), namely *front* and *back* object types. Extending the model in 3D cell segmentation can be considered as one of the future research directions.

Our second model is a new cell segmentation algorithm in fluorescence microscopy images. The model makes use of a marker-controlled watershed algorithm in which markers are iteratively identified by employing h-minima transform. The conventional h-minima transform applies the same h value on the entire component or the entire image which disregards an obvious fact; cells have different morphological properties such as size and shape necessitating a different h value for each. More specifically, smaller h values are sufficient for cells with homogenous intensity distribution inside, whereas it is likely to fail on cells having inhomogeneities inside. On the other hand, keeping the h value too large yields suppressing noises together with true markers, which correspond to actual cells. To tackle this problem, we propose defining markers iteratively, using a different h value in each iteration, with the aim of suppressing noise at different levels. The iterative identification of markers allows us to define better markers for nuclei showing different characteristics, cascading to have a better segmentation method.

Our experiments on fluorescence microscopy images demonstrate that this algorithm gives better markers for nuclei of both isolated and confluent cells compared to conventional h-minima transform and other cell segmentation methods.

We also show the applicability of the method to other image types. For that purpose, we conduct experiments on images obtained from tissue sections of mouse liver. The images were taken under fluorescent microscope. The experiments show that the model can be easily extended to other image types obtained under fluorescent microscope. We also conduct experiments on bright-field microscopy images which contains cells with various outlook. The preliminary results reveal that identifying cell locations iteratively outperforms the traditional approach.

As a future direction to this study, extendibility of the model to other microscopy domains such as confocal microscopy, which produces cell images with higher resolution for detailed visualization of interior structure of cells, can be investigated. The application of our model to different image types taken under confocal microscope could be considered as a future work. In this work, we mainly focus on finding better markers. We use a relatively simple region growing algorithm to delineate nucleus boundaries. Yet another future work can be to design better techniques for marker growing. Here one could consider designing iterative methods also in the region growing process. Another possibility is to explore the use of other types of maps, on which the growing takes place.

Our last but not least model presents a representation method which convert image into one where cells are emphasized and look more distinguishable. The model can work for any cell types since it automatically learns high-level features from image data. The new deep model approaches cell detection as a regression problem, where the normalized distance from each pixel to the closest background is learned by training a deep convolutional neural network (CNN). Then, for a given image, cell detection is achieved by generating a normalized distance map of the image pixels with the trained CNN and finding regional maxima of the generated map. The model is further improved replacing the patch-based CNN model with an end-to-end multi-tasking cascaded U-Net model and conventional h-maxima with the iterative h-maxima, given in Chapter 4.

Our experiments on six different cell line types reveal that the proposed model successfully identifies the cell locations, improving the results of both the traditional approach and the previous deep learning based methods. The improved

version of the model utilizes cascade multi-task U-Net model which learns a binary map and two distance maps, namely *inner* distance map and *outer* distance map, at the same time. The *inner* distance map gives better information about cell locations, and the *binary* map differentiates cellular regions from the background quite accurately. One can consider combining these two maps in order to delineate cell boundaries after cells' locations are identified. For that, a marker-controlled watershed algorithm can be employed in which markers are identified on the *inner* distance map and growing is applied on the *binary* map. This can be considered as one of the future research directions.

Bibliography

[1] C.F. Koyuncu, R. Cetin Atalay, and C. Gunduz Demir, "Object oriented segmentation of cell nuclei in fluorescence microscopy images," *Cytom. Part A*, 2018 (Accepted).

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274 - 2281, 2012.

[3] C.F. Koyuncu, E. Akhan, R. Cetin Atalay, and C. Gunduz Demir, "Iterative h-minima based marker-controlled watershed for cell nucleus segmentation," *Cytom. Part A*, vol. 89, no. 4, pp. 338 - 349, Apr. 2016.

[4] X. Chen, X. Zhou, and S.T. Wong, "Automated segmentation, classification, and tracking of cancer cell nuclei in time-lapse microscopy," *IEEE Trans. Biomed. Eng.*, vol. 53, no. 4, pp. 762 - 766, Apr. 2006.

[5] H. Zhou and K. Mao, "Adaptive successive erosion-based cell image segmentation for p53 immuno-histochemistry in bladder inverted papilloma," in *Proceed. 27th Annual International Conference Eng. Medicine and Biology Society*, pp. 6484 - 6487, 2005.

[6] N. Harder, F. Mora-Bermudez, W.J. Godinez, J. Ellenberg, and R. Eils, "Automated analysis of the mitotic phases of human cells in 3D fluorescence microscopy image sequences," *Proceed. 9th international Conference Med. Image Comput. Computer-Assisted Interven.*, vol. Part I, pp. 840 - 848, 2006.

[7] N. Kharma, H. Moghnieh, J. Yao, Y.P. Guo, A. Abu-Baker, J. Laganiere, G. Rouleau, and M. Cheriet, "Automatic segmentation of cells from microscopic

imagery using ellipse detection," in *IET Imag. Process*, vol. 1, pp. 39 - 47, 2007.

[8] G. Kayser, D. Radziszowski, P. Bzdyl, R. Sommer, and K. Kayser, "Theory and implementation of an electronic, automated measurement system for images obtained from immunohistochemically stained slides," *Anal. Quant. Cytol. Histol.*, vol. 28, pp. 27 - 38, 2006.

[9] K. Kayser, H. Schultz, T. Goldmann, J. Grtler, and G. Kayser, "Theory of sampling and its application in tissue based diagnosis," *Diagn. Pathol.*, vol. 4, pp. 1 - 13, 2009.

[10] A.A. Dima, J.T. Elliott, J.J. Filliben, M. Halter, A. Peskin, J. Bernal, M. Kociolek, M.C. Brady, H.C. Tang, and A.L. Plant, "Comparison of segmentation algorithms for fluorescence microscopy images of cells," *Cytom. Part A*, vol. 79A, no. 7, pp. 545 - 559, Jul. 2011.

[11] D. Fenistein, B. Lenseigne, T. Christophe, P. Brodin, and A. Genovesio, "A fast, fully automated cell segmentation algorithm for high-throughput and high-content screening," *Cytom. Part A*, vol. 73, pp. 958 - 964, 2008.

[12] F. Yang and T. Jiang, "Cell image segmentation with kernel-based dynamic clustering and an ellipsoidal cell shape model," *J Biomed. Inform.*, vol. 34, pp. 67 - 73, 2001.

[13] M. Park, J.S. Jin, M. Xu, W.S.F. Wong, and S. Luo, "Microscopic image segmentation based on color pixels classification," *Proceed. 1st International Conference Internet Multim. Comput. Service*, pp. 53 - 59, 2009.

[14] M.N. Gurcan, L.E. Boucheron, A. Can, A. Madabushi, N.M. Rajpoot, and B. Yener, "Histopatological image analysis: a review," *IEEE Rev. Biomed. Eng.*, vol. 2, pp. 147 - 171, 2009.

[15] H. Irshad, A. Veillard, L. Roux, and D. Racoceanu, "Methods for nuclei detection, segmentation, and classification in digital histopathology: a review: current status and future potential," *IEEE Rev. Biomed. Eng.*, vol. 7, pp. 97 - 114, 2014.

[16] A.M. Khan, N. Rajpoot, D. Treanor, and D. Magee, "A nonlinear mapping approach to stain normalization in digital histopathology images using image-specific color deconvolution," *IEEE Trans. Biomed. Eng.*, vol. 61, pp. 1729 - 1738, 2014.

[17] C. Jung, C. Kim, S.W. Chae, and S. Oh, "Unsupervised segmentation of overlapped nuclei using Bayesian classification," *IEEE Trans. Biomed. Eng.*, vol. 57, pp. 2825 - 2832, 2010.

[18] M. Plissiti and C. Nikou, "Overlapping cell nuclei segmentation using a spatially adaptive active physical model," *IEEE Trans. Imag. Process*, vol. 21, pp. 4568 - 4580, 2012.

[19] X. Yang, H. Li, and X. Zhou, "Nuclei segmentation using marker-controlled watershed tracking using mean-shift and Kalman filter in time-lapse microscopy," *IEEE Trans. Circuits Syst. I Reg. Papers*, vol. 53, no. 11, pp. 2405 - 2414, Nov. 2006.

[20] C. Park, J.Z. Huang, J.X. Ji, and Y. Ding, "Segmentation, inference and classification of partially overlapping nanoparticles," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 669 - 681, 2013.

[21] X. Zhou, F. Li, J. Yan, and S.T. Wong, "A novel cell segmentation method and cell phase identification using Markov model," *IEEE Trans. Inf. Technol. Biomed.*, vol. 13, pp. 152 - 157, 2009.

[22] J. Cheng and J.C. Rajapakse, "Segmentation of clustered nuclei with shape markers and marking function," *IEEE Trans. Biomed. Eng.*, vol. 56, pp. 741 - 748, Mar. 2009.

[23] C. Jung and C. Kim, "Segmenting clustered nuclei using h-minima transform-based marker extraction and contour parameterization," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 10, pp. 2600 - 2604, Oct. 2010.

[24] H. Yang and N. Ahuja, "Automatic segmentation of granular objects in images: Combining local density clustering and gradient-barrier watershed," *Pattern Recognit.*, vol. 47, no. 6, pp. 2266 - 2279, 2014.

[25] M.R. Jeong, B.C. Ko, and J.Y. Nam, "Overlapping nuclei segmentation based on Bayesian networks and stepwise merging strategy," *J. Microsc.*, vol. 235, no. 2, pp. 188 - 198, 2009.

[26] C.F. Koyuncu, S. Arslan, I. Durmaz, R. Cetin-Atalay, and C. Gunduz Demir, "Smart markers for watershed-based cell segmentation," *PLoS ONE*, vol. 7, no. 11, pp. e48664, Nov. 2012.

[27] S. Arslan, E. Ozyurek, and C. Gunduz-Demir, "A color and shape based algorithm for segmentation of white blood cells in peripheral blood and bone marrow images," *Cytom. Part A*, vol. 85A, pp.480 - 490, 2014.

[28] C. Wahlby, I.M. Sintorn, F. Erlandsson, G. Borgefors, and E. Bengtsson, "Combining intensity, edge and shape information for 2d and 3d segmentation of cell nuclei in tissue sections," *J. Microsc.*, vol. 215, pp. 67 - 76, 2004.

[29] S. Kumar, S.H. Ong, S. Ranganath, T.C. Ong, and F.T. Chew, "A rule-based approach for robust clump splitting," *Pattern Recognit.*, vol.39, pp. 1088-1098, 2006.

[30] M. Farhan, O. Yli-Harja, and A. Niemisto, "A novel method for splitting clumps of convex objects incorporating image intensity and using rectangular window-based concavity point-pair search," *Pattern Recognit.*, vol. 46, pp. 741 - 751, 2013.

[31] H. Chang, J. Han, A. Borowsky, L. Loss, J.W. Gray, P.T. Spellman, and B. Parvin, "Invariant delineation of nuclear architecture in glioblastoma multiforme for clinical and molecular association," *IEEE Trans. Med. Imaging*, vol. 32, no. 4, pp. 670 - 682, Apr. 2013.

[32] X. Bai, P. Wang, C. Sun, Y. Zhang, F. Zhou, and C. Meng, "Finding splitting lines for touching cell nuclei with a shortest path algorithm," *Comput. Biol. Med.*, vol. 63, pp. 277 - 286, Aug. 2015.

[33] M. Maitra, R.K. Gupta, and M. Mukherjee, "Detection and counting of red blood cells in blood cell images using Hough transform," *Int J. Comput. Appl.*, vol. 53, no. 16, pp. 13 - 17, 2012.

[34] J. Ge, Z. Gong, J. Chen, J. Liu, J. Nguyen, Z. Yang, C. Wang, and Y. Sun, "A system for counting fetal and maternal red blood cells," *IEEE Trans. Biomed. Eng.*, vol. 61, no. 12, pp. 2823 - 2829, Dec. 2014.

[35] B. Parvin, Q. Yang, J. Han, H. Chang, B. Rydberg, M.H. Barcellos-Hoff, "Iterative voting for inference of structural saliency and characterization of subcellular events," *IEEE Trans. Imag. Process*, vol. 16, pp. 615 - 623, 2007.

[36] H. Xu, C. Lu, and M. Mandal, "An efficient technique for nuclei segmentation based on ellipse descriptor analysis and improved seed detection algorithm," *IEEE J. Biomed. Health Inform.*, vol. 18, pp. 1729 - 1741, 2014.

[37] F. Xing, H. Su, J. Neltner, and L. Yang, "Automatic ki-67 counting using robust cell detection and online dictionary learning," *IEEE Trans. Biomed. Eng.*, vol. 61, no. 3, pp. 859 - 870, Mar. 2014.

[38] C. Lu, H. Xu, J. Xu, H. Gilmore, M. Mandal, and A. Madabhushi, "Multi-Pass Adaptive Voting for Nuclei Detection in Histopathological Images," *Scientific Reports*, vol. 6, pp. 33985 EP, 2016.

[39] X. Qi, F. Xing, D. J. Foran, and L. Yang, "Robust segmentation of overlapping cells in histopathology specimens using parallel seed detection and repulsive level set," *IEEE Trans. Biomed. Eng.*, vol. 59, pp. 754 - 765, 2012.

[40] N. Malpica, C.O. de Solorzano, J.J. Vaquero, A. Santos, and I. Vallcorba, "Applying watershed algorithms to the segmentation of clustered nuclei," *Cytom. Part A*, vol. 28, pp. 289 - 297, 1997.

[41] M.E. Plissiti, C. Nikou, and A. Charchanti, "Combining shape, texture and intensity features for cell nuclei extraction in pap smear images," *Pattern Recognit. Lett.*, vol. 32, pp. 838 - 853, 2011.

[42] M. Maska, O. Danek, S. Garasa, A. Rouzaut, A. Munoz-Barrutia, and C.O. Solorzano, "Segmentation and shape tracking of whole fluorescent cells based on the Chan-Vese model," *IEEE Trans. Med. Imaging*, vol. 32, no. 6, pp. 995 - 1006, 2013.

[43] P. Wu, J. Yi, G. Zhao, Z. Huang, B. Qiu, and D. Gao, "Active contour-based cell segmentation during freezing and its application in cryopreservation," *IEEE Trans. Biomed. Eng.*, vol. 62, no. 1, pp. 284 - 295, 2015.

[44] A. Tareef, Y. Song, W. Cai, D.D. Feng, and M. Chen, "Automated three-stage nucleus and cytoplasm segmentation of overlapping cells," *Control Automat. Robot. Vision*, pp. 865 - 870, 2014.

[45] A. Lucchi, K. Smith, R. Achanta, V. Lepetit and P. Fua, "A Fully Automated Approach to Segmentation of Irregularly Shaped Cellular Structures in EM Images," in *International Conference Med. Image Comput. Computer-Assisted Interven.*, Lecture Notes in Computer Science, 2010.

[46] Y. Song, D. Ni, L. He, S. Chen, B. Lei and T. Wang, "Segmentation and Splitting of Touching Vaginal Bacteria Based on Superpixel and Effective Distance," in *IEEE International Conference Data Mining Workshop*, pp. 976 - 981, 2014.

[47] A. Genctav, S. Aksoy, and S. Onder, "Unsupervised segmentation and classification of cervical cell images," *Pattern Recognit.*, vol. 45, no. 12, pp. 4151 - 4168, Dec. 2012.

[48] H. Su, Z. Yin, S. Huh, and T. Kanade, "Cell segmentation in phase contrast microscopy images via semi-supervised classification over optics-related features," *Med. Image Anal.*, vol. 17, no. 7, pp. 746 - 765, Apr. 2013.

[49] C. Zhang, J. Yarkony, F.A. Hamprecht, "Cell detection and segmentation using correlation clustering," in *Proceed. International Conference Med. Image Comput. Computer-Assisted Interven.*, vol. 8673, pp. 9 - 16, 2014.

[50] Z. Lu, G. Carneiro, and A.P. Bradley, "Automated nucleus and cytoplasm segmentation of overlapping cervical cells," in *International Conference Med. Image Comput. Computer-Assisted Interven.*, pp. 452 - 460, 2013.

[51] C. Arteta, V. Lempitsky, J. A. Noble, A. Zisserman, "Learning to detect cells using non-overlapping extremal regions," in *Proceed. International Conference Med. Image Comput. Computer-Assisted Interven.*, pp. 348 - 356, 2012.

[52] C. Arteta, V. Lempitsky, J.A. Noble, and A. Zisserman, "Detecting overlapping instances in microscopy images using extremal region trees," *Med. Image Anal.*, vol. 27, pp. 3 - 16, 2016.

[53] P.D. Kostelec, L.M. Carliny, and B. Glocker, "Learning to detect and track cells for quantitative analysis of time-lapse microscopic image sequences," in *Proceed. IEEE International Symp. Biomed. Imag.*, pp. 1544 - 1547, 2015.

[54] R. Bise and Y. Sato, "Cell detection from redundant candidate regions under nonoverlapping constraints," *IEEE Trans. Med. Imaging.*, vol. 34, pp. 1417 - 1427, 2015.

[55] S.K. Nath, K. Palaniappan, and F. Bunyak, "Cell Segmentation Using Coupled Level Sets and Graph-Vertex Coloring," in *International Conference Med. Image Comput. Computer-Assisted Interven.* pp. 101 - 108, 2006.

[56] Y. Al-Kofahi, W. Lassoued, W. Lee, and B. Roysam, "Improved automatic detection and segmentation of cell nuclei in histopathology images," *IEEE Trans. Biomed. Eng.*, vol. 57, pp. 841 - 852, 2010.

[57] X. Lou, U. Koethe, J. Wittbrodt, and F. Hamprecht, "Learning to segment dense cell nuclei with shape prior," in *Proceed. IEEE Conference Comput. Vis. Pattern Recog.*, pp. 1012 - 1018, 2012.

[58] S. Arslan, T. Ersahin, R. Cetin-Atalay, and C. Gunduz-Demir, "Attributed relational graphs for cell nucleus segmentation in fluorescence microscopy images," *IEEE Trans. Med. Imaging*, vol. 32, pp. 1121 - 1131, 2013.

[59] K.Z. Mao, P. Zhao, and P.H. Tan, "Supervised learning-based cell image segmentation for p53 immunohistochemistry," *IEEE Trans. Biomed. Eng.*, vol. 53, no. 6, pp. 1153 - 1163, 2006.

[60] H. Kong, M. Gurcan, and K. Belkacem-Boussaid, "Partitioning histopathological images: An integrated framework for supervised color-texture segmentation and cell splitting," *IEEE Trans. Med. Imag.*, vol. 30, no. 9, pp. 1661 - 1677, 2011.

[61] C. F. Koyuncu, I. Durmaz, R. Cetin-Atalay and C. Gunduz-Demir, "A supervised learning model for live cell segmentation," in *IEEE 22nd Signal Processing and Communications Applications Conference*, pp. 1971-1974, 2014.

[62] H. Su, F. Xing, J.D. Lee, C.A. Peterson, and L. Yang, "Automatic myonuclear detection in isolated single muscle fibers using robust ellipse fitting and sparse representation," *IEEE Trans. Comput. Biology Bioinfo.*, vol. 11, no. 4, pp. 714 - 726, 2014.

[63] H. Su, F. Xing, J.D. Lee, C.A. Peterson, and L. Yang, "Learning based automatic detection of myonuclei in isolated single skeletal muscle fibers using multi-focus image fusion," *IEEE International Symp. Biomed. Imag.*, pp. 432 - 435, 2013.

[64] P. Khurd, L. Grady, A. Kamen, S. Gibbs-Strauss, E.M. Genega, and J.V. Frangioni, "Network cycle features: application to computer-aided gleason grading of prostate cancer histopathological images," *IEEE International Symp. Biomed. Imag.*, pp. 1632 - 1636, 2011.

[65] C. Sommer, L. Fiaschi, F.A. Hamprecht, D.W. Gerlich, "Learning-based mitotic cell detection in histopathological images," in *IEEE International Conference Pattern Recognit.*, pp. 2306 - 2309, 2012.

[66] J. Xu, L. Xiang, Q. Liu, H. Gilmore, J. Wu, J. Tang, and A. Madabhushi, "Stacked sparse autoencoder (SSAE) for nuclei detection on breast cancer histopathology images," *IEEE Trans. Medical Imag.*, vol. 35, pp. 119 - 130, 2016.

[67] H. Su, F. Xing, X. Kong, Y. Xie, S. Zhang, and L. Yang, "Robust cell detection and segmentation in histopathological images using sparse reconstruction and stacked denoising autoencoders," in *Proceed. International Conference Med. Image Comput. Computer-Assisted Interven.*, pp. 383 - 390, 2015.

[68] Y. Song, L. Zhang, S. Chen, D. Ni, B. Lei, and T. Wang, "Accurate segmentation of cervical cytoplasm and nuclei based on multiscale convolutional

network and graph partitioning," *IEEE Trans. Biomed. Eng.*, vol. 62, pp. 2421 - 2433, 2015.

[69] Y. Song, E.L. Tan, X. Jiang, J.Z. Cheng, D. Ni, S. Chen, B. Lei, and T. Wang, "Accurate cervical cell segmentation from overlapping clumps in pap smear images," *IEEE Trans. Med. Imag.*, vol. 36, no. 1, pp. 288 - 300, Jan. 2017.

[70] D.C. Cireşan, A. Giusti, L.M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Advances Neural Information Process. Systems*, pp. 2843 - 2851, 2012.

[71] D.C. Cireşan, A. Giusti, L.M. Gambardella, and J. Schmidhuber, "Mitosis detection in breast cancer histology images with deep neural networks," in *International Conference Med. Image Comput. Computer-Assisted Interven.*, pp. 411 - 418, 2013.

[72] B. Dong, L. Shao, M.D. Costa, O. Bandmann, and A.F. Frangi, "Deep learning for automatic cell detection in wide-field microscopy zebrafish images," in *Proceed. IEEE 12th International Symp. Biomed. Imag.*, pp. 772 - 776, Apr. 2015.

[73] Y. Xie, F. Xing, X. Kong, H. Su, and L. Yang, "Beyond classification: structured regression for robust cell detection using convolutional neural network," in *International Conference Med. Image Comput. Computer-Assisted Interven.*, pp. 358 - 365, 2015.

[74] Y. Mao, Z. Yin, and J.M. Schober, "Iteratively training classifiers for circulating tumor cell detection," in *Proceed. IEEE 12th International Symp. Biomed. Imag.*, pp. 190 - 194, Apr. 2015.

[75] S.K. Sadanandan, P. Ranefall, S.L. Guyader, and C. Wahlby, "Automated Training of Deep Convolutional Neural Networks for Cell Segmentation," *Scientific Reports*, vol. 7, no. 7860, Aug. 2017.

[76] P. Kainz, M. Urschler, S. Schulter, P. Wohlhart, and V. Lepetit, "You should use regression to detect cells," in *International Conference Med. Image Comput. Computer-Assisted Interven.*, pp. 276 - 283, 2015.

[77] Y. Xie, X. Kong, F. Xing, F. Liu, H. Su, and L. Yang, "Deep voting: A robust approach toward nucleus localization in microscopy images," in *International Conference Med. Image Comput. Computer-Assisted Interven.*, pp. 374 - 382, 2015.

[78] K. Sirinukunwattana, S.E.A. Raza, Y.W. Tsang, D. R. Snead, I.A. Cree, and N.M. Rajpoot, "Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images," *IEEE Trans. Medical Imag.*, vol. 35, pp. 1196 - 1206, 2016.

[79] H. Chen, W. Wang, and P.A. Heng, "Automated mitosis detection with deep regression networks," in *IEEE 13th International Symp. Biomed. Imag.*, pp. 1204 - 1207, 2016.

[80] Y. Xie, F. Xing, X. Shi, X. Kong, H. Su, L. Yang, "Efficient and robust cell detection: A structured regression approach," *Medical Image Analys.*, vol. 44, pp. 245 - 254, 2018.

[81] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceed. IEEE Conference Comput. Vision Pattern Recognit.*, pp. 3431 - 3440, 2015.

[82] X. Weidi, J.A. Noble, and A. Zisserman, "Microscopy cell counting with fully convolutional regression networks," in *International Conference Med. Image Comput. Computer-Assisted Interven.*, 1st Workshop on Deep Learning in Medical Image Analysis, Nov. 2015.

[83] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference Med. Image Comput. Computer-Assisted Interven.*, pp. 234 - 241, 2015.

[84] W. Xie, J. A. Noble, and A. Zisserman, "Microscopy cell counting and detection with fully convolutional regression networks," *Comput. Methods Biomechanics Biomed. Eng.: Imag. & Visualization*, vol. 6, pp. 283 - 292, 2018.

[85] L.C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A.L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets,

atrous convolution, and fully connected crfs," *IEEE Trans. Pattern Anal. Machine Intelligence*, vol. 40, pp. 834 - 848, 2018.

[86] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Sys. Man.*, vol. 9, pp. 62 - 66, 1979.

[87] F. Raimondo, M.A. Gavrielides, G. Karayannopoulou, K. Lyroudia, I. Pitas, and I. Kostopoulos, "Automated evaluation of Her-2/neu status in breast tissue from fluorescent in situ hybridization images," *IEEE Trans. Imag. Process*, vol. 14, pp. 1288 - 1299, 2005.

[88] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceed. International Conference Learn. Representations*, 2015.

[89] S. Ram, V.T. Nguyen, K.H. Limesand, and M.R. Sabuncu, "Joint cell nuclei detection and segmentation in microscopy images using 3d convolutional networks," *arXiv preprint arXiv:1805.02850*, 2018 .

[90] H. Chen, X. Qi, L. Yu, Q. Dou, J. Qin, and P.A. Heng, "Dcan: Deep contour-aware networks for object instance segmentation from histology images,", *Medical Image Analys.*, vol. 36, pp. 135 - 146, 2017.

[91] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, pp. 41 - 75, 1997.

[92] G. Litjens, T. Kooi, B.E. Bejnordi, A. A. Setio, F. Ciompi, M. Ghafoorian, J.A. Van der Laak, B. Van Ginneken, and C.I. Sánchez, "A survey on deep learning in medical image analysis," *Medical Image Analys.*, vol. 42, pp. 60 - 88, 2017.