March 15, 2013

A General System for Supervised Biomedical Image Segmentation

Cheng Chen

Department of Biomedical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Dr. Gustavo K. Rohde, Chair
Dr. Jelena Kovacevic
Dr. Ge Yang
Dr. Chunming Li

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Keywords: image segmentation, supervised learning, pixel classification, intensity neighborhood, data selection, majority voting, statistical modeling, template matching, non-rigid registration

The fundamental elements required for completing this doctoral dissertation are confidence, enthusiasm, and hard work, but it would never have been completed if without the support from my family. Thus, I dedicate this important achievement to my family. I would never have been able to reach this goal without their support and understanding.

Abstract

Image segmentation is important with applications to several problems in biology and medicine. While extensively researched, generally, current segmentation methods perform adequately in the applications for which they were designed, but often require extensive modifications or calibrations before used in a different application. We describe a system that, with few modifications, can be used in a variety of image segmentation problems. The system is based on a supervised learning strategy that utilizes intensity neighborhoods to assign each pixel in a test image its correct class based on training data. In summary, we have several innovations: (1) A general framework for such a system is proposed, where rotations and variations of intensity neighborhoods in scales are modeled, and a multi-scale classification framework is utilized to segment unknown images; (2) A fast algorithm for training data selection and pixel classification is presented, where a majority voting based criterion is proposed for selecting a small subset from raw training set. When combined with 1-nearest neighbor (1-NN) classifier, such an algorithm is able to provide descent classification accuracy within reasonable computational complexity. (3) A general deformable model for optimization of segmented regions is proposed, which takes the decision values from previous pixel classification process as input, and optimize the segmented regions in a partial differential equation (PDE) framework. We show that the performance of this system in several different biomedical applications, such as tissue segmentation tasks in magnetic resonance and histopathology microscopy images, as well as nuclei segmentation from fluorescence microscopy images, is similar or better than several algorithms specifically designed for each of these applications.

In addition, we describe another general segmentation system for biomedical applications where a strong prior on shape is available (e.g. cells, nuclei). The idea is based on template matching and supervised learning, and we show the examples of segmenting cells and nuclei from microscopy images. The method uses examples selected by a user for building a statistical model which captures the texture and shape variations of the nuclear structures from a given data set to be segmented. Segmentation of subsequent, unlabeled, images is then performed by finding the model instance that best matches (in the normalized cross correlation sense) local neighborhood in the input image. We demonstrate the application of our method to segmenting cells and nuclei from a variety of imaging modalities, and quantitatively compare our results to several other methods. Quantitative results using both simulated and real image data show that, while certain methods may work well for certain imaging modalities, our software is able to obtain high accuracy across several imaging modalities studied. Results also demonstrate that, relative to several existing methods, the template based method we propose presents increased robustness in the sense of better handling variations in illumination, variations in texture from different imaging modalities, providing more smooth and accurate segmentation borders, as well as handling better cluttered cells and nuclei.

Acknowledgments

First and foremost, I am deeply grateful to my advisor Dr. Gustavo K. Rohde. It has been a great pleasure to be his Ph.D. student and work with him during the past four and a half years. He is a very nice person, who has taught me a lot on how to think of scientific problems mathematically, and how to become a good researcher. I sincerely appreciate all his contributions of time, ideas, and funding to make my Ph.D. experience productive and stimulating during these years. His enthusiasm for his research also motivates me to work hard and never give up on my research projects, even when dealing with some difficult research problems in the Ph.D. pursuit.

Furthermore, I am very grateful to my Ph.D. thesis committee members: Dr. Gustavo K. Rohde (chair), Dr. Jelena Kovacevic, Dr. Ge Yang, and Dr. Chunming Li, for taking time to serve as my committee members. I would like to thank them for their insightful comments and suggestions for my Ph.D. research and this dissertation.

I also want to thank our collaborator, Dr. John A. Ozolek from Children's Hospital of Pittsburgh, University of Pittsburgh Medical Center (UPMC). I have been collaborating with Dr. Ozolek since I started my Ph.D. program in Carnegie Mellon University in 2008. Dr. Ozolek not only provided biomedical data for my projects, but also taught me a lot of basic biomedical knowledge, which helps me to better understand the biological meaning of these projects and how important it will be if the problem can be solved.

Specifically, I want to thank all the members in Dr. Rohde's group, together with other colleagues in Center for Bioimage Informatics (CBI) for their immense contribution to my personal and professional time in Carnegie Mellon. The group has been a source of friendships as well as good advice and collaboration. These past and present members that I have had the pleasure to work with include: Wei Wang, Tao Peng, Aabid Shariff, Anupama Kuruvilla, Jieyue Li, Saurav Basu, Soheil Kolouri, Akif Burak Tosun, Jia Guo, Hu Huang, Se Rim Park, etc., and many, many summer and rotation students who have come through CBI.

Finally, I would like to thank my family for all their love, support and encouragement. For my parents who raised me with a love of science and supported me in all my pursuits. For my grandparents with whom I grew up during my childhood. And most of all for my girlfriend Ming Feng, who always supports me faithfully during the final stage of this Ph.D. Thank you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Image segmentation is a classical research problem that has been investigated for many years in the area of digital image processing. Generally, image segmentation is to subdivide an image into several constituent regions (named as segments), and each segment contains a set of pixels, where pixels in the same segment share certain visual characteristics such as color, intensity, or texture. The goal is to simplify the representation of image into something that is more meaningful and easier to analyze. When the regions of interest (e.g. objects/boundaries) in an application have been isolated, the segmentation process stops. To be more precise, in order to recognize the isolated objects/boundaries, each segment will be assigned a label, and in general, adjacent segments are assigned with different labels, as they are significantly different with respect to the same characteristic(s).

1.1 Review of current image segmentation techniques

Segmentation is one of the most studied problems in image processing. Early in 1975, Otsu [1] has proposed a simple and automated segmentation method based on the shape of histogram, which separates a gray level image into foreground and background via a threshold value. Till now, while Otsu's algorithm is still being used widely in many applications, several thousand papers are being published each year at the same time. Here we give a brief overview which is not meant to be exhaustive or complete. Rather our intent is to highlight the main ideas used so far in the field. We start by summarizing the current segmentation techniques into two big categorites based on the level of image information they use for segmentation. The first category include those image segmentation techniques that exploit low level image information, such as image intensity and image gradient information. There are several typical classes of methods, which will be described and commented as follows:

Thresholding methods are usually regarded as the simplest segmentation methods, which uses one threshold value or multiple threshold values to turn a gray level images into a binary image or a segmented image with multiple classes. One famous example is Otsu's method [1] mentioned above, which assumes that the optimal threshold value makes the intra-class variance of the two classes (e.g. foreground and background) to be minimal. Such a method can be extended to multi-level thresholding [2, 3]. In summary, we mention that there are many variants

of thresholding based algorithms, such as: (1) histogram shape based algorithms, such as Otsu's method [1], balanced histogram thresholding [4], etc. (2) maximum entropy based algorithms [5]: instead of maximizing the intra-class variance, the method aims to find an optimal threshold value that maximizes the inter-class entropy. Considering that images may be screened under uneven illumination, adaptive thresholding algorithms [6, 7] are proposed, which selects the local threshold values at different regions of the image intelligiently.

Edge detection methods are usually designed for detecting the boundaries of regions. Common edge detection algorithms design various of spatial filters [8, 9] to compute the first-order or second-order gradient information of the image. For example, Sobel or directional derivative masks [9] are used to calculate gradient information, while Laplacian filters [9] are used to calculate the second-order gradient information. It is worth noting that edge detection itself cannot be called segmentation, while in most segmentation cases, people are more interested in isolating the regions of interests rather than finding the boundries of these regions. Thus, these detected regions must be linked by the edge-linking procedures in order to form closed regions. However, due to the noise, artifacts, or geometrical occulsions inside the images, it becomes a quite difficult job to link edges very well. Some bounary tracking methods have been proposed. One way using graph based search [10, 11] is to find paths between the start and end pixels that minimizes a cost function based on the distance and transition probabilities. Among several graph based search algorithms, the $A*$ algorithm [10, 11] is widely used. In addition, Hough transform [12] is widely used to detect straight lines and other parametric curves such as circles, ellipses, and so forth.

Region based methods include region growing [13, 14], region merging methods [14, 15] and reigon splitting methods [15]. Region growing method usually find the regions of interest by examining pixels in the neighborhood from the starting places based on a pre-defined similarity criterion. The neighboring pixels that satisfy the similarity criterion will be merged into the current regions for segmentation. Such an idea is also extended to merge not only pixels but also larger meaningful regions of similar properties. Region merging is often used as a post processing step after segmentation so that these tiny segmented regions can be merged to the neighboring larger regions for better recognition and interpretation. Comparatively, region splitting methods start from the whole image or larger regions, and split them into two or more regions based on a dissimilarity criterion. Many criteria have been proposed, for example, if a region has a bimodal distribution of gray value histogram, it can be split into two regions of connected pixels with gray values falling in their respective distributions. In summary, region based methods are often combined with other segmentation techniques and used as a post processing step for improving the segmentation performance.

Specifically, **watershed algorithms** [9, 16, 17] are usually regarded as one of the most famous and frequently used morphological based/region growing methods for image segmentation. The basic concept is to view a gray level image, for example, as a topographic relief, where the gray level of a pixel is interpreted as its altitude in the relief. A drop of water falling on a topological relief flows along a path to finally reach a local minimum. The set of pixels at which water would be equally likely to fall to more than one such minimum form crest lines on the topographic surface are called watershed lines which separate regions. There are many different ways for building the topological relief besides using gray levels directly [18]. For example, one intuition is to regard the area with high gradient values as the edges, thus, the topographic relief

can be built by using the gradient image. Another common way is based on shape information, where a distance transformation map (Euclidean distance is often chosen) can be calculated from a binary image after the initial segmentation and utilized as the topographical relief. It is worth mentioning that the direct use of watershed algorithms is very easy to produce over-segmentation results. Thus, marker controlled watershed (or called seeded watershed) algorithm [9, 16, 17] is more often used for practical segmentation. A marker is defined as a connected component belonging to an image, where internal markers are associated with objects of interest, while external markers are associated with background. Usually, markers can be manually labeled, or selected automatically based on some pre-defined criteria. Finally, the topographical relief is modified as minimum only exist at the foreground and background marker locations in order to prevent over-segmentation.

Although there are still many other existing algorithms that can be categorized into the first category, we omit them for brevity and conclude that these methods that explore low level image information for segmentation share the characteristics that they are easy to implement, however, most of them are not easy to achieve satisfactory results. Thus, we will turn to the second category of segmentation methods that exploit constraints derived from the image data together with high level a priori knowledge of the objects. Since higher level knowledge is incorporated into these methods, a more robust segmentation can be obtained. In the following, we will discuss several typical methods:

Multi-scale methods [19, 20, 21] segment images at multiple scales in scale space. The general idea is to propagate from coarse to fine scales for image segmentation. Lindeberg [22, 23] studied the problem of linking local extrema and saddle points over scales, and thus, represented the image in the way called "the scale-space primal sketch", which makes explicit the relations between structures at different scales and the stable image features over scales. Based on the theory, a couple of methods have been proposed. For examples, edges can be first detected at coarse scales in scale-space and then, traced back to finer scales. Watershed can also be implemented in multi-scales [21]. And these ideas have been incorporated into automated system [24] and extensively tested on some databases (e.g. brain) [25].

Deformable model based methods [26, 27, 28] is a group of computer algorithms sharing the common characteristics that they model the variablity of a certain class of objects. If defined within image domain, deformable models are usually regarded as curves or surfaces that can move under the influence of internal forces, which are defined within the curve or surface itself, while the external forces can be calculated from the image data. The use of internal force is to keep the model smooth during deformation, while the use of external force is to move the model toward an object boundary or other desired features within an image. Deformable models have been extensively studied and widely used in biomedical image segmentation [26] with promising results, since they are able to offer robustness to both image noise and boundary gap by constraining extracted boundaries to be smooth and incorporating other prior information such as the object shape. To summarize, current deformable model based methods for image segmentation can be divided into two big categories: parametric/explicit based methods [29, 30, 31] and non-parametric/implicit based methods [28, 32, 33]. In parametric deformable models, curves and surfaces are represented explicitly in their parametric forms during deformation. One classical example is active contour model, or called "snakes" [34, 35], which delineates the outline of an object by minimizing the energy function as the sum of external and internal energy, where

external energy utilizes the image gradient information directly, while internal energy is based on curvature of contours. Parametric deformable models allow direct interaction with the model and can be implemented quite fast. However, it has the disadvantage of topology constraint, which cannot deal with splitting or merging cases during the process of deformation. Thus, non-parametric deformable model is proposed, where curves and surfaces are represented by a level set of a higher-dimensional scalar function [28, 36, 37]. The advantage of level set is that numerical computation involving curves and surfaces on a fixed Cartesian grid can be performed without parameterization. In addition, level set makes it very easy to follow shapes that change topology. One popular deformable model based on level set is Chan and Vese (C-V) model [38] . We note that there are a great number of variant deformable models and we will discuss them in Chapter 4.

Graph partition methods [39, 40, 41, 42] implement image segmentation by modeling the image as a weighted, undirected graph. Usually, each pixel inside the image is regarded as one node in the graph, while edges are built by connecting pairwise nodes. Usually, only neighboring pixels are connected by edges, and edge weights define the similarity/dissimilarity between the neighboring pixels, such as distance, intensities. Image segmentation is implemented by partitioning the graph according to different criteria designed to model "good" clusters. Each partition of the nodes corresponds to a set of pixels as an isolated region in the image. Popular methods include: normalized cuts [39], random walker [40], minimum cut [41], etc.

Especially, we would like to mention the **graph cuts** method [43, 44, 45] for image segmentation. Although this method is still a graph based algorithm, it is different from previous graph partition methods in that it formulates the image segmentation in terms of energy minimization, and represents such an energy objective function into a graph. The optimization of the energy function is realized via finding the minimum cut/maximum flow of the graph. In most cases, the minimum energy solution corresponds to the maximum a posterior estimate of a solution.

Template based methods (or called atlas based methods) [46, 47] are based on the assumption that the structures of objects to be segmented have a repetitive form of geometry. Such cases are often seen in biomedical application, for example, segmentation of gray matter and white matter tissues from human brain MR/CT images [47, 47], where the brain structures between different people are similar. Therefore, one can manually label a few examples and then, build a probabilistic model that explains the variation of the shape of the organ. This model can be used to impose prior constraints when segmenting an image. Usually, such a task consists of several steps: (1) register the training sample to the same pose, (2) calculate the probabilistic representation of the variation of the registered samples, and (3) statistical inference between the model and the image to be segmented. State of the art methods in the literature for knowledge-based segmentation include active shape [48, 49] and active appearance models [50, 51].

Learning based method [52] are those who utilize machine learning techniques for image segmentation. Generally, current methods can be divided into two categories: unsupervised learning based methods [52, 53, 54, 55] and supervised learning based methods [52, 56, 57]. Unsupervised learning tries to find hidden structures in unlabeled data. As far as we know, most unsupervised learning based segmentation methods use clustering to segment images. Here, we show a few typical ones: The simplest one is k-means clustering [58, 59], where usually, pixels' intensities are clustered into k clusters so that the whole image is segmented into regions belonging to k different classes. Another clustering method commonly used is called fuzzy c-

means (fcm) clustering [60]. Unlike k-means clustering, fuzzy c-means algorithm allows one piece of data to belong to two or more clusters with a weight updated at each iteration, and fuzzy partitioning is carried out through an iterative optimization of the objective function. In addition, by assuming that the pixels' intensities of each class follows a Gaussian distribution (a common assumption), expectation maximization (EM) algorithm [61] is used for estimating the parameters of Gaussian distributions and clustering. To summarize, most clustering algorithms are easy to implement, however, it is quite difficult for them to generate satisfactory results when dealing with complicated image data. Comparatively, supervised learning infers a classification function from labeled training data that consists of a set of training examples (typically a vector and a label value representing which class the example belongs to), and predict the class label for unknown examples. For image segmentation, supervised learning methods [57, 62] have the advantage of handling complicated image data by not only using color information but also using a huge domain knowledge such as texture. One common way is to predict the label of each pixel in the unknown images so that the pixels with the same label value can be isolated as the segmented regions. Many existing supervised learning techniques can be utilized, such as neural networks [63], support vector machines (SVM) [52, 64], etc.

1.2 Image segmentation for practical applications

Various of image segmentation techniques have been developed and widely utilized in many practical applications in the real world. Here, we illustrate a few common applications including:

Content based image retrieval (CBIR) [65], also known as content-based visual information retrieval, aims to search for digital images in large databases. The segmentation process is one of the first and important steps. In CBIR systems, the segmentation step aims to facilitate the interpretation of images and scenes by identifying homogeneous and significant regions, which can be further analyzed by the system using regions features and possibly their semantic meaning.

Medical imaging [66] is the technique and process that creates images of the human body for clinical diagnosis, or medical science such as the study of anatomy and physiology. One important task in medical imaging field is to analyze the imaging results intelligently, where segmentation is the first and significant step that isolates the biological structures of interest, such as organs, tissues, etc. Quality of segmentation determines the following analysis process directly.

Object detection [67, 68, 69] is a computer technology related to computer vision and image processing that aims to detect objects of a certain class (such as human, face, cars, etc) from digital images and videos. Popular research topics on object detection include face detection and pedestrian detection in video surveillance. Since segmentation is a process that partitions an image into regions. In the problem of face detection, one way is to segment skins in the image, which helps to identify the probable regions containing the faces and reduce the search space. For pedestrian detection, image segmentation can also be useful. One example is to segment edges in order to recognize the pedestrians in the videos.

We mention here that there are still many fields where image segmentation plays an important role, and we will not discuss them for brevity. Specific application of image segmentation in

biomedical imaging will be discussed carefully in the next section.

1.2.1 Application of segmentation in biomedical imaging

To be specific, we focus on discussing the use of image segmentation in biomedical imaging field in this section. In the past decades, new imaging technologies have been increasingly developed and applied to different biomedical research and clinical problems. Given the enormous success with which some of these technologies have been applied to research and clinical tasks, the trend of image technology development (and their application in novel biomedical problems) is likely to continue. Thus, image segmentation will be of great importance in the application of imaging technology to many biomedical problems. Tissues, cells, and organs must often be segmented and isolated from two or three dimensional digital image data for subsequent quantitative analysis in a variety of experimental biological studies and diagnosis in clinical medicine. Popular examples include: segmentation of gray and white matter tissues from magnetic resonance imaging (MRI) brain scans for studying neurological diseases (e.g. Alzheimer's); segmentation of cells and tissues from histopathology image data to assist in the diagnosis of different lesions; segmentation of nuclei cells and subcellular structures for characterizing their distributions relating to cancer diagnosis, etc.

Due to the vast increase in the capability of image acquisition during the past couple of decades, a huge amount of data (e.g. digital images) is usually produced. Thus, manual segmentation is no longer viable effective for many types of quantitative studies, and computer-aided segmentation methods become an urgent need. The increase in computational power in recent decades has spurred the development of a number of image segmentation algorithms [34, 38, 44, 70, 71, 72, 73] that have been applied to a variety of biomedical and clinical applications with successful use. Here, we list a few examples in the following:

- **MR-images segmentation**: Magnetic resonance imaging (MRI) is a significant medical imaging technique used in radiology to visualize detailed internal structures, which makes use of the property of nuclear magnetic resonance to image nuclei of atoms inside the body. Since MRI is able to provide good contrast between the different soft tissues of the body, it has been widely used in imaging the brain, muscles, the heart, and the cancers, etc. One popular example is to use MRI to screen human brains for clinical diagnosis. For quantitative diagnosis purpose, different tissues in brain MR data, such as gray matter, white matter, etc., need to be segmented accurately. In the past decades, a variety of computational algorithms have been proposed for brain MR-segmentation. For example, Wells et al. [74] present a method that couples tissue classification with in-homogeneity correction based on maximum-likelihood parameter estimation, and they use the expectation-maximization (EM) algorithm to estimate the unknown bias field and the classification. Leemput et al. [75] extend this approach by using Gaussian mixture density estimation to estimate the grayscale intensity PDFs for each tissue type. They also apply the EM algorithm to estimate these PDFs as well as the bias and, in turn, the classification. In addition, Zhang et al. [70] extended the EM-classification algorithm to incorporate spatial smoothness via Markov priors on the label image. In [76], Awate et al. estimate the density distribution through nonparametric Parzen density window and utilize intensity neighborhood to keep

the Markov property. In addition, the active contour model [77, 78] has also been used to impose smoothness constraints for segmentation. These methods typically attempt to minimize the area of the segmentation boundary simultaneously with proper fidelity to the data. However, the drawbacks of active contour based methods are their sensitivity to noise and manual parameter tuning. Another significant application of MRI is for the diagnosis of cancer, such as prostate cancer, which is a leading cause of cancer death for men in the United States. MRI has been used as a noninvasive imaging method and has shown to be useful to visualize and localize the prostate gland. Automated detection/localization of prostate cancer within prostate MR images is usually required for further quantitative diagnosis. In [79], conditional random fields (CRF) combined with SVM is used to detect and localize the area of prostate cancer. In [80], an active appearance model (AAM) utilizing level set implementation has been proposed to overcome the constraints of landmarks, and multiple features have been used to help locate the region of interests. Some other MRI applications include: segmentation of heart from 3D MRI imaages [81, 82], segmentation of blood vessels [83], etc. We note that there are still many, many example for MRI applications, and the discussions will be omitted here for brevity.

- **Histology image segmentation**: Histology is the study of the microscopic anatomy of cells and tissues of plants and animals. It is usually performed by staining the tissue slice and examining it under a light microscope or an electronic microscope. Since accurate diagnosis of cancer and other diseases usually requires histopathological examination of samples, it is important to study the diseased tissues under the microscope. Currently, histological images are generated by scanning the stained tissue slices using a high resolution scanner. When a large amount of histological images are produced, computational diagnosis methods become necessary, in which the detection or segmentation of tissue/histological structures (e.g. cancer nuclei, glands) becomes the first significant step for the further quantitative analysis such as disease grading, as the presence, extent, size, shape and other morphological appearance of these structures are important indicators for severity of disease. Tissue/histological structure detection or segmentation in different types of histological images is a very challenging task as the tissues inside the histological images are usually very complicated, and thereby, difficult to model [84, 85, 86]. Many methodologies have been developed for segmenting specific objects of interest from histology images in the past years. For example, some methods [36, 87, 88] focus on segmenting various structures in histology images, such as segmenting nuclei [87, 89], segmenting lumen and epithelial cytoplasm [36]. Comparatively, identifying and segmenting the whole tissue regions rather than inner structures from histology images is a more challenging task, and we mention that machine learning based methods are preferred due to the fact of their learning abilities. For these data whose tissues are relatively simple, the unsupervised learning methods such as K-means algorithm utilizes color information for clustering works well as we think the color properties of each tissue class should be similar [90]. When the tissues become complicated, researchers developed supervised learning methods and utilized different features for segmentation or identification. For example, in [91, 92], a multiresolution approach has been used to classify the whole slide histology images in high resolution. In [93], Bhagavatula et al. used histopathology vocabulary features for tissue

identification in teratoma histological images. In [57], intensity neighborhood is utilized as a general feature, which provides decent segmentation performance on histology images. A more detailed discussion about techniques used for histology image analysis can be seen in [94].

- **Fluorescence microscopic image segmentation**: Fluorescence microscopic images are another type of data commonly seen. Usually, cells, sub-cellular structures, nuclei are screened by high throughput fluorescence microscopy and a great amount of data can be produced. Objects of interest, such as cells, nuclei, etc., must be accurately segmented for further quantitative analysis. Nuclei segmentation is a relatively easy task. Usually, nuclei are well screened, and even simple intensity based thresholding methods can work well. Sometimes, when nuclei are screened under uneven illumination environment, intensity heterogeneity of the data has to be processed before segmentation. When dealing with the touching or overlapped cases of nuclei/cells, seeded watershed method [17] is usually utilized for separation. Comparatively, cell segmentation is more difficult, since cells usually touch each other tightly. A general solution in current literatures is to first stain DNA channel of the cell in order to get the nuclei images. Nuclei will be segmented to provide prior information of cells, such as the amount, location, etc. Then, the actin channel of the cell is stained to show the cytoplasms, which will be segmented and separated using the prior information of the nuclei. In [95, 96], these methods are discussed and compared.

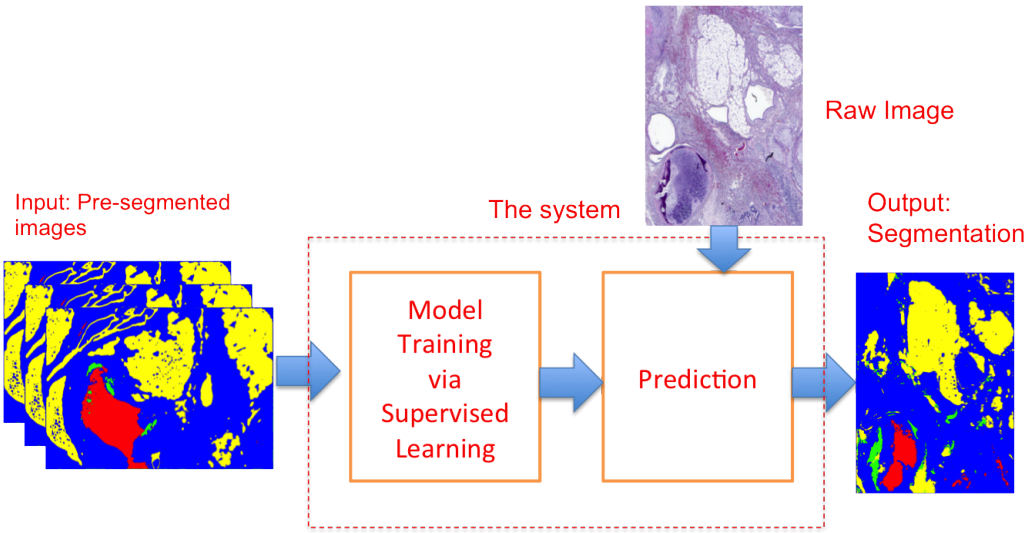
1.3 Motivation for the research

In the previous sections, we have seen a number of image segmentation techniques and their applications in biomedical imaging. While extensively researched, generally, current segmentation methods perform adequately in the applications for which they were designed. However, these methods still have a couple of drawbacks: a large number of algorithms successfully used in real applications were specifically designed for the given application. In our previous experience, before an algorithm that was specifically designed for one application can be used in another, a significant amount of tuning and calibration is usually required. Even then, in many cases, the chosen method may not perform satisfactorily. Therefore, a researcher faced with a new problem or application, must often spend considerable resources to modify (or develop a new) a reliable segmentation method capable of extracting the structures of interest for the given application.

Thus, we would like to propose a general purpose segmentation system, which can be applied across different biomedical applications with little tuning or calibration, and achieve comparable or even better results compared with the state-of-the-art methods specifically designed for each application. For the purpose of general application, the proposed system should be able to segment a variety of biological objects with different structures (e.g. tissues, nuclei, cells, etc.) from different imaging modalities (MRI, CT, histology, etc). In addition, the system can be used with two or three-dimensional data, as well as scalar or vector valued (e.g. color) images. If the proposed aim can be achieved, such a system can be utilized as a general segmentation tool, which can be widely used in a variety of biomedical applications.

Unlike many segmentation approaches, which aim to design specific mathematical models to describe the biological objects which are to be segmented, we choose the supervised learning

strategy as the basic framework, which makes the system general in the sense that, given sufficient training data, it can be used to segment a variety of biological structures, from different imaging modalities. In general, a user provides the system with a few hand labeled/segmented images, and the system proceeds by ”learning” a classification-based function that can be used to estimate the class of a given pixel (e.g. background vs. object). Once such a function is learned, it is then coupled with a region optimization algorithm (together with any geometric/shape bias terms available from the training data) for producing the segmentation of any unseen images. In Figure 1.1, we show a schematic organization for such a system.



(a)

Figure 1.1: Schematic organization for a general image segmentation system using supervised learning strategy

1.4 Innovations of the thesis

The need for robust automated image segmentation in modern quantitative imaging pipelines is clear. While a plethora of methods can be found in the current literature, it is often the case that these are specifically designed and tuned to a particular application. Before an algorithm that was specifically designed for one application can be used in another, a significant amount of tuning and calibration is usually required. Even then, in many cases, the chosen method may not perform satisfactorily. Thus, our major goal is to propose a system which can be used more generally, with little or no adaptation.

Our system [57, 62] aims to make use of the fact that often times a few segmented images (labeled pixel data) are already available or can be easily obtained, together with the ever increasing capabilities of modern computers, to arrive at a general segmentation tool that can be used in a variety of applications. To design such a general segmentation system, we consider choosing a supervised learning strategy where the pixels of a few segmented images are used to train a classifier capable of accurately determining the class (e.g. background vs organs/tissues of

interest) of each pixel in unlabeled images of the same kind. The idea of using supervised learning strategy to classify pixels for image segmentation is attractive because, in principle, given enough training samples, the strategy can be used to construct algorithms capable of performing accurately across different image modalities (computed tomography (CT), magnetic resonance (MR), microscopy, for example) and different structures of interest (organs, tissues, or cells).

The idea of pixel level classification for image segmentation is not new, and many existing algorithms, for example, thresholding methods, or unsupervised clustering methods, etc., can be regarded as pixel classification methods since the segmentation is implemented via labeling every pixel into different classes such as foreground versus background. However, to our best knowledge, no general purpose approach using such an idea for segmenting different structures (cells or tissue types) in different imaging modalities has been proposed and shown to work well across different applications, due to the fact that it is not an easy task to construct such a system successfully. We propose to address three main difficulties: (1) *Building the training set*: In its simplest form, all available pixels should be utilized as the training data (accounting for normalization), while different variations and scales of the same data (which are simulated as described below) will also need to be taken into account. This strategy, when applied to its full extent, would be computationally impractical. Since very often millions of pixels are available for training, it is clear that a training data selection approach must be taken. (2) *Designing a fast classification algorithm*: our current prototype uses existing classification algorithms, such as the support vector machine (SVM), for a pixel by pixel prediction (with large computational costs associated). Thus, developing a fast classification algorithm is necessary for the practical use of the system. (3) *Region optimization*: Usually, pixel level classification results are not good enough to be directly used as the final segmentation results, which have jagged borders/surfaces and holes inside. As a common requirement of segmentation results is to have smoothing borders/surfaces, the classification results must be refined before outputting.

In this dissertation, we propose a general segmentation system, which is able to solve the challenges mentioned above, and we have several innovations listed as follows:

A general image segmentation framework [57, 62]: Instead of utilizing parametric features as usually done, we choose instead to use intensity neighborhoods as a generic non-parametric feature, with the hope that it can be used more generally than texture features for example. In our work, we hypothesize that all the necessary information for classification, such as color, texture, structure, etc, is contained in the local neighborhood window, and thus, the intensity neighborhood feature has the potential capability to distinguish different tissue classes in general. There are some thoughts for designing such a system: (1) For a method to work well across several applications, it is important that the information present in the training set be appropriately utilized for the classifier training. This requires that the data be appropriately normalized and that all necessary information be taken into account. We note that the feature vectors, if directly comprised from square or cubic windows of intensity neighborhood, are not rotational invariant. In most biomedical imaging applications of interest, however, a fixed coordinate frame cannot be assumed and rotational variations must be accounted for. Therefore, we can virtually augment the set of training pixels to include rotated versions of each window. This is accomplished by rotating each example image using linear interpolation. In addition, we include flipped (coordinate reversed) versions of windows to increase useful information. Such an implementation is easy, however, may cause the problem of augmenting the data. An alternative way is to choose a

rotational invariant way for representing the intensity neighborhoods. (2) In addition, our prototype should also work across multiple scales. While for some types of tissues/cells, information for distinguishing a given object from others may be present in relatively small (local) neighborhoods, for others, the overall architecture (big picture) of the neighborhood is necessary to provide crucial information. Arguably this could be achieved by simply considering the size of the neighborhood (N) to be large enough and leave it to a classifier to determine the critical information for each class. However, large neighborhoods have the disadvantage that they amount to high dimensional spaces, making estimation and pixel classification difficult. In order to utilize small neighborhoods (e.g. N = 3 or 5) for maintaining reasonable computational costs while being able to capture the useful information for different classes, we utilize multiple windows of the same size, but each window corresponds to a scale (obtained through different levels of blurring and sub-sampling). More specifically, given scales $s \in 0, 1, 2, ..., S$, neighborhood patches are assembled by sub-sampling at every 2^s pixels (after smoothing). Therefore we can associate each pixel in the training set (as well as during actual classification of a test pixel) several sets of neighborhoods with the same size N but under different resolutions, comprising the multiple scales associated with that pixel. Details of designing such a framework will be discussed in Chapter 2.

An efficient data selection method for fast classification: As mentioned above, one big challenge in developing a practical segmentation system is how to deal with big amount of training data, which will cause a high computational cost for training classifiers. Thus, data selection is a necessary step here, which aims to extract a small subset that contains useful classification information from the initial large sized training set. Although a large number of data selection algorithms have been developed in data mining field, these existing ones cannot be applied directly to image data due to their high data reduction rates (defined as the ratio between the subset size and the whole set size), which still, generates a large sized subset and thus, makes computational cost for both training and testing processes large. In order to achieve a reasonable computational cost, the ideal case should be to extract a small sized subset, and train a simple classifier based on the given subset. We proposed a new majority voting based data selection criterion, based on which a fast segmentation scheme is utilized to segment images. Such a criterion is designed based on the theorem that once given a group of weak classifiers (defined to be a classifier which is only slightly correlated with the true classification, meaning that it can label examples better than random guessing), the classification accuracy by majority voting within the group increases monotonically with the size of the group as long as all weak classifiers' accuracies are higher than 50%. To be specific in image data, by utilizing the characteristics of spatial connectivity between neighboring pixels, it is reasonable for us to make an assumption that generally, all pixels within a local window in the images belong to the same class. Thus, we can hypothesize that it is very possible to increase and thus, achieve a high classification accuracy via majority voting spatially within local windows, even using a weak classifier derived from a small sized training set. In chapter 3, we will show that the proposed algorithm is able to achieve comparable or even better segmentation performance with a much lower computational cost compared to many state-of-the-art algorithms, which makes it practical to use.

A general region optimization model: A common requirement for the segmented regions is to have smooth borders/surfaces, however, the segmented regions, if obtained by directly grouping the pixels with the same predicted label into objects, are not always satisfactory, as

they usually have jagged borders with holes inside. Thus, it is necessary to optimize the classification result before the final output. Our goal is to develop a robust method for automatically delineating regions that encompass each single object in the image, and thus, we propose to use a more flexible approach based on deformable model. More precisely, in order to deal with the result from a pixel level classification procedure, our model takes the decision information from previous pixel classification process as input, and takes both region information and edge information into consideration for optimization. We note that the proposed model is general, which can be utilized after pixel classification processes in different types of experiments to optimize the segmented regions. In chapter 4, we will discuss the proposed model in details and show that it is able to provide much better segmented regions.

A general system for segmenting objects with a strong prior shape information [46, 97]: Specifically, we propose another general segmentation system for biomedical applications with a strong prior shape information available. Here we explain that a strong prior shape information means that the shapes/structures of objects to be segmented can be estimated statistically from the given samples, e.g. cell nuclei, human brain, bones, etc., while some examples of weak or no prior shape information include: tissue regions in histopathological images, where shape of tissue regions can not be estimated from the given samples. In this work, we focus on segmentation of cell nuclei. This task is very necessary and important for many scientific and clinical applications due to the fundamentally important role of nuclei in cellular processes and diseases. Thus, we describe a generic nuclear segmentation system based on the combination of template matching and supervised learning ideas. We propose a segmentation system that can be used effectively for segmenting nuclei for many different types of cells imaged under a variety of staining or fluorescence techniques. We aim to guarantee robust performance by allowing the method to calibrate itself automatically using training data, so that it will adapt itself to segmenting nuclei with different appearances (due to the staining techniques for example) and shapes. The method is also constrained to produce smooth borders. Finally, given that the objective function used in the segmentation process is the normalized cross correlation (NCC), the method is also able to better handle variations in illumination within the same image, as well as across images. We will describe this system in details in Chapter 5.

1.5 Overview of the thesis

The rest of the dissertation will be organized as follows:

Chapter 2 first investigates the idea of a general segmentation framework, where we use intensity neighborhood directly as a general feature, and use a pixel level classification process based on supervised learning to segment biomedical images. We validate such an idea by applying to different types of data, e.g. fluorescence images, histopathology images, MRI images, etc., and comparing with the state-of-the-art methods specially designed for these data. The qualitative and quantitative evaluations show that the proposed approach is able to provide comparable or even better performance across different types of datasets.

Chapter 3 continues the research in Chapter 2, which aims to solve the drawback of high computational cost mentioned in Chapter 2. In this chapter, we will discuss the data selection techniques, which extract a small subset that contains useful classification information from

the initial large sized training set. We proposed a new majority voting based data selection criterion, based on which a fast segmentation scheme is utilized to segment images. We test such an algorithm on the same data used in the previous chapter, and show that the proposed algorithm is able to achieve comparable or even better segmentation performance with a much lower computational cost, which makes it practical to use.

Chapter 4 aims to solve another drawback of unsmoothed segmented regions in Chapter 2, and describes an optimization step that follows the classification step, which is used to optimize the segmented region (each segmented region refers to a set of connected pixels with the same predicted label), such as smoothing the borders of segmented regions, filling the holes inside the regions, and removing noise, etc. In this chapter, we will discuss the existing various algorithms for optimizing the segmented regions, such as deformable models, graph cuts, etc., and then, propose a deformable model based on level set, which utilizes the information from classification step to help improving the quality of segmentation results. We show that the proposed deformable model can be utilized as a general approach for improving the segmentation performance.

Chapter 5 investigates some specific biomedical applications when strong priori shape information exists. Different from the research in previous chapters, which segments the regions of interests via predicting every single pixel, we focus on segmenting biomedical objects with a strong prior shape information available, such as nuclei, cells, etc. In this chapter, we propose a general system based on template matching, where templates are generated via statistical learning from a group of pre-selected samples, and objects are detected and segmented via maximizing the normalized cross correlation value between the target object and corresponding template. We show experimentally that our proposed approach is able to achieve comparable or even better results. We also mention that such a framework can be applied to higher dimensional data and structures.

Chapter 6 summarizes and concludes the whole research work in this dissertation. In addition, the future research work will be discussed.

Chapter 2

Pixel Level Classification: A General Framework For Biomedical Image Segmentation

2.1 Review of biomedical image segmentation methods

In the past few decades, we have witnessed a great increase in the development of new imaging modalities, and their applications to different biomedical research and clinical problems. Given the enormous success with which some of these technologies have been applied to research and clinical tasks, the trend of image technology development (and their application in novel biomedical problems) is likely to continue. Segmentation is of great importance in the application of imaging technology to many biomedical problems. Tissues, cells, and organs must often be segmented and isolated from two or three dimensional digital image data for subsequent quantitative analysis in a variety of experimental biological studies and diagnosis in clinical medicine. Popular examples include quantification of gray and white matter tissues from magnetic resonance imaging (MRI) brain scans for studying neurological diseases (e.g. Alzheimer's), segmentation of cells and tissues from histopathology image data to assist in the diagnosis of different lesions, segmentation of cells and subcellular structures for characterizing their distribution, to name a few.

Due to the vast increase in the capability of image acquisition during the past couple of decades, manual segmentation is no longer effective for many types of quantitative studies. The increase in computational power in recent decades has spurred the development of several image segmentation algorithms (see [34, 43, 70, 98, 99, 100, 101] for reviews on this topic) that have had a significant impact in several clinical and research applications. It is worth noting, however, that many algorithms successfully used in real applications were specifically designed for the given application. In our previous experience, before an algorithm that was specifically designed for one application can be used in another, a significant amount of tuning and calibration is usually required [34, 43, 99]. Even then, in many cases, the chosen method may not perform satisfactorily. Therefore a researcher faced with a new problem or application, must often spend considerable resources to modify (or develop anew) a reliable segmentation method capable of

extracting the structures of interest for the given application.

Among the several currently available alternatives for image segmentation, methods based on pixel classification utilizing learning based classification strategies are attractive because, in principle and given enough training samples, the strategy can be used to construct algorithms capable of performing accurately across different image modalities (computed tomography (CT), MRI, microscopy, for example) and different structures of interest (organs, tissues, or cells). Several algorithms based on this strategy have been described. We mention a few recent ones, focused on biomedical image segmentation applications for different modalities of data: Debeir et al. [102] used statistical features from color channels, together with a pixel classification by decision tree method, to segment pigmented skin lesions in epiluminescence microscopy (ELM) images. Madabhushi et al. [103] described a pixel level classification system that utilized 3D texture features to segment the malignant regions in MR prostatic images for diagnosis of prostatic adenocarcinoma. In [93], Bhagavatula et al. used histopathology vocabulary (HV) features and pixel classification strategy to identify germ-layer components in hematoxylin and eosin (H&E) stained histological images of teratomas. Considering that segmentation methods based on pixel classification are often computationally inefficient, Dam and Loog [104] proposed a general sparse pixel classification method. While reasonably high accuracies can be obtained with the methods mentioned above, to our best knowledge, no general purpose approach for segmenting different structures (cells or tissue types) in different imaging modalities has been proposed and shown to work well across different applications. We believe that, in part, this is due to the fact that different applications, structures, and imaging modalities require careful selection of relevant parametric features to work well.

Here we describe a system for biomedical image segmentation based on pixel classification using intensity neighborhoods [57, 62]. While intensity neighborhoods have been used to classify texture patches before (several works are discussed in the following section), we show here that such an approach (with adaptations) can be used as a general segmentation tool for biomedical images. The system is general in the sense that, given sufficient training data, it can be used to segment a variety of biological structures, from different imaging modalities. In addition, the algorithm can be used with two or three-dimensional data, as well as scalar or vector valued (e.g. color) images. Our algorithm requires as input a few already segmented images/structures (often obtained manually) with which a classifier is trained. In order to arrive at a generic segmentation/classification method, with each pixel in the training set we associate a nonparametric feature vector consisting of the neighborhood intensities for building the classification-based segmentation system. The system aims to maximize the available training data by exploring several orientations as well as scales for each labeled pixel. We compare the performance of our general image segmentation method to several state-of-the-art methods for segmenting brain MR images, histopathological images, and fluorescence images of nuclei. We show that, using the methodology we describe in detail below, such a generic segmentation system can achieve accurate segmentation results in a variety of applications.

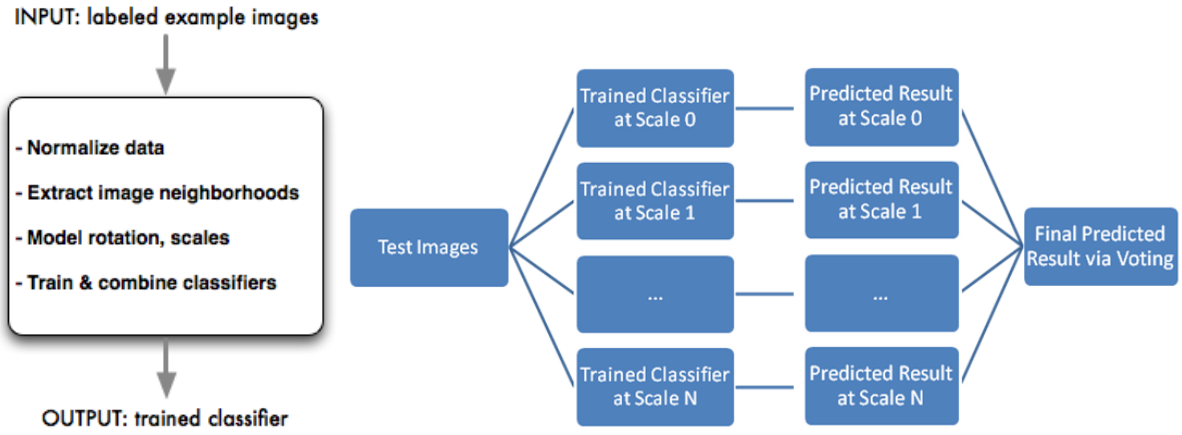
2.2 Methodology

Our algorithm aims to make use of the fact that often times a few segmented images (labeled pixel data) are already available or can be easily obtained, together with the ever increasing capabilities of modern computers, to arrive at a general segmentation tool that can be used in a variety of applications. This can be achieved within a supervised learning strategy where the pixels of a few segmented images are used to train a classifier capable of accurately determining the class (e.g. background vs organs/tissues of interest) of each pixel in unlabeled images of the same kind. For the method to work well across several applications, it is important that the information present in the training set be appropriately used during training of the classifier. By imposing a pre-defined size window centered at the given pixel, in which pixels inside the window are regarded as neighbors, the intensity neighborhood vectors are thus constructed by reordering the pixels' intensities inside the window into a vector and used instead of parametric feature vectors in order to allow the method to be general. This, however, requires that the data be appropriately normalized and that different variations such as scales, rotations, etc. be taken into account. In its simplest form, such a classifier would need to be trained utilizing all available pixels as training data (accounting for normalization), while different variations and scales of the same data (which are simulated as described below) will also need to be taken into account. This strategy, when applied to its full extent, would be computationally impractical. For example, the support vector machine classifier we select to use in our implementation has a computational cost of order P^3 (upper bound) [52], with P the number of pixels. Since very often millions of pixels are available for training, it is clear a different approach must be taken.

In this chapter, we propose an empirical approach for reducing the amount of training data, while retaining important information that allows one to differentiate between pixel classes, thus allowing for execution in reasonable computation time. The strategy is conceptually simple (it is summarized in Figure. 2.1), and we show it works well in several practical applications. In our algorithm, the training stage occurs by first normalizing the input data, extracting important pixels from the input data, modeling rotation and scalings of the data via filtering and resampling, and training the actual classifiers at different scales (Figure. 2.1(a)). In the testing stage, the final result is given through a voting procedure for combining the predictions under different scales (shown in Figure. 2.1(b)). We also note that several of the sub-steps in our procedure (e.g. multiscale filters, classification method, etc.) may be replaced by other options, in the presence of more detailed application specific information.

2.2.1 Intensity normalization

Given an application (e.g. segmentation of tissues in histopathological images), it is important to account for variations in intensity (e.g. due to images being taken with different microscopes under different illumination environments, etc.) that may be present from image to image. In MRI, for example, intensity normalization for correction of different bias fields is an extensively studied subject [75, 105]. The calibration for intensity in histopathological images, or fluorescent images can also be performed [95], in limited controlled settings, however. Since our aim is to design a system for general purposes, we do not rely on intensity corrections specifically designed for any single imaging modality (or application) but rather take the general approach of



(a) (b)

Figure 2.1: System schematic organization for neighborhood-based image segmentation using supervised learning (a) training stage (b) testing stage

normalizing all image data to fit the intensity range of $[0, 1]$ by scaling the minimum and maximum of each image (discounting outliers set at 1% in our implementation). We note, however, that given more detailed imaging/application specific information, this method can be replaced by other alternatives.

2.2.2 Intensity neighborhood statistics: what information should be considered?

Instead of parametric features (e.g. Haralick texture features, Gabor filters, etc.), we select each pixel's neighborhood intensities as a general non-parametric feature set for classification. When a large enough window is chosen, the advantage of such an approach is that all information relevant for differentiating different classes (tissues, cells, etc.) will be preserved. The disadvantages are that, especially if large neighborhoods are necessary, the method will involve classification in high dimensional spaces, often making accurate estimations difficult. The idea of utilizing window neighborhoods is not novel, and has been used in numerous applications in image processing. For example, Varma et al.[106] describe a classification method based on pixels' neighborhood intensities. They showed the method compared favorably to the more traditional parametric filter-bank method on an application related to texture segmentation. Awate et al. also present the neighborhood based method for texture segmentation[107] and brain-MR segmentation [76]. In addition, similar neighborhood ideas have been used in image filtering and denoising fields recently. Buades et al. introduced the non-local means (NLM) method[108] for image denoising, in which similarities of image neighborhood are calculated as weights for averaging pixel intensities. In [76], this neighborhood weighted averaging method is interpreted statistically. In order to reduce the computational cost caused by the high dimensionality of neighborhood in NLM method, principal component analysis (PCA) of neighborhood is used in NLM method[109]. We note, however, that although neighborhood-based image segmentation methods have been described in the past, to our knowledge, no general algorithm has been

proposed, validated, and compared to state of the art techniques in several different biomedical image segmentation problems (including different imaging modalities). Here we describe several adaptations and modifications of the non parametric intensity neighborhood idea to arrive at a generic image segmentation system capable of performing well across a variety of biomedical applications.

In our implementation we utilize a $N \times N$ square patch for 2D data (and a $N \times N \times N$ cube patch for 3D data), to comprise the N^2 (or N^3) non-parametric feature vector associated with the pixel at the center of the patch (N is chosen to be an odd number in our application) (Figure 2.2). Since the image patches are not rotationally invariant, the training set as described could potentially be sub-optimal since it does not include rotated versions of the feature vector while for most biomedical applications of interest, a fixed coordinate frame for the tissues/cells of interest cannot be assumed. To overcome this limitation, we synthetically augment the training set of images by including image patches that are also rotated (about multiple axes if necessary). Image rotation is computed using linear interpolation in our case. Consideration of computational complexity limits us to utilize only a limited number of angles (described in detail below for each application), however. In addition, we include flipped (coordinate reversed) versions of windows to increase more useful information. In order to further reduce the number of potential training pixels, we also make the assumption that it is more important to include variations of patches from pixels near the boundaries between two or more classes (e.g. tissues) than interior pixels, allowing us to consider only multiple rotated and flipped versions of boundary-type pixels' neighborhood patches in our training set, reducing the computational complexity of the procedure. A general way to decide the target pixel's type is to see whether a square window centered at the target pixel includes more than two classes. If so, such pixel will be judged as boundary type pixel, otherwise, it will be judged as interior type pixel. In Figure 2.3, we illustrate examples of both boundary type pixel and interior type pixel. Figure 2.3(a) shows the original image, in which the boundary type pixel is represented by a red dot, and interior type pixel is represented by a blue dot. For better understanding, we also show two square windows centered at two pixels. In Figure 2.3(b) and (c), we zoom in the window to show clear tissue appearances around the pixels. Clearly, we can see that the square window patch of boundary type pixel contains two different tissue textures, while the square window of interior type pixel just contains one type of tissue.

Finally, it is also important to consider multiscale information. While for some types of tissues/cells, information for distinguishing a given object from others may be present in relatively small (local) neighborhoods, for others, the overall architecture (big picture) of the neighborhood is necessary to provide crucial information. More generally, we believe information from multiple scales should be used to characterize each class in each application. Arguably this could be achieved by simply considering the size of the neighborhood (N) to be large enough and leave it to a classifier to determine the critical information for each class. However, large neighborhoods have the disadvantage that they amount to high dimensional spaces, making estimation and pixel classification difficult. In order to utilize small neighborhoods (e.g. $N = 3$ or 5) while being able to capture the useful information for different classes, we use a standard multi-resolution sequence[110]. In short, N remains constant for each scale. At each scale, images belonging to the training set are first convolved with a Gaussian kernel for smoothing, and sampled (according to the scale chosen) to construct the neighborhood set associated with a particular scale. More

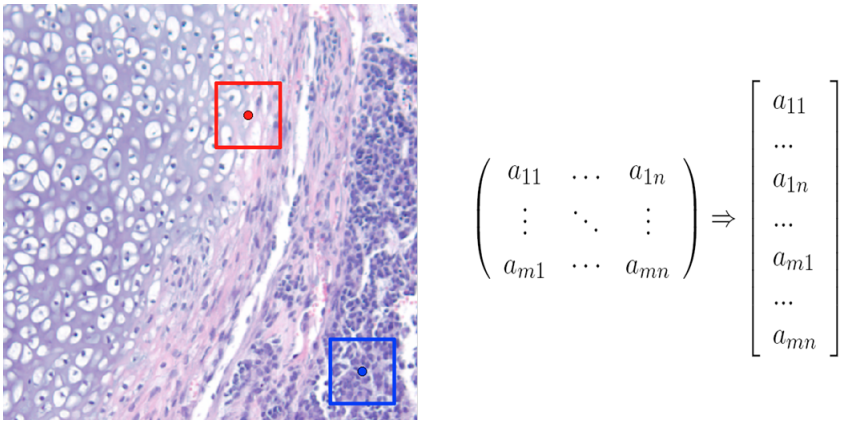


Figure 2.2: Representing an intensity neighborhood into a feature vector

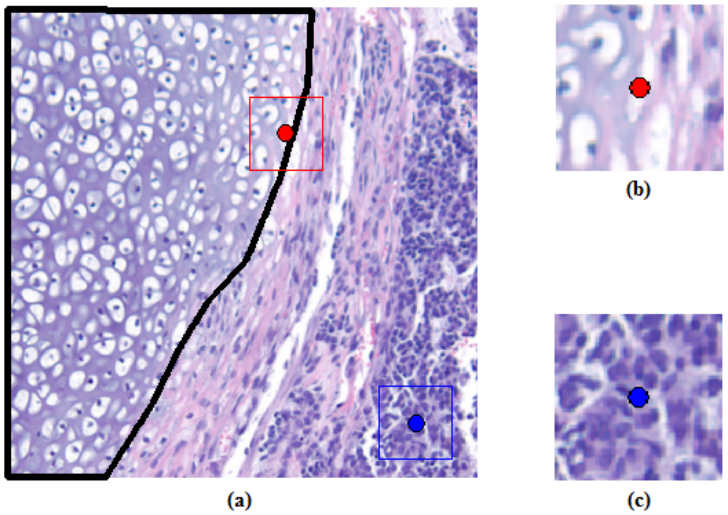


Figure 2.3: Examples of boundary type pixel and interior type pixel. (a) original image with two pixels marked. (b) magnified window of boundary type pixel. (c) magnified window of interior type pixel. Black line indicates boundary between tissue types.

specifically, given scales $s \in \{0, 1, 2, ..., S\}$, neighborhood patches are assembled by subsampling at every 2^s pixels (after smoothing). Therefore we can associate each pixel in the training set (as well as during actual classification of a test pixel) several sets of neighborhoods with the same size N but under different resolutions, comprising the multiple scales associated with that pixel. In Figure 2.4, we show the case of extracting the intensity neighborhood under scale $s = 1$.

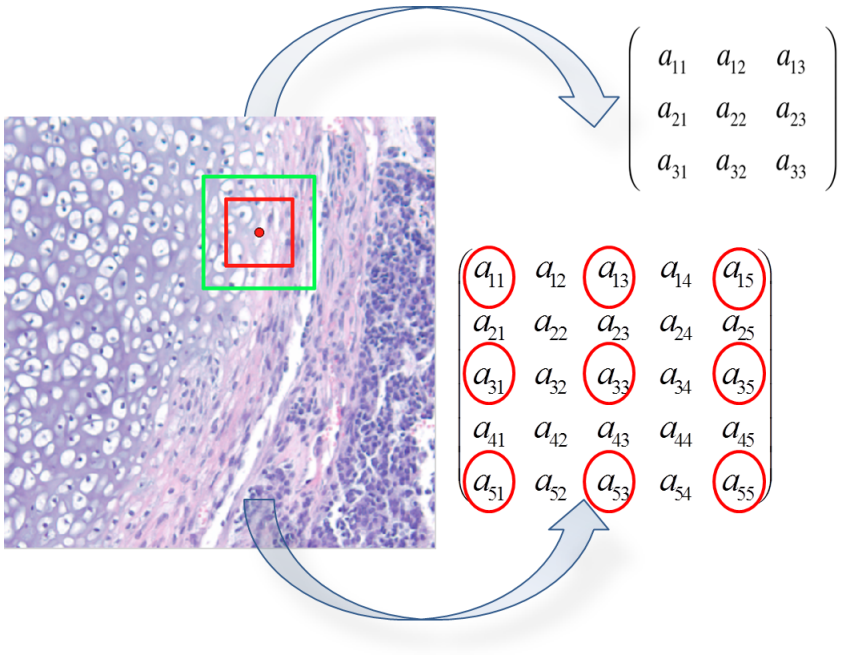


Figure 2.4: Intensity neighorhood under multi-scale

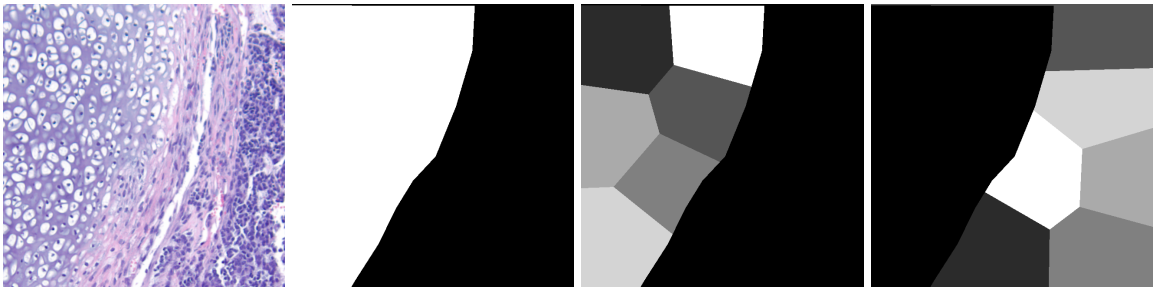
2.2.3 Algorithm for selecting representative pixel neighborhoods

Our system works by utilizing a set of pixels from pre-segmented images to train a classifier and then segmenting new unseen images. For many applications however, even relatively few images (when the variations in rotation and scale are included) can contain several million pixels in the training samples. Given the computational complexity of pertinent classifiers, which generally ranges from $O(P^2)$ to $O(P^3)$, with P the number of training samples[52, 98], simply utilizing all available pixels for training is not a practical strategy. Given the maximum number of training samples (denoted as Q and pre-selected in order to satisfy computational cost considerations) to be utilized for training, one alternative for selecting such Q pixels would be to do so at random. Here, we describe an alternative procedure based on the K-means algorithm for selecting a set of training pixels that retains the main "trends" of the information contained in the training pixels.

We note that since the computational complexity of the K-means procedure is also of $O(P^2)$, its straight forward application is difficult. We therefore resort to the following simplification. We first divide the P training samples into different subsets (in this work, the subsets are divided according to their spatial location in the image). We then apply the K-means method to each

subset separately and combine all the clustered samples into a pre-defined training set of size Q samples. The procedure can be summarized as follows:

1. We use the K-means method to cluster the pixels' spatial coordinates, which divides the input training images into R non-overlapping spatial regions, with R selected manually (see Figure 2.5).
2. Each of the pixels in the regions defined by step 1 are categorized as either a boundary-type pixel or an interior-type pixel. In each region, the K-means procedure is used to select only certain pixels (and their windows) for training (the amount selected is calculated based on R and Q). This is done for both boundary (including rotated and flipped windows) and interior type pixels, and is repeated for each scale s.
3. Finally, for each scale s, all the clustered samples for both boundary-type and interior-type of all subsets and all classes are combined together as the training set containing Q samples and thereby, several training sets associating with different scales $s \in \{0, 1, ..., S\}$ can be built.



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

Figure 2.5: Examples of spatial sample separation (a) original image (b) ground truth (c) spatial sample separation in interested tissue (d) spatial sample separation in background/other tissues

2.2.4 Algorithm for reducing the dimensionality of the training set

In addition to the procedure for reducing the number of training pixels explained above, we also investigate the reduction of feature dimensions and its related effect. In this work, we utilize a PCA-based procedure to reduce the size of each feature vector (which in our case is composed of intensity neighborhoods). Since neighborhood feature vector is typically high dimensional, for example, for a $N \times N \times N$ neighborhood, the dimensionality becomes N^3, which will increase greatly when the neighborhood size increases. Thus, we consider reducing the dimensionality of vectors by first converting the intensity neighborhood vector to a feature vector containing uncorrelated variables via principal component analysis (PCA) [111] and then discard useless variables for classification. Based on an existing training set containing Q samples of N^3 dimensions, PCA method is implemented in a standard procedure in which a covariance matrix is built from the samples and the eigenvectors of the covariance matrix are calculated and sorted in order of descending eigenvalues; then, for each sample in the training set, the dot product between each

eigenvector and that pixel's neighborhood vector is calculated as the coefficient in this sample's PCA based feature vector. Then, stepwise discriminant analysis (SDA) [111] is used for automated determination of subspace dimensionality and selection of the principal components from the PCA feature vectors. The SDA method chooses the variables from the feature vector which maximize the ratio of the variance between the classes to the variance within the classes. Thus, we are able to produce a new training set with the same samples but associating with a lower dimensional PCA based subspace feature vector by selecting the necessary principal components from the PCA based feature vector using SDA method.

Additionally, we note that the computational complexity of actual classification (testing phase) using Support Vector Machine (SVM) with a non-linear kernel such as RBF kernel is $O(nd)$, where n denotes the number of support vectors, and d the feature dimensions [112]. Thus, based on the same training set but with a lower dimension (PCA vector), computational efficiency can be improved.

2.2.5 Algorithm for classification

We utilize the support vector machine (SVM) classifier [52, 113] as the algorithm for classifying each pixel. In a two class problem, given a training set containing Q samples (each with d dimensions), we denote the feature-label pairs as $(X_i, Y_i), i = 1, \ldots, Q$, where $X_i \in \mathbf{R}^d$, $Y_i \in \{-1, +1\}$. The support vector machine determines parameters w and b such that

$$\arg\min_{w,b,\xi} \left\{ \frac{1}{2} w^T w + \varphi \sum_{i=1}^{Q} \xi_i \right\} \tag{2.1}$$

$$subject\ to:\ Y_i(w^T \phi(X_i) + b) \geq 1 - \xi_i, \xi_i \geq 0 \tag{2.2}$$

is minimized, yielding a hyperplane (defined by w and b) that linearly separates the data. Since the dataset is not always linearly separable, the ξ_i represents the distance of each error point i to its correct plane, and φ is a penalty constant for the error term. ϕ is a fixed non-linear mapping function (known as basis function) that extends training vectors X_i into higher dimensional space $\phi(X) : \mathbf{R}^d \mapsto \mathbf{R}^m$. This problem is usually solved in its dual representation, where the data always occur in pairs, with the aid of the kernel function [113, 114] $K(X_i, X_j) = \phi(X_i)^T \phi(X_j)$. Common kernel functions include: the linear kernel, the Gaussian radial basis function (RBF) kernel, polynomial kernel, sigmoid kernel, etc. In our work, the RBF kernel $K(X_i, X_j) = exp(-\gamma \|X_i - X_j\|^2), \gamma \geq 0$ was selected for all applications, because such kernel is able to handle the case when the relation between class labels and attributes is nonlinear by nonlinearly mapping samples into a higher dimensional space. For problems with more than two classes, we use the "one-against-one" classification strategy [115] to reduce the single multi-class problem into multiple binary problems and use a max-wins voting strategy to combine these binary results and classify the testing instance. The classification system was implemented utilizing the LibSVM software [116] package.

The optimal parameters, such as the penalty constant φ and the RBF kernel size γ, are selected using a cross validation procedure[116]. We use k-fold cross validation to further separate the training set into two parts (lower-level), and search for the parameters (φ, γ), which have the best

accuracy in this k-fold cross validation. We set $k = 10$, and perform an exhaustive search for the two parameters: firstly, we search (for (φ, γ)) broadly using a large step size in the range which could be considered reasonable (determined empirically). If any optimal parameters are selected on their lower (or upper) bounds of the ranges, we decrease (or increase) the corresponding bounds by 5 times, and repeat the rough (broad) search. If the optimal parameters are selected inside the ranges, we then select smaller ranges around the optimal parameters, and choose a smaller step size to find the final optimal parameters locally. The ranges (or the upper bounds) are first determined empirically. If any optimal parameters are selected on their lower (or upper) bounds of the ranges, we decrease (or increase) the corresponding bounds by 5 times, and repeat the rough search. More details of the descriptions can be seen in [117]. After the optimal parameters are selected, we use them to build the classifiers and evaluate their performance on the testing data.

2.2.6 Multi-scale classifier ensemble: how to integrate information from individual classifiers?

As we stated above, we associate each pixel in the training set of images several neighborhoods each containing different scale information (scales $s \in \{0, 1, 2, ..., S\}$ defined above). Instead of combining all scales into a single high dimensional vector (an option we investigated but did not yield adequate segmentation results presumably due to aforementioned difficulties with dealing with high dimensional data), we opt instead to train S classifiers and combine them so that the final overall accuracy is higher than that obtained from any single scale. We note that several methods for combining multiple classifiers exist [118, 119, 120, 121, 122] some of which have been applied to the problem of pixel classification using parametric features (see for example [123, 124]). Amongst the most simple, are voting-based methods. In our system we test two popular voting strategies: majority voting, and confidence-based voting.

(1) **Majority voting**: Given C classes of tissues to be segmented, we train S (number of scales chosen) individual classifiers separately. For a given pixel i, its intensity neighborhood vector at the scale s, $s \in \{0, 1, 2, ..., S\}$, is denoted as x_i^s, and the prediction label at the scale s is denoted as l_i^s. In the majority voting algorithm the final label l_i^{final} (class assignment for pixel i) is given as:

$$l_i^{final} = \underset{c \in \{1,2,...,C\}}{\arg\max} \sum_{s=0}^{S} \delta(l_i^s, c) \tag{2.3}$$

where $\delta(i, j)$ is defined as $\delta(i, j) = \begin{cases} 1, i = j \\ 0, i \neq j \end{cases}$.

(2) **Confidence-based voting**: In the confidence voting algorithm, classifiers are also trained separately for each scale s. After the training procedure, it is often possible to assign a confidence to the assignment of any pixel, denoted as $F(l_i^s = c)$. In our case, we utilize the posterior probability in SVM: $P(l_i^s = c|\mathbf{x}_i)$ ($\mathbf{x}_i$ is the intensity neighborhood vector for the given pixel i) as the confidence measurement, although other alternative confidence estimates can be used. For multiple classes, the posterior probability in SVM can be estimated by combining all the pairwise class probabilities [125]. The confidence-based voting strategy is then implemented as:

$$l_i^{final} = \underset{c\in\{1,2,...,C\}}{\arg\max} \sum_{s=0}^{S} \lambda_s F(l_i^s = c). \tag{2.4}$$

We test and compare two confidence-based voting strategies: weighted and unweighed confidence voting. In the equation above λ_s are the weights for each individual classifiers.

When unweighted voting is used, all weights are set to 1. When weighted confidence-based voting is utilized, the weights are calculated by choosing the ones that maximize overall classification accuracy in the training dataset, in a cross validation procedure described in [52]. Mathematically, for a given group of weights $W = \{\lambda_0, \lambda_1, ..., \lambda_S\}$, the labeled (segmented) training data is denoted as $R_{train}(W)$, and the ground truth of training data is denoted as G_{train}. Our goal is to find the optima set of weights $W_{optimal}$ that maximizes the objective classification accuracy function M as:

$$W_{optimal} = \underset{W\in R^{S+1}}{\arg\max}\, M(R_{train}(W), G_{train}) \tag{2.5}$$

where $M(\cdot,\cdot)$ is a measurement of overall classification accuracy. In this work, the overall classification accuracy is defined as $A_{correct}/A_{all}$, where $A_{correct}$ is the number of correctly classified pixels, and A_{all} is the total number of pixel to be classified, both of which can be calculated easily when $R_{train}(W)$ and G_{train} are given.

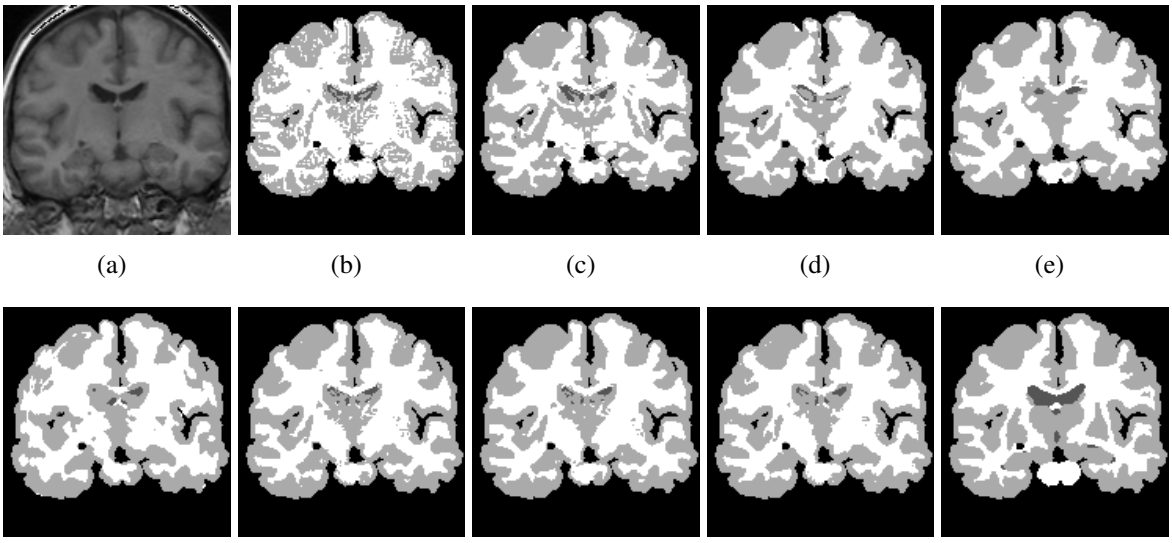
2.3 Experimental results

In this section, we describe both qualitative and quantitative evaluation of the performance of our classification system in three different datasets. We first compare the performances of several voting strategies, as mentioned above. We then test our system on several segmentation tasks and compare the results to those produced by several algorithms selected (and designed) for each application.

With the exception of the aforementioned classification training parameters optimized using cross validation, the parameters pertaining to our algorithm were selected considering the limitations of the available computing power. With the exception of the training set size, all parameters are kept constant throughout all computations in this work. We set the neighborhood size as $3 \times 3 \times 3$. In addition, for boundary-type pixels, image windows were rotated along the Z axis both clockwise and counter-clockwise by 45 degrees. Also, these image windows (including rotated versions) were flipped left and right, up and down in X-Y plane. For the implementation of multi-scale framework, a total of 5 scales (denoted scale 0 to scale 4) were used.

2.3.1 Voting strategy comparison

In this subsection, we compare the performances of the voting strategies discussed above. For this experimental evaluation, we chose the IBSR real T1-weighted brain-MR dataset [126], from which it is possible to obtain already segmented (ground truth) tissues such as cerebrospinal fluid (CSF), gray matter (GM) and white matter (WM). In the experiments, four 3D brain images were



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

(f) (g) (h) (i) (j)

Figure 2.6: Performances comparison of brain under different cases: (a) original image; (b) result at scale 0; (c) result at scale 1; (d) result at scale 2; (e) result at scale 3; (f) result at scale 4; (g) result of majority voting; (h) result of unweighted confidence voting; (i) result of weighted confidence voting; (j) ground truth

randomly selected as the training set and one brain image was randomly selected from the rest for testing. In this computation, the training set size contains 1.6×10^5 pixels.

Figure. 2.6 shows the segmentation result for one brain. In this figure, the dark gray color represents cerebrospinal fluid (CSF) tissue, light gray represents gray matter (GM) tissue, and white represents white matter (WM) tissue. Part (a) shows the original image, while parts (b-f) show the segmentation results by individual classifiers at different scales from 0-4. Parts (g-i) show the segmentation results by majority voting (MV), unweighted confidence voting (UCV), and weighted confidence voting (WCV) methods, respectively, while part (j) shows the ground truth information available. Visual interpretation shows that the classification results produced by each individual scale are inferior to results produced by the voting strategies. It is, however, difficult to discern visually which of the voting strategies performs best.

We compare the results quantitatively by using both classification accuracy as well as the dice metric [127]. The dice metric is defined as:

$$2|\widehat{T}_c \cap \widetilde{T}_c| \Big/ (|\widehat{T}_c| + |\widetilde{T}_c|) \tag{2.6}$$

where $\widehat{T}_c$ denotes the segmented region of pixels for tissue c, and $\widetilde{T}_c$ denotes the ground truth set of pixels for tissue c, and $|\cdot|$ denotes the set size. As with previous works [75, 76, 128, 129], we focus on segmentation of GM and WM tissues. The overall classification accuracy for each scale as well as for the various voting strategies is shown in Table 2.1. From this table it is clear that the weighted confidence voting method is able to achieve higher accuracy than individual classifiers or any other voting methods (on this dataset). We have also performed the same comparison on

a histology dataset (described in more detail below) and results were similar. These are omitted here in the interest of brevity. We have chosen to use the weighted confidence voting scheme in the overall assessment of our segmentation algorithm below.

Table 2.1: Quantitative Evaluation for Brain MR-data

Case	Scale 0	Scale 1	Scale 2	Scale 3	Scale 4	MV	UCV	WCV
Overall Accuracy	81.99%	84.77%	84.48%	85.05%	84.09%	88.45%	88.83%	89.11%
GM (dice metric)	0.8411	0.8709	0.8681	0.8744	0.8666	0.9033	0.9070	0.9092
WM (dice metric)	0.8055	0.8277	0.8274	0.8338	0.8161	0.8723	0.8712	0.8806

2.3.2 Segmentation of tissues in brain MRI datasets

As mentioned above, segmentation of gray matter tissue (GM) and white matter tissue (WM) from brain-MR images is a popular procedure in biomedical imaging, and several methods have been proposed to this end [75, 76, 128, 129]. We test our system on this application. As above, we use the same brain-MR dataset provided in [126], in which a total of 18 3D MRIs of different brains exist. Each 3D image contains 128 scanned slices. In our test, we use the same 4 brain images selected in the previous section for training, and the remaining 14 brain images are used for testing. The training set contained 1.6×10^5 pixels. To better understand the quality of the results produced by our method, we compare it to the results produced by the method presented in [76], which also utilizes the pixels' intensity neighborhoods as an adaptive nonparametric model of Markov statistics, and produces an optimal classification by iteratively maximizing a mutual-information metric that relies on Markov probability density function.

As in [76], we calculate the mean, median and standard deviation of the dice metric for the WM and GM tissue classes. The results of our method (as well as a summary of the results from [76]) are provided in Tabel 2.2. In short, the results indicate that, in general, the accuracies on the target tissues (GM and WM) are similar for both methods.

Table 2.2: Comparison: Mean, Median and Standard Deviation for GM and WM Tissues using Different Methods

Statistical Measure	GM (proposed system/[76])	WM (proposed system/[76])
Mean	0.9053/0.8074	0.8198/0.8868
Median	0.9092/0.8009	0.8382/0.8913
Standard deviation	0.0304/0.0426	0.0676/0.0179

2.3.3 Segmentation of tissues in histology images

Here we demonstrate the application of our system to the task of segmenting tissues from histology (H&E stained) images of teratoma derived from human and nonhuman primate embryonic

Table 2.3: Quantitative Evaluation for Images of Teratoma Histology

Statistical Measure	Bone	Cartilage	Fat	Background/Others
Our Method (Accuracy)	59.70%	73.18%	91.09%	88.93%
Color K-means (Accuracy)	29.79%	51.06%	58.73%	55.20%
Color SVM (Accuracy)	27.87%	29.09%	66.16%	68.12%

stem cells [84]. Generally speaking, tissue segmentation from histology images of this type is a challenging task due to the complex variation in texture, color, shape, structure, etc. of the tissues of interest. In addition, teratoma-specific challenges include low intra-class similarity (the same type of tissue often has different visual appearance) and high inter-class similarity (multiple types of tissues can have similar visual appearance) [85, 86, 130].

To the best of our knowledge, no standard approach for segmenting such images reliably and accurately is known. Since color is an important information in the segmentation of histological images, we test the K-means-based color segmentation approach, which has been used in the past to segment similar images [131, 132, 133]. The approach usually taken is to convert the color image from $R * G * B$ space to $L * a * b$ space, and then to use the K-means method to cluster pixels in $a * b$ color space. Here we compare our segmentation results to the method described in [131, 132, 133]. In order to make the comparison fair, we also utilize the color $a * b$ vector together with the same SVM method we use in our system, for the purpose of comparing the performances of different features.

The images used in this portion of our validation study were acquired by using a slide scanner to scan the whole sectioned teratoma slices at a high resolution after H&E staining. The color images obtained contain $R * G * B$ channels, and the resolutions are 3.527 microns/pixel (0.2834646 pixels/micron) and approximately 3-5 micron thickness (z-dimension). Since the number of pixels in raw images is usually large (e.g. size 4824×4014), image patches of smaller sizes are randomly cropped (e.g. size from 896×932 to 1438×1106) for simplification. In this test, 4 images were selected randomly for training, and the remaining 10 images were used for testing. The ground truth segmentations (used in both training and validation) were provided by the pathologist (J.A.O). As before, a 3×3 window was chosen for each resolution and the size of training set was chosen to be 1.2×10^5 pixels. In these tests, however, each neighborhood also contained 3 channels corresponding to the $R * G * B$ channels of each image. Figure 2.7 shows the results of automatic segmentation of four tissue classes: Bone (B), Cartilage (C), Fat (F), and background/other tissues (O). For the approach we proposed, together with the color K-means approach and color SVM approach, the green color represents B tissue, red color represents C tissue, yellow color represents F tissue, and blue color represents O tissue regions.

In this application, color represents useful information for classification, although it is clear that a method relying purely on color information cannot perform well in this application since most tissues are heterogeneous in terms of color content. The method we propose, on the other hand, is able to perform reasonably well. To quantify the result, Table 3.2 reports the overall pixel classification accuracy (a metric also used for quantitative assessment of segmentation of histology images in [93, 103, 134]).

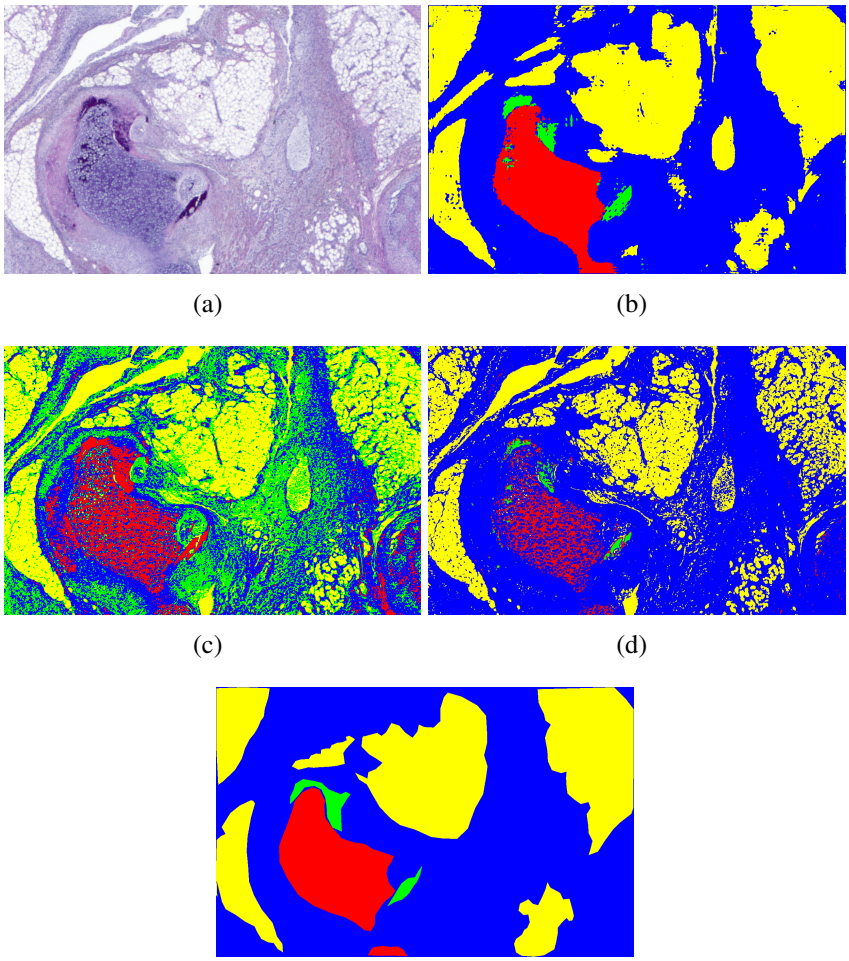
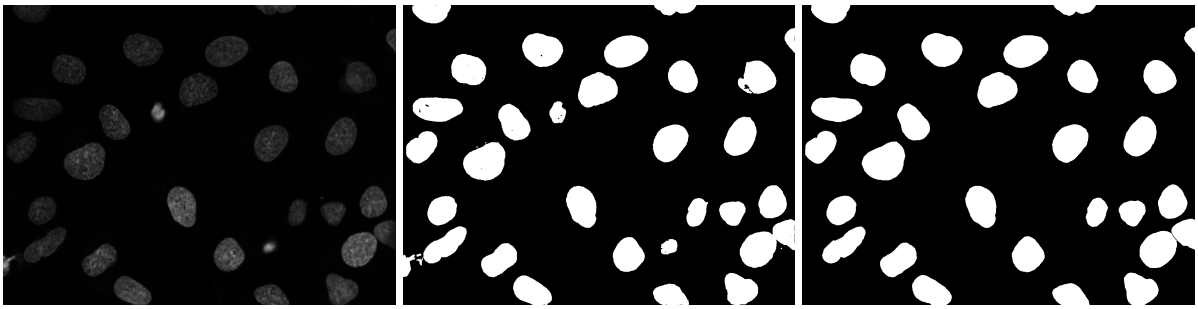


Figure 2.7: Segmentation of tissues for images of teratoma histology. (a) original image (b) our result (c) color K-means' result (d) color SVM's result (e) ground truth

2.3.4 Segmentation of cell nuclei from fluorescence microscope images

We test our approach in the task of segmenting nuclei from fluorescence microscope images. The image dataset chosen in this application is available from Dr. Robert Murphy's group at Carnegie Mellon University [135]. The dataset consists of 48 images, out of which 6 images were randomly selected for training and the remaining images were used for testing. We chose the neighborhood window size to be 3×3 and the training set size as 8.0×10^4 pixels here. In Figure 2.8 part (a) a sample image is shown. In part (b) the segmentation result produced by our system and in part (c) the ground truth image (provided by human observers as described in [135]). Visually the results seem satisfactory, although one may notice small openings in the interior of some nuclei. Such relatively minor artifacts could be easily removed using simple morphological operations (e.g. closing and opening). However, we have purposely refrained from using any post processing methods on our results throughout all examples shown in this chapter since our main focus is in describing our system and evaluating its performance.



(a) (b) (c)

Figure 2.8: Segmentation of nuclei (a) original image (b) our result (c) ground truth

Reference [135] reviews and compares several methods that are commonly used for nuclei segmentation. These methods include: (1) three thresholding based methods: Ridler-Calvard [136], Otsu [1], and mean pixel value; (2) seeded watershed method [100, 101] operating on a blurred version of the image and the gradient of the image; (3) active masks [137]; (4) a merging algorithm [16]. Several metrics are used for quantitative evaluation: (1) The Rand and Jaccard Indices: the Rand index (RI) measures the fraction of the pixel pairs where the segmented nuclei and the ground truth agree, which ranges from 0 to 1, which 1 means the perfect agreement. The Jaccard index (JI) is another metric for measuring the fraction of the pixel pairs agreed between the segmented ones and the ground truth. There is no upper bound for the Jaccard index. For both RI and JI metrics, the higher values mean the better segmentation. (2) Spatially-Aware Evaluation Metrics: both the Hausdorff metric and the normalized sum of distances (NSD) metric evaluate the segmented results spatially. For each pixel in the segmented image, its distance to the reference border can be calculated. The Hausdorff metric refers to the largest distance among the pixels which have a disagreement between the segmented objects and the ground truth, and NSD metric refers to the normalized sum of distances over these pixels. Note that for NSD metric, 0 means perfect agreement and 1 means no overlap. (3) Error Counting: Errors in the segmentation result are counted by comparing each segmented object with the referenced object in the ground truth with which it shares the most pixels. Four classes of errors are counted. Split: two segmented nuclei correspond to the same reference nucleus in the ground truth; merged: two reference nuclei in the ground truth correspond to the same segmented nucleus; added: a segmented nucleus corresponds to the background in the ground truth; missing: a reference nucleus correspond to the background in the segmented image. More details on these metrics can be seen in [135].

For validation, we compare our results of segmenting the nuclei on the dataset mentioned above to the methods reviewed in [135] using the same metrics, and the quantitative results are shown in Table 5.2. From the comparison, we can see that although the Hausdorff distance metric and the added error metric of our method is much larger than those of other methods due to the noise effect, which can be easily improved by using some simple morphological operations, the overall results are comparable or better to the results in [135].

Table 2.4: Quantitative Comparison of Nuclei Segmentation

Algorithm	RI	JI	Hausdorff	NSD($\times 10$)	Split	Merged	Added	Missing
AS Manual	95%	2.4	9.7	0.5	1.6	1.0	0.8	2.2
RC Threshold	92%	2.2	34.8	1.2	1.1	2.4	0.3	5.5
Otsu Threshold	92%	2.2	34.9	1.2	1.1	2.4	0.3	5.6
Mean Threshold	96%	2.2	26.5	1.0	1.3	3.4	0.9	3.6
Watershed (direct)	91%	1.9	34.9	3.6	13.8	1.2	2.0	3.0
Watershed (gradient)	90%	1.8	34.6	3.0	7.7	2.0	2.0	2.9
Active Masks	87%	2.1	148.3	5.5	10.5	2.1	0.4	10.8
Merging Algorithm	96%	2.2	12.9	0.7	1.8	2.1	1.0	3.3
Our Result	97%	2.5	119.3	0.8	0.8	2.8	3.5	0.3

2.3.5 Results comparison between non-PCA method and with PCA method

Figure 2.9 contains raw unprocessed images (a,e,i), ground truth (d,h,l), and results of using our segmentation method without dimension reduction (c,g,k) and with dimension reduction (b,f,j). We note that the results presented here are without any post processing operations. In our experience, results for each application can be made visually more appealing (and quantitatively more accurate) by utilizing simple post-processing operations such as morphological closing and opening. Table 2.5 contains quantitative results on each application as described above (Note that we applied confidence voting directly with no weights for scales here). In the brain MR image dataset, on average, our method (both non-PCA and PCA) performs as well or better than the method described by Awate et al [76]. In the teratoma histology dataset, the overall classification accuracy of four tissues (both non-PCA and PCA) outperforms the color K-means method. In the nuclear dataset, although Hausdorff distance metric and added error metric of our results are much larger than the compared result due to the noise effect, which can be easily improved using some simple morphological operations, the overall performances are comparable to the results in [135]. We note that the PCA results are, on average, only slightly less accurate than the non-PCA results.

2.4 Conclusion and discussion

In this chapter, we described a supervised learning-based system for segmenting different types of biomedical images. Our focus was to describe a general purpose system that does not require extensive customization to each segmentation application, so long as enough labeled data is available for training. Different from current learning based methods, which often aim to design specific parametric features for different applications [93, 103], we use intensity neighborhoods as nonparametric feature vectors for pixel classification. Rotations, coordinate inversions and scales are modeled digitally using standard image processing methods. In addition, a subset sampling strategy based on the K-means algorithm is also described. The system is able to handle segmentation of 2D, 3D, as well as scalar and non-scalar (e.g. color) images.

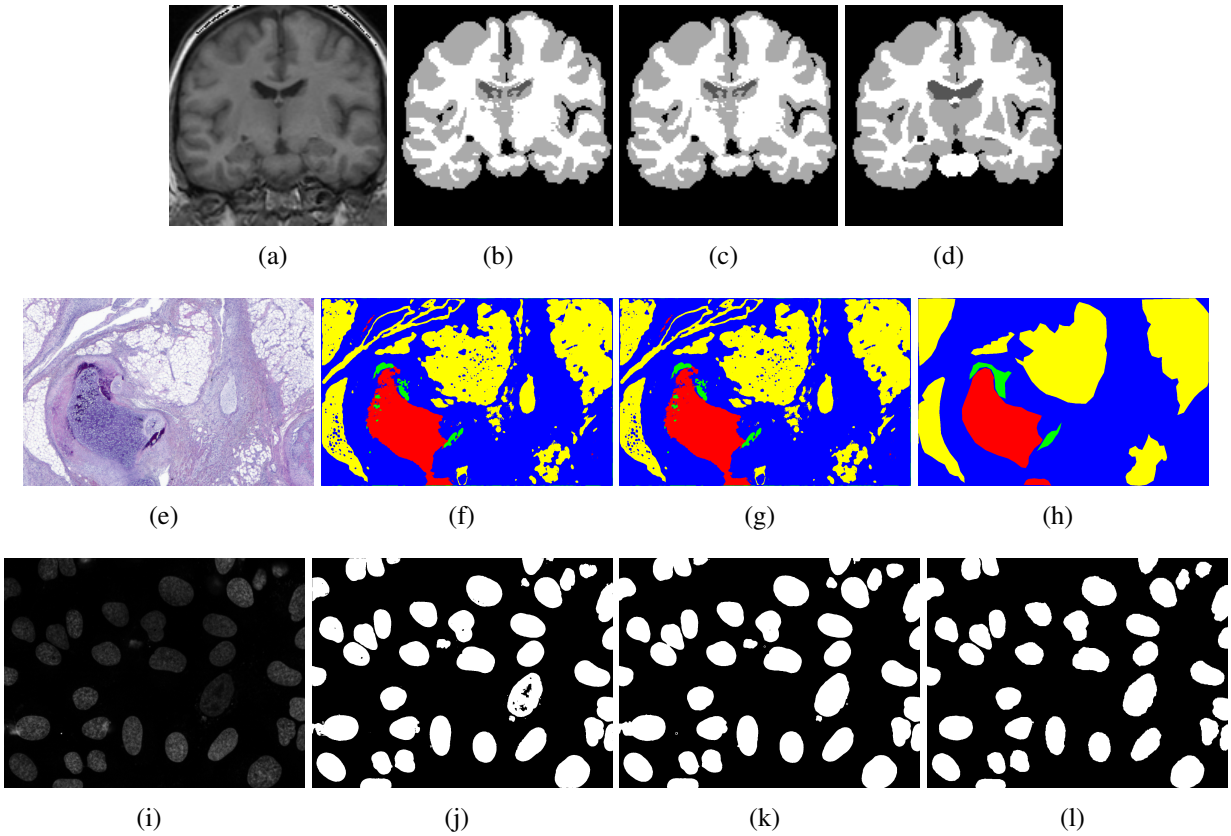


Figure 2.9: Performance of our system in three applications: (a), (e) and (i) are original brain-MR, histology and nuclei images, respectively. Panels (b), (f) and (j) are results produced by our system with dimension reduction. Panels (c), (g) and (k) are results using non-PCA vectors. Panels (d), (h) and (l) are ground truth images.

We compared the application of our method to several other segmentation approaches in three distinct biomedical image segmentation tasks: segmentation of tissues from 3D brain M-R images, segmentation of tissues in color histology images, and segmentation of nuclei from gray-scale fluorescence microscopy images. We have chosen at least one other relatively modern segmentation method for comparison in each application. Overall, our general purpose segmentation system performed as well as (or at times better than) some of the best available custom tailored methods in each application. We also note that the system we described could be further improved by using other post processing operations such as morphological processing. No post processing operation was used for any of the results presented in this chapter.

We note that our system contains several limitations, which offer numerous tasks for future work. An obvious current drawback of our system is that it trades computation time for generality. Let N be the window neighborhood size, and Q_{train} and Q_{test} be the numbers of training pixels and testing pixels respectively, for a 2D/3D image segmentation problem, the training computational complexity is $O(Q_{train}^3 N^2)$ / $O(Q_{train}^3 N^3)$, and the testing computational complexity is $O(Q_{test} N^2)$ / $O(Q_{test} N^3)$. In our implementation (all computation times reported

Table 2.5: Quantitative Evaluation for Three Applications

Case	Non-PCA	PCA	Comparison
brain (GM)	0.9028	0.8953	0.8074
brain (WM)	0.8185	0.8024	0.8868
histology (O)	89.39%	86.19%	55.20%
histology (B)	53.01%	29.03%	29.79%
histology (C)	69.11%	70.45%	51.06%
histology (F)	85.76%	85.08%	58.73%
nuclei (RI)	96%	96%	96%
nuclei (JI)	2.5	2.5	2.2
nuclei (Hausdorff)	130.2	132.6	12.9
nuclei (NSD$\times$10)	0.86	1.12	0.7
nuclei (Split)	0.8	0.2	1.1
nuclei (Merged)	2.8	3.0	1.2
nuclei (Added)	6.1	5.8	0.3
nuclei (Missing)	0.1	0.1	2.9

were based on a single 2.0GHz Intel Xeon processor), the computing time for training the SVM classifier for the brain-MR segmentation task (the training set contained 1.6×10^5 samples) was 12.4 hours for one single scale. Segmenting one 3D MR image containing 9.3×10^4 pixels (128 slices) took 3.8 hours. In our histology image segmentation tests, the corresponding computation times were 14.5 hours (1.2×10^5 training samples) for training, and 6.4 hours for segmenting an image of size 1103×1421 pixels. Finally, in the nuclear segmentation application, the corresponding computation times were 1.55 hours for training (using 8.0×10^4 samples), while segmenting one single image (size 1030×1349) took 0.82 hours.

Another current limitation of our system is related to the selection of the necessary parameters. While some parameters were selected using well-known cross validation strategies, others (such as the neighborhood window size N, the training set size Q) were selected based on empirical procedures (related to prior experience) as well as computational complexity considerations. We note again that, with the exception of the number of training pixels, all other parameters were kept constant throughout all experiments in this chapter. Given the reasonable accuracies obtained in all experiments, we do not expect the accuracies to change significantly for small changes in these parameters. However, we have not exploited to issue of parameter selection appropriately, and plan to do so in future work.

Yet another limitation is related to the number of training pixels available for each class (tissue type). We have noticed that when one class contains much fewer pixels than others in the training procedure, the classifier will tend to give low importance to making an error in such class in the testing dataset. An example of this can be seen in the classification of CSF tissue class in brain-MR segmentation (Figure. 2.6), in which the accuracy for classification of CSF tissue is worse than other tissues. Strategies for minimizing such artifacts are also the subject of future studies. We also mention that our method depends on accurately labeled/segmented data. Generally, the labeled data is randomly selected and the amount of training data is selected

empirically considering the tissue differences in different modalities of data.

In addition, the results showed that when reducing the dimensionality from 27 (non-PCA vector) to 14 (PCA vector), for example, the average practical computational time for training a SVM model for brain segmentation reduces from 4.48×10^4 seconds to 1.53×10^4 seconds (approximately 66% savings) and the time for segmenting one brain data reduces from 1.14×10^4 seconds to 9.33×10^3 seconds (approximately 18% savings), which proves the computational efficiency of using PCA vectors. Finally, we note that the comparable methods were often designed for each specific application, while our system is general: the same system, with identical training procedure, was used for all applications.

We predict that the performance of our system could be further improved by fine tuning several of the steps involved in training and classifications. In Chapter 3, we will investigate the approaches for selecting representative training data from the original large scale training data, and thus, design algorithms for fast pixel classification. In Chapter 4, we will investigate the approaches for region optimization in order to provide better segmentation results.

Chapter 3

Data Selection: A Fast Supervised Segmentation Method

In recent years, with the fast development of computational technologies (e.g. CPU, memory), the idea of pixel level classification using supervised learning strategy has been increasingly utilized for image segmentation. Generally, after the features have been selected, a set of observed samples (e.g. one feature vector per pixel) can be constructed, and a predictive model (e.g. a classifier) can be trained from the observed samples. Then, the segmentation is implemented via predicting the label of every single pixel in the unknown images. The pros of such an idea is that it can be applied to the general data and usually performs robustly [57], while the cons is that it will bring a huge computational cost related to the amount of pixels. For example, utilizing all pixels in an image (e.g. size: 1000×1000) as the training data to train a SVM classifier will definitely result in a huge computational cost. (note that the computational complexity of training a support vector machine (SVM) classifier is $O(N^3)$ with N the size of training data)

The idea of pixel level classification has already been applied to many image segmentation problems. Early in 1978, Panda and Rosenfeld [138] have used gray level and edge value as a two-dimensional feature and a simple classifier to predict pixel by pixel for segmentation purpose. In [139], a local binary pattern (LBP) operator is used to represent textures, and a log-likelihood measure is used to assign each pixel in the unknown images to the most similar class. In [140], intensity neighborhood is directly used as a general feature (e.g. reorder the intensities inside a square window centered at a given pixel into a vector), and a discriminant model based on sparse decomposition is built, which assigns each pixel in the unknown images to the class with the minimum reconstruction error. In [57], by representing intensity neighborhood in a multi-scale framework and training a svm classifier at each scale, the prediction for each pixel in unknown images is obtained via voting through all scales. In [141, 142], color features and gabor filter-based texture features are utilized together, and unknown images are segmented via training a SVM classifier and making predictions pixel by pixel.

Although successful segmentation performance has been achieved on different types of images, such as texture images [139, 140], fluorescence images [57], and histopathology images [57], etc., as far as we know, none of these methods have been able to provide an effective way to manage the training data. In [138, 139, 140], all pixels are utilized directly for training. This may work well when the size of training image is small, however, it will become impractical

when the size of training image becomes large. In [57, 141, 142], several different clustering based algorithms are utilized, which reduce the size of training data to an arbitrarily selected amount. However, it is necessary to point out that these data selection algorithms are ad-hoc with no evidence showing how well the arbitrarily selected training subset keeps the discriminant information. Therefore, in this chapter, we will investigate and develop an effective data selection method, which is able to reduce the size of the given large training set, while keeps the discriminant information of the original set.

3.1 Review of data selection methods

In supervised learning, a training set containing previously known information is provided to classify new coming data. In this chapter, we call each element in the raw training set (denoted as T) as instance. Usually, the raw training set (e.g. the histology images annotated by the expert) contains useless information for the classification task, such that superfluous instances can be noisy or redundant. Thus, training a classifier using the raw training set could be great time-consuming, while the classification accuracy will not be improved. Therefore, it is necessary to discard the superfluous instance from the raw training set T, and thus get a subset S that keeps useful instances. Such a process is named as data selection or instance selection. To better understand such a process, we cite the Figure. 1 in [143] and show it here as Figure 3.1.

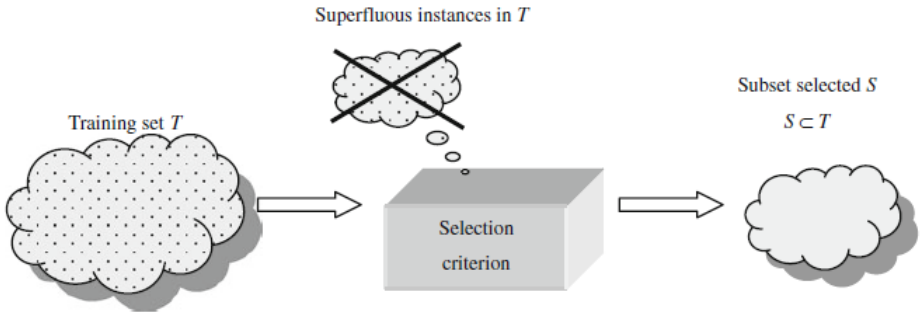


Figure 3.1: The process of data selection

Data selection is a classical research topic in data mining [143]. Here, we focus on selection of training data for supervised learning. Consequently, the runtime in the training process can be reduced via this instance selection process, especially for instance-based classifiers such as KNN method. Currently, many instance selection methods have been proposed, which can be categorized into two types based on the differences of selection criteria: (1) Wrapper methods, which depend on the classification accuracy of a specific classifier (usually 1-nearest neighbor (NN) or k-NN classifier) to select instances. (2) Filter methods, which use a selection function that does not depend on a classifier. A assumption that is commonly used in filter methods is that instances near the discrimination regions between different classes (e.g. decision boundary) are able to provide useful discriminant information for classification [144, 145]. For the reason of brevity, we will only mention a few existing methods in each category. A more detailed review of these data selection methods can be found in [143, 146].

Wrapper Methods: Most of the wrapper methods proposed are based on KNN classifier. Some classical wrapper methods include:

- Condensed Nearest Neighbor (CNN) [147]: It is usually regarded as the simplest wrapper method, which follows an incremental way to select instances. The initial subset S consists in randomly one instance belonging to each class. Then, the algorithm increases the subset by randomly choosing one instance each time and using 1-NN classifier to test whether it can be correctly classified based on the current subset. For a misclassified instance, it will be added to the current subset. CNN is easy to implement, however, it has the cons that its performance is not guaranteed and the selected subset may be different in each selection process, since the instances are randomly tested at each iteration.
- Generalized Condensed Nearest Neighbor (GCNN) [148]: GCNN method is an variant of previous CNN method. Unlike CNN, GCNN selects instances into the current subset S based on an selection criterion related to a threshold. For each instance p to be determined, its nearest neighbors x and nearest enemies w (nearest instances belonging to a different class) will be calculated and their distances (e.g. L2 distance) to the current instance will be compared. If $|p - w| - |p - x| > \delta$ where δ is a threshold value, the current instance p will be ignored, otherwise, it will be added into the current subset S. It is shown in [148] that the subset selected by GCNN performs better in classification accuracy than CNN, however, it will cause a higher data reduction rate.
- Edited Nearest Neighbor (ENN) [149]: Wilson proposed the ENN method based on the idea of removing noise from the raw training set, where an instance is regarded as noise if its class label is different from the majority class of its k nearest neighbors, which will be discarded. Such a method is implemented in a decremental way. However, it is shown in [143] that ENN does not help to improve the classification accuracy too much. It is also mentioned that ENN can be combined with other data selection methods such as CNN in order to remove noise and select useful instances simultaneously.
- Decremental Reduction Optimization Procedure (DROP) methods [144]: In these methods, $Associates$ of an instance p are defined as those instances such that p is one of their k nearest neighbors. DROP1 discards an instance p from T if the $associates$ of p in S are correctly classified without p. DROP2 searches the $associates$ of an instance in the whole training set. DROP 3, 4, and 5 are some variants of previous methods.
- Support Vector Machine (SVM): SVM method can be regarded as another wrapper method since it find the support vectors as the subset for later classification. In some applications, SVM method is combined with some other data selection methods. For example, DROP2 algorithm is applied on the support vectors for double selection. Another method is to apply k-means algorithm to for clustering support vectors. The cons is that SVM method usually associates with huge computational complexity and thus, is not suitable for selecting instances from large scale dataset.
- Fast Condensed Neaerest Neighbor (FCNN) method [150]: It can be regarded as another variant of CNN method. Considering that a random data selection process will cause the randomness of classification performance, Angiulli [150] proposed a fast condensed nearest neighbor (FCNN) method, which is able to select instances from large data sets

in a deterministic way. In [150], four different data selection criteria are discussed and comparable performance is provided. We also mention that our proposed algorithm is an extension of FCNN method with the application on image segmentation.

Filter Methods: Unlike wrapper methods, filter methods do not depend on a classifier to determine the instances to be discarded from the training set. In this category, a border instance is defined if it belongs to one class, while it is the nearest neighbor of an instance from other classes; otherwise, it is called as interior instance. It is assumed that the border instances provide the information of decision boundary in the feature space, thus, the goal is to select only border instances into the subset.

- Pattern by Ordered Projections (POP) [151]: POP methods define a $weakness(p)$ concept as the numbers of times that an instance p is not a border instance in a class, and discards irrelevant instances such that $weakness(p) = m$, where m is the total number of features describing p. One variant of POP is POP-Nearest neighbor (NN) [152], where data is selected by computing the mean of the instances in each class. To find a border instance in one, POC-NN method computes the mean of the instances in the opposite class and then the instance belonging to previous class which is the nearest to the mean of opposite class is regarded as the border instance and selected. The border instances in the opposite class are found in a similar way; finally, only the border instances are selected.
- Clustering methods [153]: Clustering is often seen in data selection methods. A typical idea is to split the the raw training set T into a couple of clusters and choose the centers of the clusters into the training subset [154]; In GCM (Generalized-Modified Chang Algorithm) method [155], the algorithm aims to merge the same-class nearest clusters and selects the centers from the new merged clusters. The NSB (Nearest Sub-class Classifier) method [156] allows selecting different number of instances (centers) per class via theMaximum Variance Cluster algorithm.

Although a large number of data selection algorithms have been developed, these existing ones cannot be applied directly to image data due to their high data reduction rates (defined as the ratio between the subset size and the whole set size), which still, produces a large sized subset and thus, makes computational cost for both training and testing processes large. In order to achieve a reasonable computational cost, the ideal case should be to extract a small sized subset, and train a simple classifier based on the given subset. However, one common sense is that it is contradictory to achieve a high classification accuracy by using a weak classifier (defined to be a classifier which is only slightly correlated with the true classification, meaning that it can label examples better than random guessing) trained from a small sized dataset. Fortunately, this dilemma can be solved by the majority voting strategy. Given a group of weak classifiers, it has been proved that the classification accuracy by majority voting within the group increases monotonically with the size of the group as long as all weak classifiers' accuracies are higher than 50%. To be specific in image data, by utilizing the characteristics of spatial connectivity between neighboring pixels, it is reasonable for us to make an assumption that generally, all pixels within a local window in the images belong to the same class. Thus, we can hypothesize that it is very possible to increase and thus, achieve a high classification accuracy via majority voting spatially within local windows, even using a weak classifier derived from a small sized training set.

Consequently, our goal in this chapter is to develop a data selection method specifically designed for image data using majority voting strategy. The basic idea of our approach is similar to FCNN [150], which iteratively absorbs the new instances into the subset based on our proposed data selection criterion in a deterministic and incremental process. With a training subset selected, segmentation on unknown images can be implemented by classifying every single pixel using 1-NN classifier and the training subset, followed by majority voting spatially within local windows. Our approach is expected to have the following properties: (1) low data reduction rate; (2) comparable and robust performance. Our first contribution is to propose a new voting based data selection criterion. By showing experimentally, we claim that the subset selected by our voting based criterion is able to provide better classification performance than several existing data selection criteria. In addition, our second contribution is to provide a simple, general and fast approach for segmenting images robustly. We show experimentally that our approach is able to provide comparable or even better segmentation results over several different image data compared to the state-of-the-art segmentation methods with much lower computational cost, and thus, our second claim is that with an effective data reduction algorithm, even very simple classifiers are able to achieve good results.

3.2 Methodology

3.2.1 The voting based instance selection algorithm

Given a set of training images with features selected, an initial training set T_N can be built by computing a feature vector $\mathbf{x}_i$ for every pixel i in the training images named as an instance and taking all instances into this set T_N, where $T_N = \{(\mathbf{x}_1, l_1), (\mathbf{x}_2, l_2), ..., (\mathbf{x}_N, l_N)\}$, N is the amount of all pixels from the training images, $\mathbf{x}_i$ is the feature vector for pixel i, and l_i is the class label of $\mathbf{x}_i$ ($l_i \in \{1, ..., C\}$, C is the number of classes).

The goal is to extract a subset U_n with the size n, which is expected to satisfy two conditions: (1) the size of this subset is restricted (e.g. $n \leq n_{max}$ where n_{max} is an arbitrarily defined maximum size of the subset) in order to reduce the computational cost of testing process. (2) the segmentation accuracy of the training images is maximized under the constraint of subset size. In addition, it is important to discuss the criteria for data selection, as it is related to the classification performance directly. Since it is usually very difficult to validate the selection criteria theoretically, most current literatures design selection criteria based on different intuitions. For example, one criterion used in FCNN [150] is that for every instance existing in the current subset, its nearest misclassified instance will be added to the subset. In this work, we have an intuition that every class is equally important in classification, and thus, we designed our data selection criterion that at each iteration, a new misclassified instance that belongs to the class with the lowest classification/segmentation accuracy will be selected and added to the current subset. Especially, we note that since majority voting spatially within local windows will be utilized after pixel level classification in the testing stage for improving segmentation accuracy, we evaluate the classification/segmentation accuracy for the training images at each iteration in the same way.

Before explaining our algorithm in details, we first cite several definitions from [150]: given

the initial set T_N and a subset U_n, let z be an element of the subset U_n, a Voronoi cell of z is defined as $Vor(z, U_n, T_N) = \{q \in T_N | z = \arg\min_{r \in U_n} d(r, q)\}$ where d is the distance metric (e.g. $L2$ metric). In addition, the centroid of a set (e.g. named R) is defined as $Centroid(R) = \{q | q = \arg\min_{r \in R} d(r, mean(R))\}$. Then, we implement our data selection algorithm in an incremental and deterministic way with the following steps:

1. Initialization: For each class c ($c = 1, ..., C$), calculate the centroid of instances $\mathbf{x_i}$ whose class label $l_i = c$ in the initial training set, and add it into the training subset U_n as the initial subset.
2. Classification: Use 1-NN classifier and current subset U_n to classify every single pixel in the training images.
3. Voting: Vote spatially within a square window (size: $w \times w$) centered at every pixel in the training images, and re-assign the class label that has the maximum number of votes in the square window to that pixel as the final classification. Calculate the classification accuracy for each class.
4. Absorption check: choose the class with the lowest classification accuracy named as "the worst class", then, find the instance in the current subset U_n whose Voronoi cell contains the largest number of misclassified instances belonging to "the worst class", and calculate the centroid of these misclassified instances within that Voronoi cell, and add the centroid instance to the subset U_n.
5. Go back to step 2: Classification. Keep iterating to grow the subset U_n until the size of subset n has grown to n_{max}.

After the process of iterations has been finished, we extract the smallest sized subset whose overall classification accuracy is higher than a threshold value as the finally selected training subset U_n, which can be represented mathematically as $U_n = \arg\min_n \{U_n | accuracy(U_n) \geq Th\}$, where the accuracy threshold value Th is set as: $Th = \max accuracy(U_n) - 0.01$ ($accuracy(U_n) \in [0, 1]$).

3.2.2 Image segmentation via classification

After the training subset U_n has been appropriately selected, we implement the image segmentation procedure via a two-step process: The first step is to classify every single pixel in the unknown images using the simplest lazy learning model: 1-NN classifier based on the training subset U_n; followed by the second step, which is to do majority voting spatially within square windows ($w \times w$) centered at every pixel in the unknown images, and assign the majority class label to that pixel as the final segmentation result.

3.3 Experimental results

In this section, we will validate the two claims via experiments. Two dataset are chosen: (1) Brodatz texture dataset [157], in which Randen and Husoy [158] created a set of gray-scale

images composed of textures from different classes with corresponding training images (size: 256×256) from each class provided. A total of 12 testing images consisting 2 classes, 5 classes, 10 classes and 16 classes can be seen in [139], and the goal is to segment different texture regions. (12 testing images are shown in Figure 3.3) (2) Histology images (hematoxylin and eosin (H&E) stained) of teratoma derived from human and nonhuman primate embryonic stem cells [84], which were acquired by using a slide scanner to scan the whole sectioned teratoma slides at a high resolution after H&E staining. The color images obtained contain $R * G * B$ channels, and the resolutions are 3.527 microns/pixel (0.283 pixels/micron) and approximately 3-5 micron thickness (z-dimension). Since the number of pixels in raw images is usually large (e.g. size 4824×4014), image patches of smaller sizes are randomly cropped (e.g. size from 896×932 to 1438×1106) for simplification. In this experiment, we utilize the same training and testing images used in [57], where 4 images were selected randomly for training, and the remaining 10 images were used for testing. The goal is to distinguish and segment the regions of 4 tissue classes: bone (B), cartilage (C), fat (F) and background/other tissues (O). In addition, the ground truth segmentations (used in both training and validation) were provided by the pathologist (J.A.O).

In order to provide a fair validation, we set up the experiments by using a very simple and general intensity neighborhood feature instead of considering those specifically designed features. For each pixel in the image, we set the neighborhood size $m = 3$ as a 3×3 square window for 2D gray-scale images and a $3 \times 3 \times 3$ cube window for RGB channel images centered at the given pixel, and intensities inside the neighborhood are represented rotationally invariant [159] by sorting the pixels' intensities square symmetrically and arranging them into a vector (shown in Figure 3.2). In this way, we prevent augumenting the data by adding rotational and flipping versions of the same intensity neighborhoods. Especially, we note that using intensities directly as the feature can be very sensitive to intensity normalization, which will result in serious performance loss even for the same training/testing data when intensities of data are not normalized properly. In this work, with no prior information about the image, we normalize the data by simply dividing the pixels' intensities by 255 to restrict the intensity range within [0, 1]. This works well for histology images, since all training/testing images' intensities are within [0, 255]. However, for Brodatz data [158], each class of texture regions in the testing image and its corresponding training image are extracted from different area in the same source texture image, and the source texture images were only globally histogram equalized prior to being used [139]. As a result, the training and testing patches extracted from different area in the large source image may have different intensity ranges. Furthermore, there are a few notable intensity differences (e.g. uneven illumination) in some texture classes in the training/testing images. These intensity heterogeneous problems will cause serious misclassification results when an intensity feature with a small neighborhood size is used and thereby, leads to a wrong conclusion about the proposed work. Thus, a reasonable remedy is to check and compare the intensity ranges between training and testing patches for each texture class. When the intensity range of training texture patch does not cover the intensity range of testing patch from the same texture class (for example, for one texture class, the intensity range of training patch is [11,244], while the intensity range of testing patch of the same texture class is [0,255]), we linearly transform the intensity range of the training patch for that texture class to [0,255]. Finally, all training images and testing images are divided by 255 as we implemented before to normalize the data.

Besides, we note that the proposed algorithm only contains two parameters, whose values are

manually selected for the following experiments: the constraint of subset size $n_{max} = 200 \times C$ (C is the number of classes) and voting window size w for Brodatz data and histology data is set to be 21 and 51 respectively.

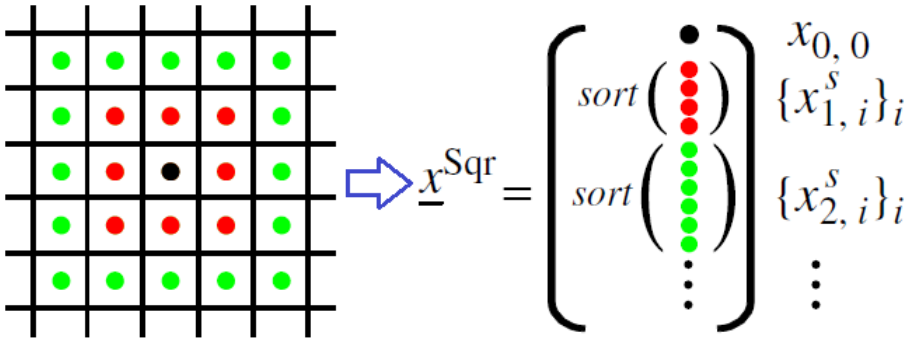


Figure 3.2: Rotational invariant representation of intensity neighborhoods

3.3.1 Comparison between different data selection criteria

Since it is usually very difficult to analyze the performance of data selection criteria theoretically, an alternative way that most literatures adopt is to compare their performance experimentally under the same condition. Here, we set up an experiment by randomly picking out one texture mosaic image (Figure 3.5(d)) consisting of 5 texture classes from Brodatz dataset for testing, together with 5 training images (one for each texture class), and comparing four different criteria for data selection: (1) our proposed criterion, which depends on the lowest voting based classification accuracies among classes at each iteration to decide which class of instance to add (the "worst" class) and then, the Voronoi cell that contains the largest number of misclassified instances belonging to the "worst" class will be chosen, from which the centroid of misclassified instances from the "worst" class will be added to the current subset. (2) select without vote criterion, which is almost the same as the proposed criterion except that pixel based classification accuracy is directly used to decide the "worst" class without majority voting spatially. (3) FCNN3 [150], which simply chooses the Voronoi cell that contains the largest number of misclassified instances, in which the closest misclassified instance (measured in $L2$ metric) will be added at each iteration; (4) FCNN4 [150], which is almost the same as FCNN3 except that the closest centroid of misclassified instances among classes will be added at each iteration instead.

Here, we evaluate the performance of different criteria by comparing their voting based classification accuracies on the same testing image using 1-NN classifier based on the subsets selected by different criteria. In Figure 3.4, we first start the subset size at 5 instances (each instance corresponds to the centroid of each texture class) and then, increase the size of subset by every 50 instances from 50 instances to 1200 instances (axis x) and show the corresponding voting based classification accuracies obtained by different data selection criteria (axis y). Obviously, we can see from Figure 3.4 that the overall performance of the proposed criterion (criterion 1) outperforms the rest three criteria, which shows experimentally that our criterion can be regarded

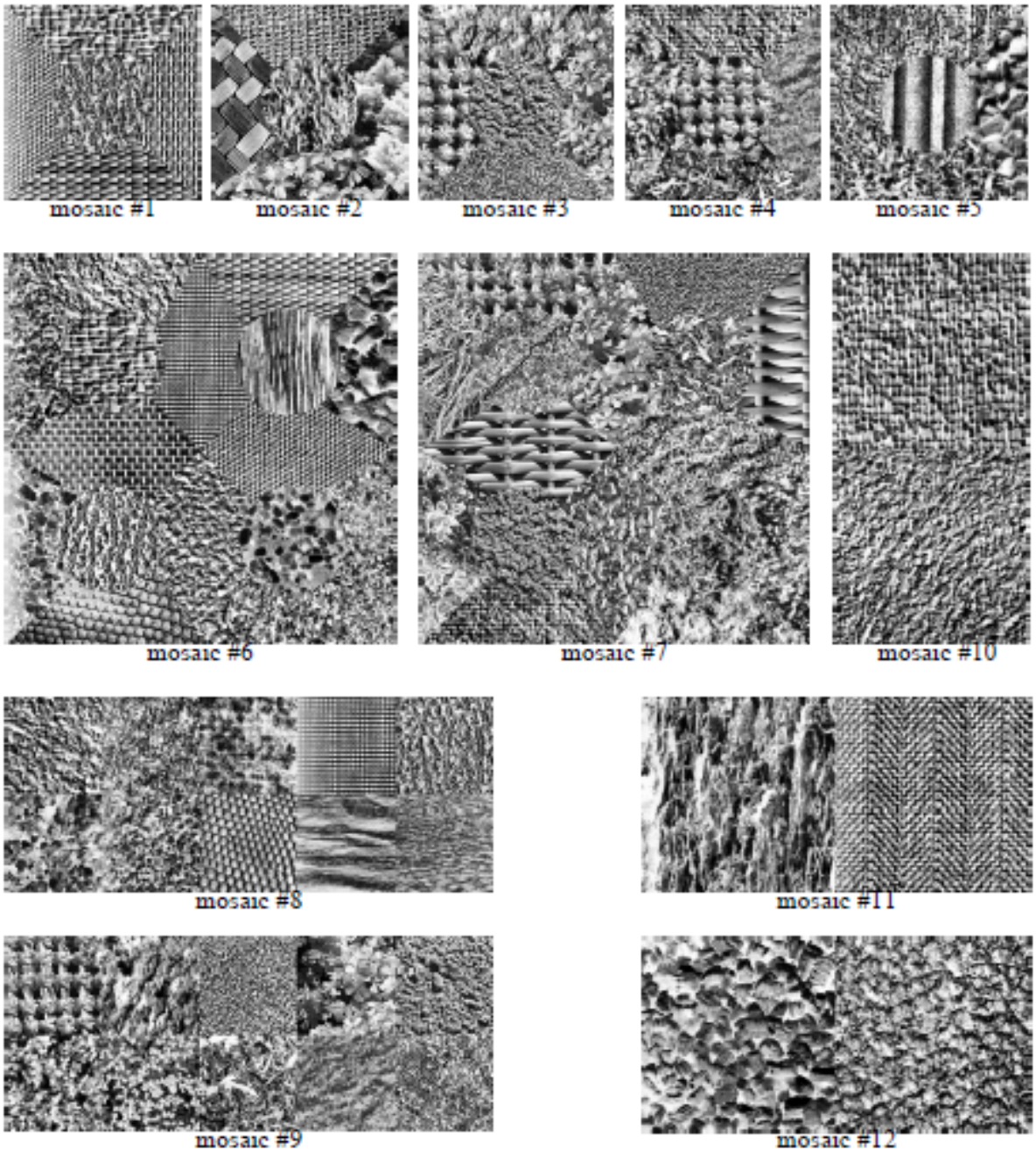


Figure 3.3: 12 texture images for testing

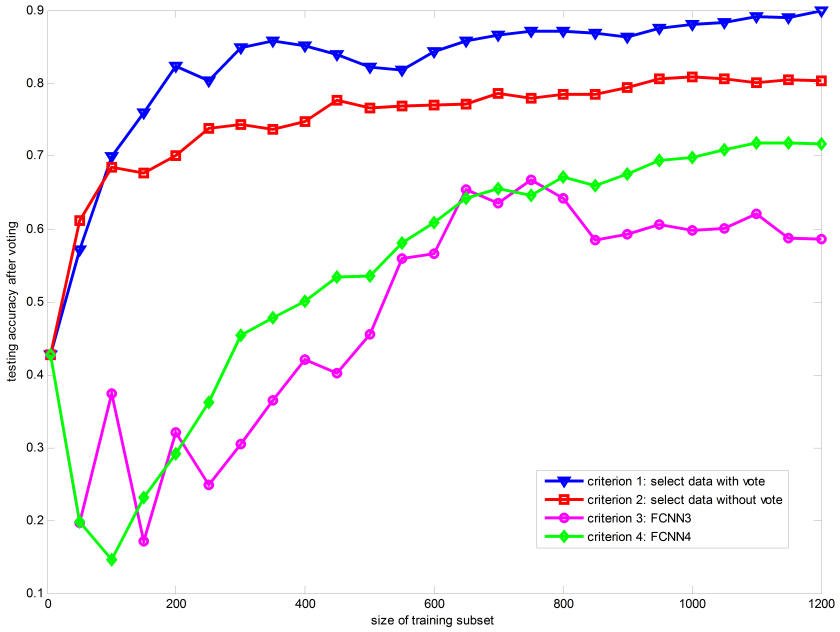


Figure 3.4: Comparison of segmentation accuracies between different data selection criteria

as a better one for selecting subset from image data. In addition, we note that such an experiment has also been applied to several other different images, which achieves the same conclusion.

3.3.2 Compare with current texture segmentation methods

Texture image segmentation is a classical problem in image processing. Here, we evaluate the proposed approach by applying it to Brodatz texture dataset [157], which is a public dataset commonly used in many literatures for evaluating texture segmentation performance, and comparing the results quantitatively with five state-of-the-art methods: filter-bank methods [158], local binary pattern descriptor [139], sparse representation [160], classification method using frequency based features [161] and discriminative learned dictionaries method [140]. Especially, we mention that discriminative learned dictionaries method [140] applied a smoothing step using graph cut alpha expansion algorithm after the classification step, which greatly improved the accuracy. Thus, in order to provide a fair evaluation, we not only simply evaluate the voting based classification results, but also evaluate the results after smoothing using the same graph cut technique [140]. Here, the graph cut algorithm is implemented based on a simple Potts model within 8-connectivity neighborhood, where the cost for associating a pixel i to the class c is set as $1-p(l_i = c)$, where $p(l_i = c) = \frac{\#(l_j=c|j\in w(i))}{\#(k\in w(i))}$ (the ratio of pixels whose labels are predicted to be c within the square window), and $w(i)$ is the square window $w \times w$ centered at the given pixel i. And a constant regularization cost between two neighboring pixels is set to be 0.8. In Figure 3.5, we show one example of texture mosaic image (left column) consisting of different number of texture classes at each row, together with the ground truth (middle column) and the segmentation

Table 3.1: Comparison of Error Rates for Brodatz Texture Data Between Different Methods

#	[158]	[139]	[160]	[161]	[140]	D.S. 1	D.S. 2
1	7.2	6.7	5.5	3.37	**1.61**	22.65	14.94
2	18.9	14.3	7.3	16.05	16.42	11.11	**5.62**
3	20.6	10.2	13.2	13.03	4.15	10.86	**3.93**
4	16.8	9.1	5.6	6.62	**3.67**	7.09	4.64
5	17.2	8.0	10.5	8.15	**4.58**	19.76	6.56
6	34.7	15.3	17.1	18.66	**9.04**	53.35	41.06
7	41.7	20.7	17.2	21.67	**8.80**	38.79	8.94
8	32.3	18.1	18.9	21.96	**2.24**	48.85	24.97
9	27.8	21.4	21.4	9.61	**2.04**	25.40	2.61
10	0.7	0.4	NA	0.36	**0.17**	1.32	1.20
11	0.2	0.8	NA	1.33	**0.60**	4.15	0.84
12	2.5	5.3	NA	1.14	**0.78**	1.56	1.36

results with graph cut smoothing (right column). From Figure 3.5, we can see that the overall segmentation on texture images is quite precise expect in Figure 3.5(l), where some regions are misclassified into wrong texture classes. In addition, we report the error rates between different approaches in Table 3.1, where the column "D.S. 1" (D.S. is short for data selection) corresponds to the error rates of standard voting based classification, while the column "D.S. 2" corresponds to the error rates with graph cut smoothing (note that the best performance is shown in bold font). From Table 3.1, we can see that although discriminative learned dictionaries method [140] (with graph cut smoothing) obtained the most best results, our proposed approach (with graph cut smoothing) still achieves overall comparable performance except the first, sixth, and eighth images. The loss of accuracies on these three images can be explained as we choose small intensity neighborhoods (3×3 window) as the feature, which is very sensitive to the notable intensity heterogeneous regions in the testing images, and causes misclassification.

3.3.3 Compare with current histology tissue segmentation methods

Segmenting tissues from histology images has been a challenging problem due to its complicated tissue appearance (e.g. texture, color, structure, etc), which is very difficult to model. For the histology images of teratoma (H&E stained) [84], the specific challenges [57] include low intra-class similarity (the same type of tissue often has different visual appearance) and high inter-class similarity (multiple types of tissues may have similar visual appearance). Although a few approaches have been developed, for example, [57] proposed a multi-scale pixel level classification approach using intensity neighborhood information and SVM classifier, which achieved good segmentation performance on histology images of teratoma, it is worth noting that this method's computational cost is quite huge, making it uneasy to use. As far as we know, currently, there is no practical approach which is able to segment such images reliably and efficiently. Here, we evaluate our proposed approach using the same histology images of teratoma (both training and testing) used in [57], and compare to the results reported in [57]. In order to compare fairly, we

Table 3.2: Quantitative Evaluation for Images of Teratoma Histology

Accuracy	B	C	F	O
Intensity neighborhood SVM	59.70%	73.18%	91.09%	**88.93%**
Color K-means	29.79%	51.06%	58.73%	55.20%
Color SVM	27.87%	29.09%	66.16%	68.12%
Data selection	**81.59%**	**78.69%**	**94.57%**	84.53%

also utilize the same multi-scale framework in [57], in which the neighborhood size $m = 3$ keeps constant, while at each scale s ($s = 0, 1, ..., S$ and $S = 4$), we first convolve images belonging to the training set with a Gaussian kernel for smoothing, and assemble the neighborhood patch centered at each pixel by subsampling at every 2^s pixels (after smoothing). Then, we are able to construct the initial training set and select the training subset at different scales following the standard steps described above. For segmenting an unknown image, again, we follow the same weighted confidence based voting framework in [57], where the final prediction l_i^{final} for each pixel i is obtained via:

$$l_i^{final} = \underset{c \in \{1,2,...,C\}}{\arg\max} \sum_{s=0}^{S} \lambda_s p(l_i^s = c) \tag{3.1}$$

where λ_s are the weights for 1-NN classifiers at different scales. The optimal weights $W_{optimal}$ can be calculated by maximizing the objective overall classification accuracy function M as:

$$W_{optimal} = \underset{W \in R^{S+1}}{\arg\max}\, M(R_{train}(W), G_{train}) \tag{3.2}$$

where $M(\cdot, \cdot)$ is a measurement of overall classification accuracy, W is a group of weights ($W = \{\lambda_0, \lambda_1, ..., \lambda_S\}$)), $R_{train}(W)$ corresponds to the labeled (segmented) training data, while G_{train} corresponds to the ground truth.

In Figure 3.6, we randomly choose one histology image of teratoma from the testing data, and show the segmentation results of several different methods including: our proposed data selection method, intensity neighborhood SVM method [57], color k-means method, color SVM method, together with the ground truth. Four classes of tissues are segmented and represented: the green color represents B tissue, red color represents C tissue, yellow color represents F tissue, and blue color represents O tissue regions. From Figure 3.6, we can see that our method (Figure 3.6(e)) achieves comparable or even better result when compared to the rest several methods and the ground truth. We also show another segmentation result in Figure 3.7. In addition, we report the pixel classification accuracies for each tissue class in Table 3.1, and compare quantitatively with the results from [57]. We can see that although our proposed method (data selection) gets a lower classification accuracy for tissue O than the intensity neighborhood SVM method [57], it achieves the best classification accuracies for the rest three tissue classes.

Table 3.3: Data Reduction Rates

Data Name	n VS. N	Reduction Rate
Brodatz (2 classes)	202 VS. 1.31×10^5	0.15%
Brodatz (5 classes)	1092 VS. 3.27×10^5	0.33%
Brodatz (10 classes)	1819 VS. 6.55×10^5	0.28%
Brodatz (16 classes)	2828 VS. 1.05×10^6	0.27%
Histology (4 classes)	154 VS. 3.99×10^6	0.00385%

3.3.4 Analysis of computational efficiency

Finally, it is important for us to analyze the computational efficiency of the proposed approach, since it is another goal we aim to achieve in this chapter. We mention that the computational cost for data selection as the training process is $O(NS_U)$ [150], where N is the amount of all pixels in the training images, and S_U is the maximum size of the subset. In addition, since 1-NN classifier is utilized for pixel classification, the computational complexity for segmenting an unknown image is $O(nK)$, where n is the size of subset finally selected, and K is the amount of pixels in the testing image. Apparently, the value N and K are determined by the size of training and testing images directly and keep unchanged once fixed. In order to save computational time for training process, we manually choose a rather small value of n_{max} (reported above). In addition, we report the average subset sizes for different data and related data reduction rates in Table 3.3.

In our implementation (all computation time reported were based on a Intel(R) Core(TM) i5 2.30GHz CPU), for Brodatz data consisting of 2 classes, 5 classes, 10 classes, and 16 classes, given the initial training set, the computing time for standard data selection process is 0.61 minutes, 6.32 minutes, 51.30 minutes, and 229.3 minutes respectively, while it takes 1.1 minutes to segment the testing image of 2 classes (size: 256×512), 0.62 minutes for the testing image of 5 classes (size: 256×256), 2.0 minutes for the testing image of 10 classes (size: 256×640), and 4.7 minutes for the testing image of 16 classes (size: 512×512). For the histology images of teratoma, it takes 2.11 hours to select a subset from 3.99×10^6 instances, and 0.41 hours to segment an image of size 1103×1421 pixels. Compared to the reported computational timein [57], where it takes 14.5 hours to train a SVM model from 1.2×10^5 instances, and 6.4 hours to segment the same histology image, although the CPU we use in this work is slightly better than the one used in [57], we can still conclude that the computational efficiency has been greatly improved by the proposed approach.

3.4 Conclusion and discussion

In this chapter, we proposed a voting based data selection method, which selects data iteratively based on the voting based classification accuracy among classes. Instead of trying to seek a subset that achieves 100% accuracy on the initial training set (named as training-set-consistent-subset), we focus on improving the computational efficiency by restricting the maximum size of the subset, and we validate the performance of our proposed method via a couple of experiments, which show that when the training data is appropriately selected, even by using very simple

features (intensity neighborhoods feature in this work), and using very simple classifier (1-NN classifier), we are still able to achieve comparable or even better results across different types of datasets compared to other state-of-the-art methods specifically designed for these datasets, together with a relatively small computational cost. Thus, this approach could be very useful when dealing with very large scale datasets. Finally, we note that although we have shown that our proposed data selection criterion outperforms others experimentally in Figure 3.4, we do not provide any theoretical support explaining why it works better than others. Thus, our future work is to find the theoretical support and also, investigate whether any better data selection criterion exists.

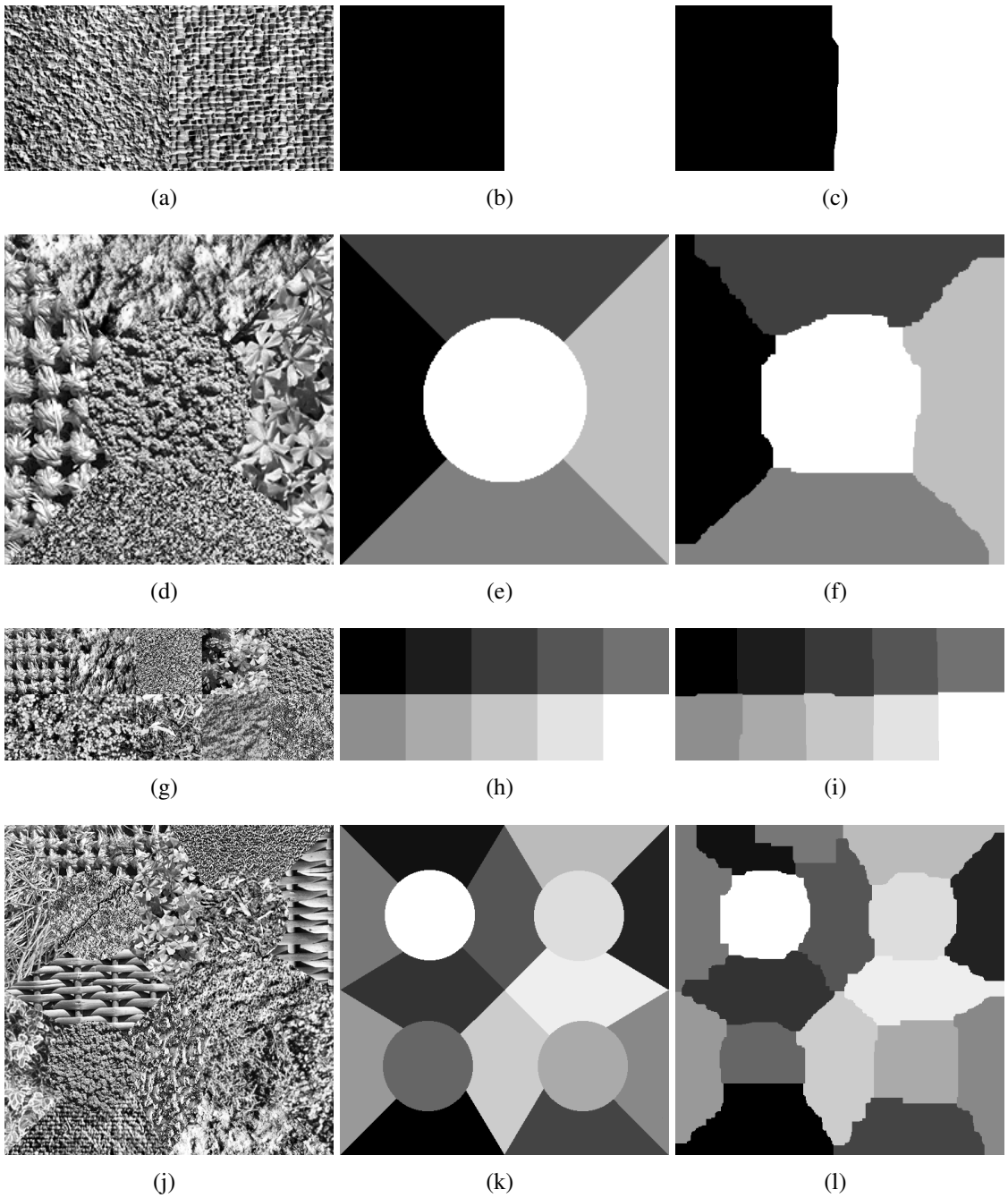
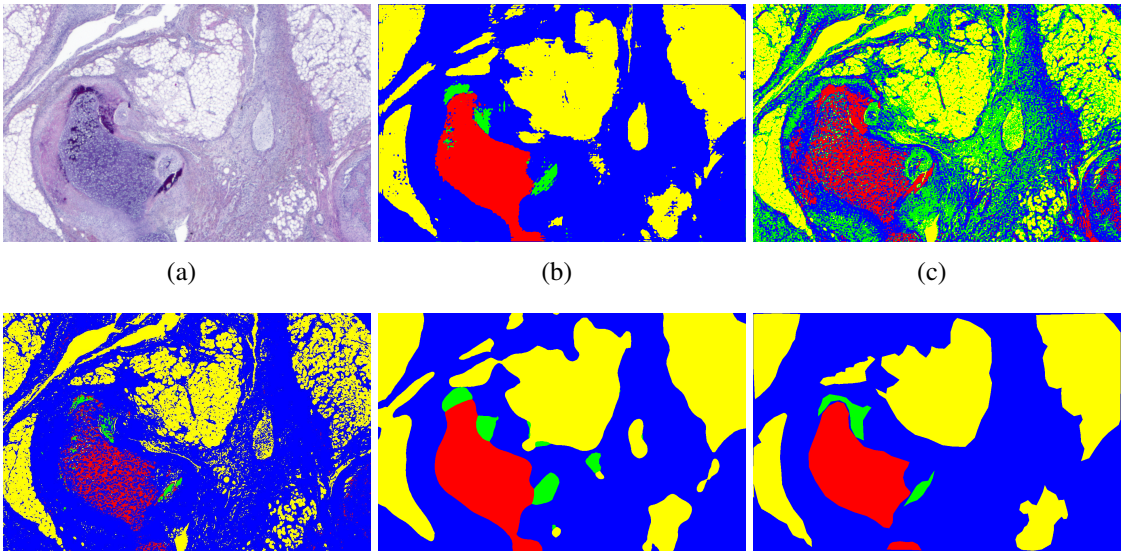


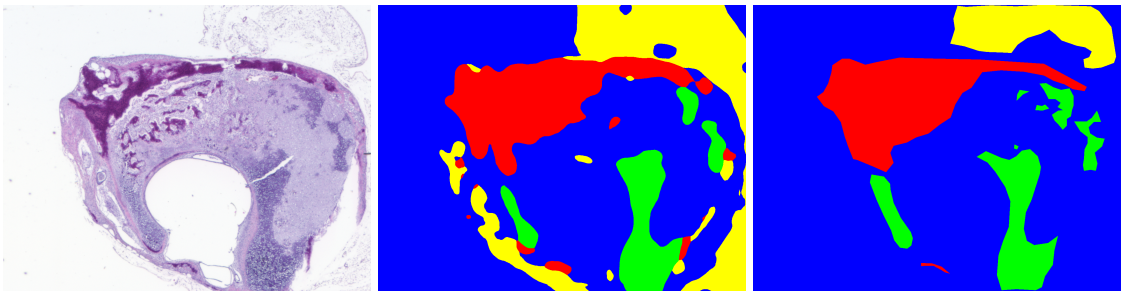
Figure 3.5: Segmentation of texture images in Brodatz data. Left column: original images; middle column: the ground truth; right column: segmentation results with graph cut smoothing.



(a) (b) (c)

(d) (e) (f)

Figure 3.6: Segmentation of tissues for histology images of teratoma (a) original image; (b) intensity neighborhood SVM's result; (c) color K-means' result; (d) color SVM's result; (e) data selection result; (f) ground truth.



(a) (b) (c)

Figure 3.7: Another segmentation of tissues for histology images of teratoma (a) original image; (b) data selection result; (c) ground truth.

Chapter 4

Region Optimization: A Post Processing Step

Image segmentation, according to its definition, is to subdivide an image into several constituent regions (named as segments), and each segment contains a set of pixels, where pixels in the same segment share certain visual characteristics such as color, intensity, or texture. Image segmentation can be regarded as a pixel classification process, once a method for determining whether a pixel belongs to a particular class (object region or boundary) has been proposed, a final segmentation result can be obtained by grouping the pixels with the same label into the same object (or background). However, the segmented regions directly obtained from the pixel level classification procedure are not always satisfactory, as they have jagged borders with holes inside. Since a common requirement for the segmented regions is to have smoothed borders/surfaces, it is necessary to optimize the classification result before the final output.

In this chapter, we will investigate the apporoaches for region optimization. We note that there are many different types of algorithms to group pixels for region optimization. In the simplest case, for example, connected components may be obtained by simple morphological analysis, such as dilation and erosion, however, segmented regions after processing usually contain jagged borders. Spatial voting method, as mentioned in the previous chapter, can be applied to smooth the borders. However, these methods tend not to be robust enough for quantitative biomedical applications. Comparatively, modern methods for region optimization usually incorporate both the region information and border information into the objective function, and optimize the objective function simultaneously.

4.1 Review of optimization methods

Based on the optimization techniques used, we would like to divide current region optimization approaches into two big categories: (1) optimization in continuous field, of which the representative method is deformable model; (2) optimization in discrete field, of which the representative method is graph cuts. In the following, we will discuss and compare these two types of methods.

- **Deformable model**: As discussed aboves in Chapter 1, deformable model simulates curves or surfaces that can move under the influence of internal forces, which are defined within

the curve or surface itself, while the external forces can be calculated from the image data. The use of internal force is to keep the model smooth during deformation, which usually depends on curvature of contours, while the use of external force is to move the model toward an object boundary or other desired features within an image based on the predesigned models. Deformable models have been extensively studied and widely used in biomedical image segmentation with promising results, since it is able to offer robustness to both image noise and boundary gap by constraining extracted boundaries to be smooth and incorporating other prior information such as the object shape. Two types of methods can be utilized to represent the deformable model: parametric (explicit) (e.g. snake based approaches [34, 162]) or nonparametric/geometric (implicit) methods (e.g. level set-based approaches [71, 99, 163]). Parametric deformable models allow direct interaction with the model and can be implemented quite fast. However, it has the disadvantage of topology constraint, which cannot deal with splitting or merging cases during the process of deformation. Thus, non-parametric deformable model represented by level set is more preferred, where curves and surfaces are represented by a level set of a higher-dimensional scalar function.

Deformable models can be used to optimize the segmented region (the process of grouping pixels belonging to the same object) in an image by incorporating the criterion of grouping pixels into the external force model and then, starting from a "guess" configuration. The deformable models can be optimized through gradient descent techniques with a stable point satisfying the appropriate Euler-Lagrange equation. Many criteria for grouping pixels have been proposed in the past years, ranging from model-driven criterion to data-driven criterion. Here, we illustrate a few examples: In [37], pixels where the borders of the regions have the largest gradient value are grouped, however, since no regional information is considered in this model, the moving curves may capture some places inside the object where high gradient exists rather than the real borders of objects. In [38], a region based criterion is proposed, which assumes that intensities should be similar for the pixels belonging to the same class, and thus, the intensity variance for the pixels in the same class should be minimized. This model is easy and robust to implement, however, it may fail when dealing with some complicated images, where the pixels in the same class share some other characteristics such as texture rather than gray level intensity, meaning that intensity variance for the pixels in the same class could be very large in fact. In addition, we note deformable models have been utilized combining with supervised and unsupervised learning strategies, as well as with region and edge based methods [164, 165].

- **Graph cut model**: Graph cut model, as mentioned before, is another effective tool for optimization. By regarding Image segmentation as a pixel labeling problem, the goal is to find an optimal labeling configuration for the pixels in the image which minimizes the predefined energy function. Usually, this energy function consists of two terms: region term and edge term, which is similar to some forms of deformable models. The region term usually incorporates the criterion of grouping pixels into the same object. For example, one criterion is to take into consideration the probability of current pixel belonging to one class. The larger the probability is, the smaller the penalty value will be, and vise versa. The edge term considers the connections between pairwise pixels belonging to

different classes. The larger difference (e.g. difference between two pixels' intensities) between pairwise pixels means smaller penalty value for the edge term. In most cases, Markov Random Fields (MRF) [44] is used to model the connections between pairwise pixels and only pairwise pixels within 4-connectivity and 8-connectivity are considered as connected. Such an energy function can be minimized by representing such an energy objective function into a graph and incorporating data term and edge term into the edge weights. The energy function can be minimized by finding a cut which separates the whole graph into two dis-connected part and also satisfies that the sum of edge weights whose edges have been disconnected by this cut (the min cut) is the minimum. Finding the min cut for a two-class problem (binary segmentation) can be realized via push-relabel algorithm [166], which is a global optimization. However, finding the min cut for multiple classes (multi-way cut problem) is NP-Hard and some approximation algorithms that produces a cut which minimizes the original energy function in a strong sense have been proposed, such as alpha expansion algorithm, alpha-beta swap algorithm, etc [44].

4.2 Methodology

In this work, our goal is to develop a robust method for automatically delineating regions that encompass each single object in the image, and thus, we propose to use a more flexible approach based on deformable model. More precisely, in order to deal with the result from a pixel level classification procedure, the first part of the region optimization work is related to constructing a function $P(x, y)$ that assigns the confidence value of a pixel $p(x, y)$ at the coordinates (x, y) to belong to one class (type of object) versus another. We start from the example of binary segmentation, which is to segment a foreground class/object vs. a background class from an image $I(x, y)$. In our case, $P(x, y)$ is estimated from the labeled/training input images as described in detail in [167]. More precisely, the function $P(x, y)$ provides a decision or confidence value for the pixel $p(x, y)$ as:

$$\begin{cases} P(x,y) < 0 \Rightarrow p(x,y) \in R_A \\ P(x,y) > 0 \Rightarrow p(x,y) \in R_B \end{cases} \tag{4.1}$$

where R_A and R_B refer to background region ("bkg") and foreground/object region ("obj") respectively. The functional form for $P(x, y)$ can vary from linear models such as:

$$P(x,y) = \int_u \int_v W_P(u,v) I(x-u, y-v) du dv + b_P, \tag{4.2}$$

where W_P and b_P are estimated from training procedures such as Fisher LDA, or linear SVM, to more complicated boundaries such as:

$$P(x,y) = \sum_{j=1}^{N_j} c_j V_j \phi_j(x, y, I) \tag{4.3}$$

where $\phi_j(\cdot, \cdot, \cdot)$ is a radial basis function of the image and input coordinates, c_j are coefficients obtained during a training procedure, and V_j a class indicator function. The training of such

classifiers can occur through well know methods [52]. In addition, a similar function $P_B(x, y)$ can also be computed from training data (taking the same functional form as above) to assign the probability of a pixel $p(x, y)$ belonging to a boundary between R_A and R_B. For several applications, given input image $I(x, y)$ we are able to provide the input maps $P(x, y)$ and $P_B(x, y)$ necessary for the subsequent objective function for optimization.

Given an image $I(x, y)$, we seek to find a partition/segmentation of it, denoted $\tilde{R}$, by minimizing the following equation:

$$E(\tilde{R}) = \underbrace{\alpha \int_0^1 g(C(s)) \cdot |C'(s)| ds}_{\text{Boundary Term}} + \underbrace{\left\{ \beta \left[\iint_{R_A} r_A(x, y) dx dy + \iint_{R_B} r_B(x, y) dx dy \right] \right\}}_{\text{Region Term}} \tag{4.4}$$

where $\tilde{R}$ is a partition of the image I ($\tilde{R} = \{R_A, R_B\}, R_A \cap R_B = \emptyset, R_A \cup R_B = I$). In the boundary term, C is the border between the regions R_A and R_B, $g(\cdot)$ is a monotonically decreasing function (e.g. Gaussian) serving as a penalty function of the pixels on the border. In the region term, the functions $r_A(\cdot)$ and $r_B(\cdot)$ are the two penalty functions that assign an error if the wrong pixel is assigned to regions R_A or R_B.

Since the goal is to minimize the energy function, for a given class we would like to design a penalty function that satisfies: when the possibility of a pixel belonging to that class becomes larger, the output of such a penalty function will be smaller, and vice versa. In addition, the output of the penalty function should be larger than 0. Thus, the two regional penalty functions for each class can be designed as follows:

$$r_A(x, y) = F_A(P(x, y)), F_A(z) = \frac{1}{1 + \exp(-k_r z)}, k_r > 0 \tag{4.5}$$

$$r_B(x, y) = F_B(P(x, y)), F_B(z) = 1 - \frac{1}{1 + \exp(-k_r z)}, k_r > 0 \tag{4.6}$$

For the edge term, a similar penalty function can be designed:

$$g(x, y) = 1 - \frac{1}{1 + \exp(-k_g P_B(x, y))}, k_g > 0 \tag{4.7}$$

It is worth noting that such forms have the advantage of symmetry, which constrain the penalty values within $[0, 0.5]$ for the pixels belonging to the correct class, while constrain the penalty values within $[0.5, 1]$ for the pixels belonging to the wrong class. The performance of two penalty functions $F_A(z)$ and $F_B(z)$ is shown in Figure 4.1.

It is commonly known that there are two ways to represent a deformable model: explicit (parametric) way and implicit (level set) way. Since implicit representation by level set has the advantage of no topology constraint and no parameterization, we will employ the level set method to optimize the objective function $E(\tilde{R})$. In two dimensions images, the level set method amounts to representing a closed curve C using an auxiliary function ϕ, called the level set function. C is represented as the zero level set of ϕ by $C = \{(x, y) | \, \phi(x, y) = 0\}$. In addition,

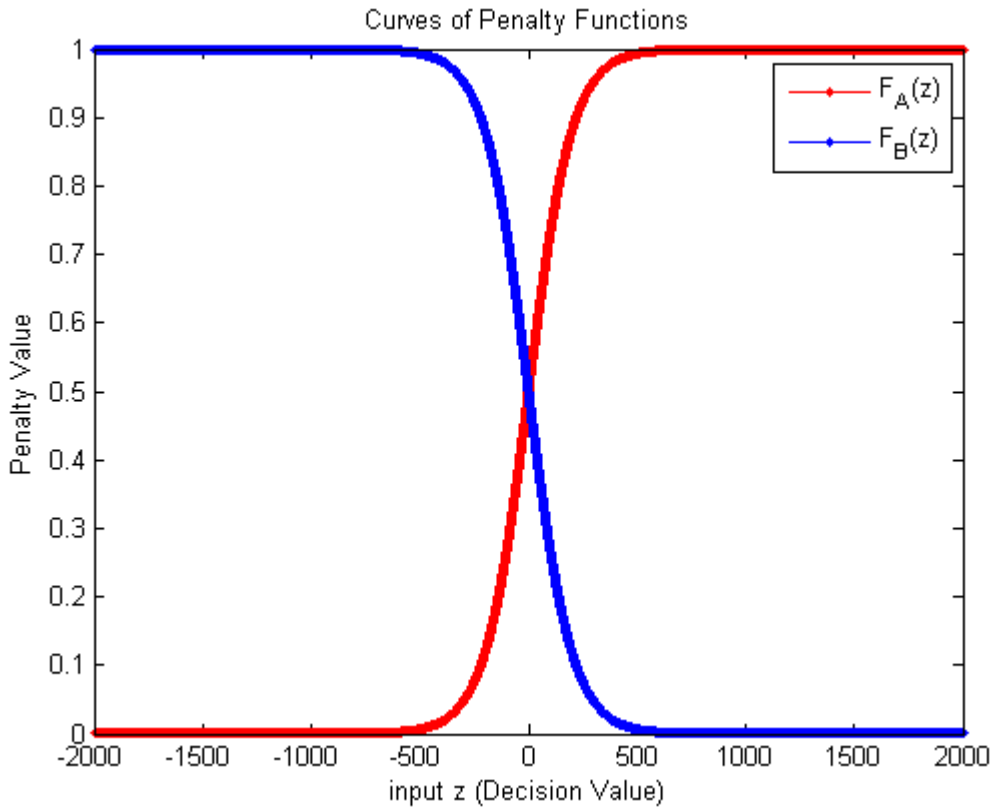


Figure 4.1: Performance of two penalty functions

when we use the level set $\phi(x, y)$ to represent the regions of two classes, where all the pixels whose $\phi(x, y) > 0$ belong to the background and whose $\phi(x, y) < 0$ belong to the foreground. A clear illustration is shown in Figure 4.2.

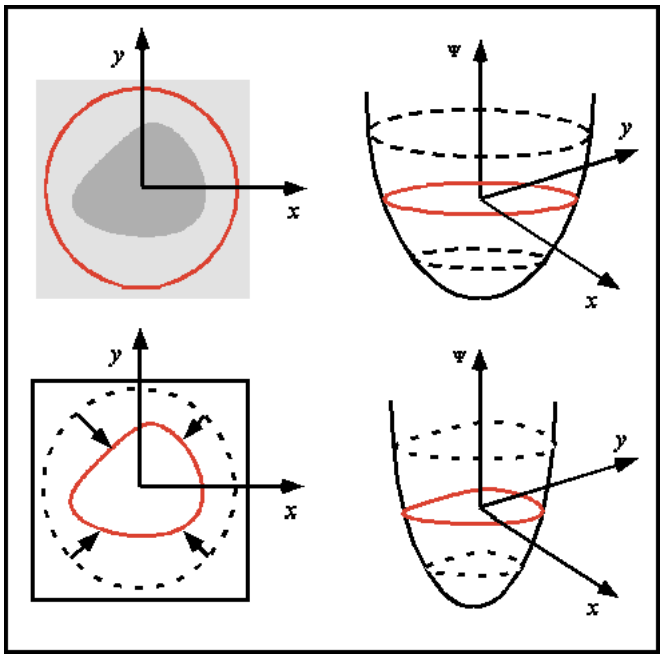


Figure 4.2: Representation of level set function

Thus, the boundary term in the energy function $E(\tilde{R})$ can be re-formulated as:

$$\begin{aligned} \int_0^1 g(C(s)) \cdot |C'(s)| ds &= \iint_I g(x,y) |\nabla H(\phi(x,y))| dxdy \\ &= \iint_I g(x,y) \delta(\phi(x,y)) |\nabla(\phi(x,y))| dxdy \end{aligned} \tag{4.8}$$

where $H(\cdot)$ is the Heaviside function.

For the regional term, assume that the region R_A corresponds to the background, while the region R_B corresponds to the foreground, the second term of the energy function can be represented as:

$$\begin{aligned} &\iint_{R_A} r_A(x,y) dxdy + \iint_{R_B} r_B(x,y) dxdy \\ = &\iint_I r_A(x,y) H(\phi(x,y)) dxdy + \iint_I r_B(x,y)(1 - H(\phi(x,y))) dxdy \end{aligned} \tag{4.9}$$

Finally, the energy function $E(\tilde{R})$ can be represented by level set as follows:

$$E(\phi) = \alpha \iint\limits_{I} g(x,y) \left|\nabla H(\phi)\right| dxdy + \beta \left[\iint\limits_{I} r_A(x,y) H(\phi) dxdy + \iint\limits_{I} r_B(x,y)(1 - H(\phi)) dxdy \right] \tag{4.10}$$

The minimization of the energy function $E(\phi), \phi = \phi(x,y)$ can be solved by the standard level set method, where we denote by $\frac{\partial \varepsilon}{\partial \phi}$ the Gateaux derivative (or first variation) [168] of the functional $\varepsilon(\phi)$ ($E(\phi(x,y)) = \iint\limits_{I} \varepsilon(\phi(x,y))dxdy$). In addition,the following evolution equation can be used to find the optimal $\phi(x,y)$ that minimizes the functional $\varepsilon(\phi)$ via gradient descent method [168]:

$$\frac{\partial \phi}{\partial t} = -\frac{\partial \varepsilon}{\partial \phi} \tag{4.11}$$

Thus, the Gateaux derivative (first variation) of the functional ε can be written as:

$$\frac{\partial \varepsilon}{\partial \phi} = -\alpha \delta(\phi) div(g \frac{\nabla \phi}{|\nabla \phi|}) + \beta \delta(\phi)(r_A - r_B) \tag{4.12}$$

Finally, the evolution equation can be written as follows:

$$\frac{\partial \phi}{\partial t} = \alpha \delta(\phi) div(g \frac{\nabla \phi}{|\nabla \phi|}) + \beta \delta(\phi)(r_B - r_A) \tag{4.13}$$

In addition, we propose a variant model $E_2(\tilde{R})$ by adding a weighted object area constraint to the previous objective function $E(\tilde{R})$, which helps to reduce the area size of object regions. This new objective function can be represented as follows:

$$E_2(\tilde{R}) = E(\tilde{R}) + \lambda \iint\limits_{R_B} g(x,y) dxdy \tag{4.14}$$

Then, we follow the standard level set algorithm step by step as shown above, and represent $E_2(\tilde{R})$ in level set form:

$$E_2(\phi) = E(\phi) + \lambda \iint\limits_{I} g(x,y)(1 - H(\phi(x,y))) dxdy \tag{4.15}$$

Finally, we can show the evolution equation for the variant model as follows:

$$\frac{\partial \phi}{\partial t} = \alpha \delta(\phi) div(g \frac{\nabla \phi}{|\nabla \phi|}) + \beta \delta(\phi)(r_B - r_A) + \lambda g \delta(\phi) \tag{4.16}$$

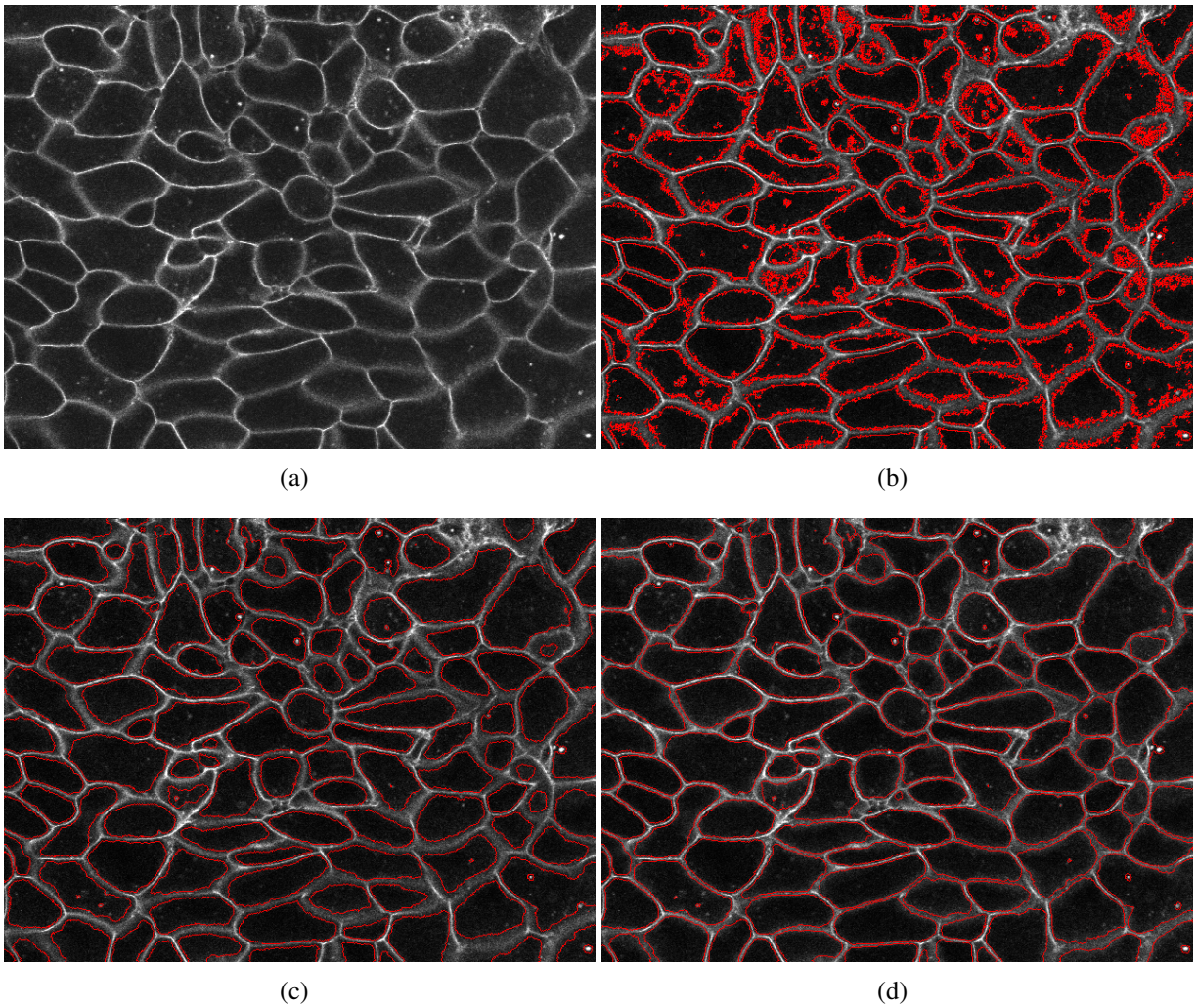
4.3 Experimental results

In this section, we will evaluation the region optimization performance of the two deformable models proposed above. We randomly choose a fluorescence image of membrane and the goal is to provide an accurate segmentation of membrane tissue regions from this image. As discussed above, before utilizing the proposed deformable models, we first implement a pixel classification process following the standard algorithm [167] in chapter 2, where with the ground truth from another image given (in which membrane tissue regions have been manually labeled), we choose a 5×5 window as the neighborhood size, and train a SVM classifier to classify pixels whether belonging to "membrane" tissue class or "non-membrane" tissue class. The output will be a decision or confidence value $P(x, y)$ for each pixel $p(x, y)$, where $P(x, y) > 0$ means such a pixel belongs to "membrane" class, while $P(x, y)$ means the pixel belongs to "non-membrane" class. In addition, a pixel classification process is implemented to distinguish whether the pixel belongs to the "borders of membrane" or not. The process is similar to the previous one, where the borders of membrane is manually labeled, and a 7×7 window is used for neighborhood size. A SVM classifier is trained, which provides each pixel $p(x, y)$ a decision value $P_B(x, y)$ ($P_B(x, y) > 0$ means the pixel $p(x, y)$ belongs to border, while $P_B(x, y)$ means it does not.) After we get the information of both $P(x, y)$ and $P_B(x, y)$, we test the proposed two deformable models, and note that we follow the numerical calculation framework in [37], where a penalty term is added to the energy function so that re-initialization at each update iteration can be prevented. In Table 4.1, we show the values of parameters concerned in the proposed models.

Table 4.1: Parameters in Our Level Set Models

parameter	α	β	λ	k_r	k_g
value	0.7	1.5	0.45	8	8

In Figure 4.3, we will compare the region optimization performance of two deformable models proposed above. In Figure 4.3(a), we show the original image of cell membrane (obtained from Dr. Alexandre Cunha's group), while in Figure 4.3(b), we show the segmented regions of membrane by grouping the pixels with the same label directly, where red contours delineate the borders of segmented regions. Obviously, the segmented regions have jagged borders with many noise regions inside, which makes the final segmentation result unsatisfactory. In Figure 4.3(c), we show the improved result obtained by our first proposed deformable model (without the weighted area constraint). Apparently, the segmented region result has been much better improved, in which the borders of segmented regions are smoothing, and most noisy regions have been removed. However, we observed that some non-membrane tissue regions have been mis-classified as the membrane tissue regions. Thus, we applied our second proposed model which added a weighted area constraint term, and the result is shown in Figure 4.3(d). Apparently, we can see that membrane tissue regions have been shrinked, which better delineates the real membrane tissue regions.



(a) (b)

(c) (d)

Figure 4.3: Comparison of region optimization performance between different methods. (a) original image; (b) segmented regions by grouping the pixels with the same label directly; (c) segmented regions by the proposed model 1 (without the weighted area constraint); (d) segmented regions by the proposed model 2 (with the weighted area constraint)

4.4 Conclusion and discussion

In this chapter, we discuss the algorithms for region optimization, which is regarded as a post-processing step after pixel classification. This is an important step in the whole segmentation system, since a common requirement for the segmented regions is to have smooth borders/surfaces, it is necessary to optimize the classification result before the final output, however, the segmented regions, if directly obtained from the pixel level classification procedure, are not always satisfactory, as they have jagged borders with holes inside, together with many noisy regions. We consider the deformable model as the tool for region optimization, due to the fact that it has the advantages of border smoothing and sub-pixel accuracy. Thus, we proposed two deformable models which utilize the decision value information obtained from previous pixel classification step for region optimization, and we validate these two models on fluorescence image of membrane. We have obtained accurate segmentation result of membrane, which shows that our proposed models work well. In addition, we will extend our current model to the multiple classes case and extend our model to 3D case, which will be our future work.

Chapter 5

Template Matching: A General Nuclei Segmentation Framework

For biomedical problems where a strong prior on shape is available (e.g. detecting nuclei in an image), a "stiff" deformable approach can be used to capture contiguous regions in an input image. Thus, in this chapter, we will investigate and propose a general segmentation system for biomedical applications with a strong prior shape information available [46, 97]. Here we explain that a strong prior shape information means that the shapes/structures of objects to be segmented can be estimated statistically from the given samples, e.g. cell nuclei, human brain, bones, etc. In this chapter, we will focus on the segmentation of nuclei. This task is very necessary and important for many scientific and clinical applications due to the fundamentally important role of nuclei in cellular processes and diseases.

5.1 Review of nuclei segmentation methods

Segmenting cell nuclei from microscopy images is an important image processing task necessary for many scientific and clinical applications due to the fundamentally important role of nuclei in cellular processes and diseases. Given a large variety of imaging modalities, staining procedures, experimental conditions, etc., many computational methods have been developed and applied to cell nuclei segmentation in 2D [16, 17, 100, 169, 170, 171, 172, 173, 174] and 3D images [175, 176, 177, 178, 179]. Thresholding techniques [1, 136], followed by standard morphological operations, are amongst the simplest and most computationally efficient strategies. These techniques, however, are inadequate when the data contains strong intensity variations, noise, or when nuclei appear crowded in the field of view [172, 180] being imaged. The watershed method is able to segment touching or overlapping nuclei. Direct use of watershed algorithms, however, can often lead to over segmentation artifacts [9, 17]. Seeded or marker controlled watershed methods [16, 17, 100, 171, 172, 181, 182] can be utilized to overcome such limitations. We note that seed extraction is a decisive factor in the performance of seeded watershed algorithms. Missing or artificially added seeds can cause under or over segmentation. Different algorithms for extracting seeds have been proposed. In [182] for example, seeds are extracted using a gradient vector field (GVF) followed by Gaussian filtering. Jung and Kim [173] proposed to find

optimal seeds by minimizing the residual between the segmented region boundaries and the fitted model. In addition, various of post-processing algorithms have been applied to improve the segmentation quality. For example, morphological algorithms (e.g. dilation and erosion)[172] can be used iteratively to overcome inaccuracies in segmentation. In [183] learning-based algorithms were used for discarding segmented regions deemed to be erroneous. Similar ideas using neural networks can be seen in [184].

When nuclei do not appear clearly in the images to be segmented (e.g. nuclear borders are not sharp enough or when a significant amount of noise is present), active contour-based methods [174, 176, 177, 185, 186, 187, 188, 189], especially those implicitly represented by level set [176, 177, 189], have been proposed to overcome some of these limitations successfully. As well known, the level set framework is well suited for accurate delineation of complicated borders and can be easily extended to higher dimensional datasets. Ortiz De Solorzano et al. [176], for example, proposed an edge-based deformable model that utilizes gradient information to capture nuclear surfaces. Considering that strong gradients at object boundaries may be blurred and the noise and intracellular structures may also show strong gradients, Mukherjee et al. [190] proposed a level set model that also incorporates a region term using the likelihood information for segmentation of leukocytes with homogeneous regions. In segmenting cells in culture or in tissue sections, Dufour et al.[177] proposed a multi-level deformable model incorporating both a gradient term and a region term, adopted from Chan and Vese model [38], to segment cells with ill-defined edges. In [96], Yan et al. also proposed a similar multilevel deformable model to segment RNAi fluorescence cellular images of drosophila. In [189], Cheng and Rajapakse utilized the Chan and Vese model [38] to obtain the outer contours of clustered nuclei, employing a watershed-like algorithm to separate clustered nuclei. Similarly, Nielsen et al [174] have described a method for segmenting Feulgen stained nuclei using a seeded watershed method, combined with a gradient vector flow-based deformable model method [162]. Considering that some nuclei may appear to overlap in 2D images, Plissiti and Nikou [191] proposed a deformable model driven by physical principles, helping to delineate the borders of overlapping nuclei. In [192], Dzyubachyk et al. proposed a modified region-based level set model, which addresses a number of shortcomings in [177] as well as speeds up computation. In order to reduce the large computational costs of variational deformable models, Dufour et al. [193] proposed a novel implementation of the piece-wise constant Mumford-Shah functional using 3D active meshes for 3D cell segmentation.

Besides the methods mentioned above, several other approaches for segmenting nuclei based on filter design [194, 195], multi-scale analysis [196], dynamic programming [197], Markov random fields [198], graph based methods [45, 199, 200] and learning based strategies [56, 201, 202, 203, 204] have been described. As new imaging modalities, staining techniques, etc., are developed, however, many existing methods specifically designed for current imaging modalities may not work well. Below we show that the application of some such methods can fail to detect adequate borders, or separate touching or overlapping nuclei, in several staining techniques. Therefore considerable resources have to be spent to modify existing methods (or developing entirely new segmentation methods) to better suit the new applications.

Here we describe a generic nuclear segmentation method based on the combination of template matching and supervised learning ideas. Our goal is to provide a method that can be used effectively for segmenting nuclei for many different types of cells imaged under a variety of stain-

ing or fluorescence techniques. We aim to guarantee robust performance by allowing the method to 'calibrate' itself automatically using training data, so that it will adapt itself to segmenting nuclei with different appearances (due to the staining techniques for example) and shapes. The method is also 'constrained' to produce smooth borders. Finally, given that the objective function used in the segmentation process is the normalized cross correlation (NCC), the method is also able to better handle variations in illumination within the same image, as well as across images. We note that template matching-based methods for image segmentation have long been used for segmenting biomedical images. One prominent example is the brain segmentation tool often used in the analysis of functional images [205]. When segmenting nuclei from microscopy images, contour templates have also been used [201, 202]. Here we utilize similar ideas with some adaptations. First, our approach is semi-automated in that it first seeks to learn a template and statistical model from images delineated by the user. The model is built based on estimating a 'mean' template, as well as the deformations from the template to all other nuclei provided in the training step. After this step, any image of the same modality can then be segmented via a template-based approach based on maximization of the normalized cross correlation between the template estimated from the input images and the image to be segmented. We describe the method in detail in the next section, and compare it to several other methods applied on different datasets in section IV. Finally, we note that our method is implemented in MATLAB computer language [206]. The necessary files can be obtained through contact with the corresponding author (GKR).

5.2 Methodology

Given the large variation in appearance of nuclei in microscopy images, a completely automated (unsupervised) approach for segmenting nuclei from arbitrary images may be difficult to obtain. We therefore focus on a semi automated approach, depicted in Figure5.1, where the idea is to first construct a statistical model for the mean texture and most likely variations of shape to be found in the dataset to be segmented from hand-delineated images. Segmentation of any image of similar type is then achieved by maximizing the normalized cross correlation (NCC) between the model and the local image region. Part A outlines the training procedure whereby the user utilizes a simple graphical user interface to isolate several nuclei samples, which are then used to build the statistical model. Part B outlines the actual segmentation procedure, which proceeds to first find an approximate segmentation (seed detection) of an input image by matching the statistical model with the given image, and then produces a final segmentation result via non-rigid registration.

5.2.1 Training

As outlined in part A of Figure5.1, we utilize a simple graphical user interface to enable an operator to manually delineate rectangular sub-windows each containing one nucleus sample from an image of the modality he or she wishes to segment. It is required by our system that each sub-window contains only one nucleus, and recommended that the set of sub-windows contain a variety of shapes and textures (small, large, bent, irregular shaped, hollow, etc.), since more

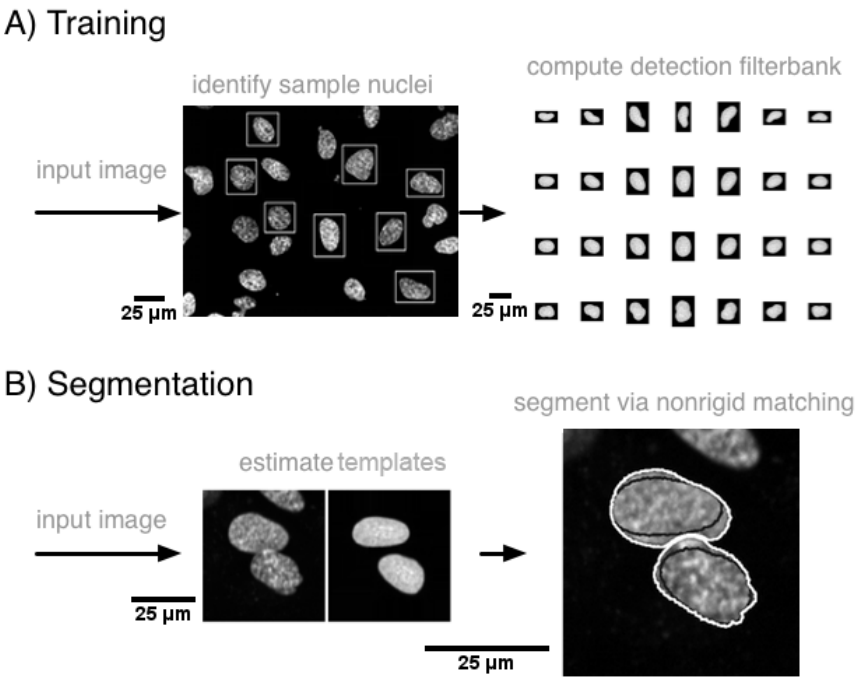


Figure 5.1: Overview of nuclear segmentation approach. Part A outlines the training procedure, which utilizes sample nuclei manually identified by the user to build a statistical model for the texture and shape variations that could be present in the set of nuclei to be segmented. The model is then sampled to form a detection filter-bank. Part B outlines the actual segmentation procedure which utilizes the detection filter-bank to produce a rough segmentation, and then refines it using non-rigid registration based on the normalized cross correlation.

variations present in the input images will translate into more variations being captured by the model. We note that it is not necessary for the user to provide the detailed outline for the nucleus present in each window. Rather, a rectangular bounding box suffices. In our implementation, given N such rectangular sub-windows, which can be of different sizes, each sub-window containing one nucleus from the training set, we first pad each sub-window image by replicating the border elements so as to render each sub-window of the same size (in terms of number of pixels in each dimension). The amount of padding applied to each sub-window is the amount necessary for that sub-window to match the size of the largest rectangular sub-window in the set. The set of sub-windows are then rigidly aligned to one sub-window image from the set (picked at random) via a procedure described in earlier work [207]. As a result, the major axis of nuclei samples are aligned to the same orientation. In this case, we choose the normalized cross correlation (NCC) as the optimization criterion for measuring how well two nuclei align and include coordinate inversions (image flips) in the optimization procedure.

The set of N rigidly aligned sub-windows, denoted as $I_1, \cdots, I_N$ from now on, is then used to estimate a template that will represent an "average" shape as well as texture for this set. Several procedures can be used for this purpose. In this work we choose the procedure outlined in Heitz et al. [208] where the idea is to iteratively deform all nuclear images (sub windows) towards a template image that is closest (in the sense of least deformation) to all other images in the set. Figure5.2 contains a diagram depicting the procedure we use. The procedure depends on the computation of a nonrigid map that aligns two images I_i, and I_j, via $I_i(f_i(\mathbf{x})) \approx I_j(\mathbf{x})$, with $\mathbf{x}$ an input coordinate in the image grid Ω, and a nonrigid mapping function $f_i(\mathbf{x}) : \Omega \rightarrow \mathcal{R}^2$. In our approach, the nonrigid registration is computed via maximization of the normalized cross correlation cost function, which is described in detail in the appendix. Given the ability to non-rigidly align two nuclear images, the template estimation procedure consists of choosing a sub-window image from the set at random and denoting it $I_0^k(\mathbf{x})$. Then, starting with the iteration $k = 1$:

1. Non-rigidly register I_0^k to each sub-window image $I_i, i = 1, 2, ..., N$ such that $I_i(f_i(\mathbf{x})) \approx I_0^k(\mathbf{x})$.
2. Calculate a temporary average shape template $\Psi^k(\mathbf{x}) = I_0^k(\bar{f}^{-1}(\mathbf{x}))$, with $\bar{f} = \frac{1}{N}\sum_{i=1}^{N} f_i$, and f^{-1} the inverse of the transformation function f (which we compute with Matlab's 'griddata' function).
3. Compute the average texture on the same average shape template above by first registering each sub-window image in the set to $\Psi(\mathbf{x})$ (i.e. $I_i(f_i'(\mathbf{x})) \approx \Psi^k(\mathbf{x})$), and update the template via $I_0^{k+1}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} I_i(f_i{}'(\mathbf{x}))$.
4. Compute error $= \|I_0^{k+1} - I_0^k\|^2$ (sum of squared errors). If error $< \epsilon$ stop, otherwise set $k = k + 1$ and go to step 1.

The end result is an image $I_0(\mathbf{x})$ that represents an average template (both in the sense of shape and texture), as well as a set of spatial transformations that map each sub-window image to the final template via $I_i(f_i(\mathbf{x})) \approx I_0(\mathbf{x}), i = 1, \cdots N$. We next apply the principal component analysis (PCA) technique [209] to derive a statistical model for the possible variations in the

shape of the sample nuclei. We encode each spatial transformation $f_i(\mathbf{x})$ as a vector of displacements via $\mathbf{v}_i = [f_i(\mathbf{x}_1), ..., f_i(\mathbf{x}_L)]^T$, with L the number of pixels in each image. Thus, the mean and the covariance of the set of spatial displacements $\mathbf{v}_1, \cdots, \mathbf{v}_N$ are:

$$\bar{\mathbf{v}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{v}_i \tag{5.1}$$

$$\mathbf{C} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{v}_i - \bar{\mathbf{v}})(\mathbf{v}_i - \bar{\mathbf{v}})^T \tag{5.2}$$

Using the PCA method, the principal deformation modes are given by the eigenvectors $\mathbf{q}_p, p = 1, 2, 3, ...$ of the covariance matrix $\mathbf{C}$ satisfying $\mathbf{C}\mathbf{q}_p = \lambda_p\mathbf{q}_p$. A statistical model for the variations in shape is obtained by retaining the top eigenvalues and eigenvectors corresponding to 95% (this percentage chosen arbitrarily) of the variance in the dataset. This means that the number of eigenvectors used in each segmentation task (imaging modality) will depend on how much variability is present in the (training) dataset. In cases where variability is large, more eigenvectors will be necessary. In cases when variability is small, a small number of eigenvectors will be used. In all cases, the accuracy of the PCA reconstruction procedure is set to 95% (of the variance). The model can be evaluated by choosing an eigenvector $\mathbf{q_p}$ and calculating $\mathbf{v}_{p,b_p} = \bar{\mathbf{v}} + b_p\mathbf{q_p}$, where b_p is a mode coefficient. The corresponding template is obtained by re-assembling $\mathbf{v}_{p,b_p}$ into a corresponding spatial transformation f_{p,b_p}, and computing $I_0(f^{-1}_{p,b_p}(\mathbf{x}))$.

In our approach, the statistical model is evaluated for $b_p \in [-2\sqrt{\lambda_p}, 2\sqrt{\lambda_p}]$ in intervals of $\sqrt{\lambda_p}$. The result of this operation is a set of images obtained by deforming the mean template and representing nuclear configurations likely to be encountered in data to be segmented. In addition, this set of images is augmented by including rotations (rotated every 30 degrees, totaling 7 orientations in our implementation) as well as variations in size (2 in our implementation). Finally, we discard the top 1% and bottom 1% (in the sense of area) of the templates to avoid potentially segmenting structures that would be too small or too large to be considered as nuclei. A data-dependent way of choosing this threshold is also described in the discussion section of this chapter. The reason being that templates that are too small may cause over segmentation, while templates that are too large may merge nuclei that are close to each other. Figure5.1 (top right) contains a few examples of template images generated in this way for a sample dataset. We denote the set of template images generated in this way as the "detection filterbank" to be used as a starting point for the segmentation method described in the next subsection.

Segmenting the mean template: our procedure depends the mean template, computed as estimated above, on being segmented in the sense that pixels in the foreground and background are known. Though many automated methods can be considered for this step, we choose to utilize a rough contour manually provided by an user. The contour is then refined utilizing a level set approach [37]. The advantage is that such a process can be repeated until a satisfactory segmentation result has been made by the user. Figure5.3 shows the outline of the procedure.

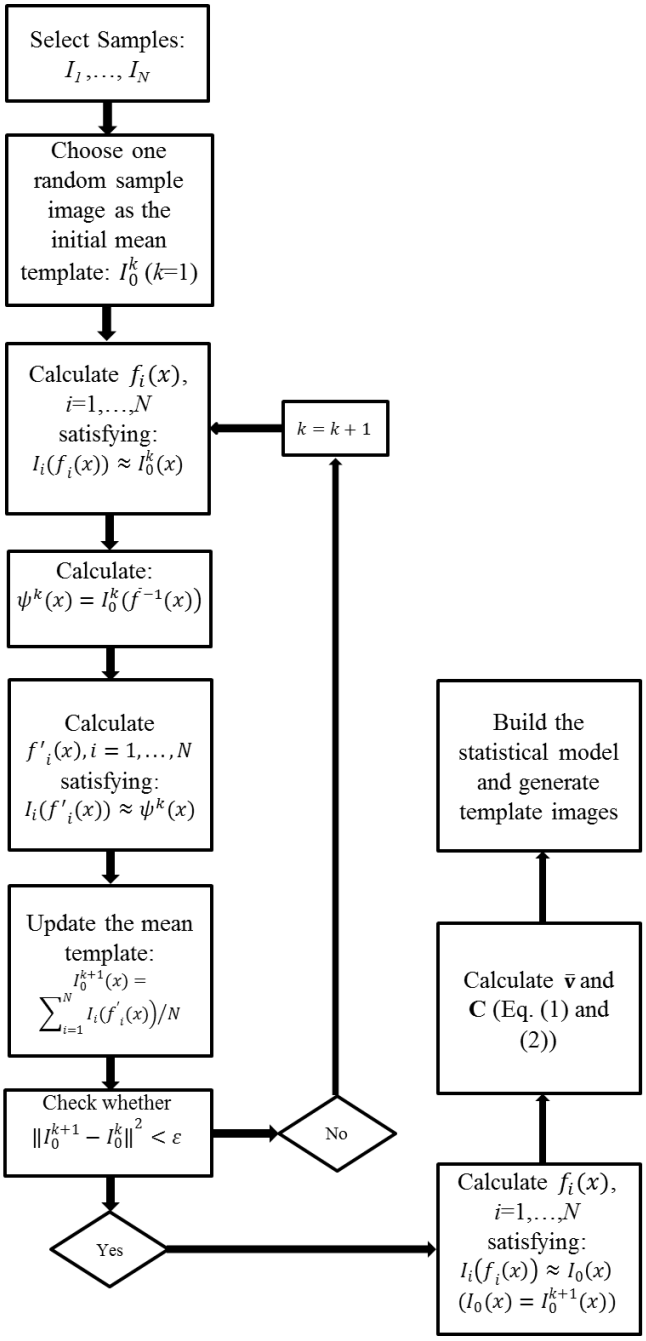


Figure 5.2: Diagram depicting training procedure.

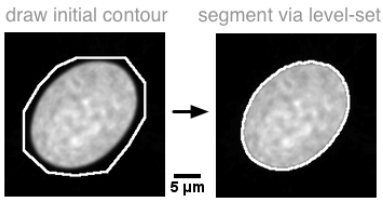


Figure 5.3: The mean template image must be segmented before the segmentation algorithm based on NCC maximization can be utilized. We utilize a semi automated approach wherein an user draws an initial contour, and a level set-based algorithm refines it to accurately match its borders.

5.2.2 Segmentation

Our segmentation algorithm is based on the idea of maximizing the NCC between the statistical model for a given dataset (its construction is described in the previous subsection) and local regions in an input image to be segmented. The first step in such a procedure is to obtain an approximate segmentation of an input image, here denoted as $J(\mathbf{x})$, by computing the NCC of the input image against each filter (template image) in the detection filterbank to obtain an approximate segmentation. To that end, we compute the NCC between each filter $W_p(\mathbf{x})$, $p = 1, \cdots, K$ and the image to be segmented via:

$$\gamma_p(\mathbf{u}) = \frac{1}{N_c} \sum_{i=1}^{N_c} \frac{\sum_{\mathbf{x}} J^i(\mathbf{x}) W_p^i(\mathbf{x}-\mathbf{u})}{\bar{J}^i(\mathbf{u})\bar{W}_p^i}. \tag{5.3}$$

where $\bar{W}_p^i = \left(\sum_{\mathbf{x}} \left(W_p^i(\mathbf{x})\right)^2\right)^{\frac{1}{2}}$, and $\bar{J}^i(\mathbf{u}) = \left(\sum_{\mathbf{x}\in\Omega_{\mathbf{u}}} \left(J^i(\mathbf{u})\right)^2\right)^{\frac{1}{2}}$, with $\Omega_{\mathbf{u}}$ denoting the neighborhood around $\mathbf{u}$ of the same size as filter W_p. We note that N_c is the number of channels in each image (e.g. 1 for scalar images and 3 for color images). A detection map denoted M is computed as $M(\mathbf{u}) = \max_p \gamma_p(\mathbf{u})$. We note that the value of the cross correlation function γ above is bound to be in the range $[-1, 1]$. We also note that the index p that maximizes this equation also specifies the template W_p that best matches the region $\mathbf{u}$ and is used later as a starting point for the deformable model-based optimization.

The detection map $M(\mathbf{u})$ is mined for potential locations of nuclei using the following two principles: (1) only pixels whose intensities in M are greater than a threshold μ are of interest. (2) The centers of detected nuclei must be at least a certain distance far away from each other. This helps to prevent, for example, two potential locations from being detected within one nucleus, causing over segmentation. These two principles can be implemented by first searching for the highest response in M. Subsequent detections must be at least a certain distance from the first. This is done by dilating the already detected nuclei (remember, the filtering step above not only defines regions where nuclei might be located, but also the rough shape of each). This process is able to detect nuclei of different shapes due to simulated templates of various shapes and orientations generated in the previous step, and is repeated until all pixels in the thresholded detection image M have been investigated. We note again that each detected pixel in M has its associated best matching template from the detection filterbank. Therefore, this part of the

algorithm not only provides the location of a nucleus, but also a rough guess for its shape (see bottom middle of Figure5.1) and texture.

Once an initial estimate for each nucleus in an input image is found via the procedure described above, the algorithm produces a spatially accurate segmentation by non-rigidly registering each approximate guess to the input image. The nonrigid registration nonlinearly adapts the borders of the detected template so as to accurately segment the borders of each nuclei in the input image. In addition, the nonrigid registration approach we use also is constrained to produce smooth borders. Details related to the nonrigid registration are provided in appendix A. Rather than optimizing all guesses at once, which could lead to difficulties such as a large number of iterations in our gradient ascent-type strategy, each nucleus is segmented separately.

Segmenting touching nuclei: An important feature of our template matching approach is that it is capable of segmenting touching nuclei without difficulties with a small modification of the procedure described above. In our method, if two (or more) nuclei are detected to be close to each other (e.g. the closest distance between their best matching templates' borders is smaller than 10 pixels), these nuclei are regarded as being potentially in close proximity to each other. If so, their best matching templates obtained from the filter bank procedure above are taken together under a sub-window and then non-rigidly registered to the same sub-window in the real image using the same optimization algorithm in the appendix. An example showing the segmentation of two nuclei in close proximity to each other is shown in Figure5.1 (bottom row). The left part of this portion of the figure shows the initial estimates from the filterbank-based estimation of candidate locations. The result of the nonrigid registration-based estimation of the contours for each nucleus is shown at the bottom right corner of the same figure. The black contours indicate the borders of the best matching templates (the initial guesses) and the white lines delineate the final segmentation result after nonrigid registration.

5.3 Experimental results

5.3.1 Data acquisition

We demonstrate our system applied to several different cell nuclei datasets: (1) a synthetic dataset BBBC004v1 generated with the SIMCEP simulating platform for fluorescent cell population images [210, 211]; (2) two real cell datasets (U2OS cells and NIH3T3 cells) acquired with fluorescence imaging [135]; (3) and a histopathology dataset obtained using thyroid tissue specimens with several different staining techniques. The primary goal for the simulated dataset is to obtain an accurate count for the number of nuclei in each field of view. Each simulated image contains 300 objects with different degrees of overlap probability (ranging from 0.00 to 0.60). The U2OS (48 images, each containing multiple nuclei) and NIH3T3 (49 images) cells were obtained with the Hoechst 33342 fluorescence signal, and the ground truth (including accurately delineated borders) is provided by experts [135]. Of these, the U2OS dataset is more challenging, with nuclei tending to be more varied in shape and more clustered together. The intensity of the NIH3T3 images, however, is more nonuniform than the U2OS dataset. In addition we apply our method to segmenting nuclei from histopathology images taken from tissue sections of thyroid specimens. Tissue blocks were obtained from the archives of the University of Pittsburgh Medical Center

(Institutional Review Board approval #PRO09020278). Briefly, tissue sections were cut at 5 μm thickness from the paraffin-embedded blocks and stained using three techniques. These include the Feulgen stain which stains deoxyribonucleic acids only. If no counterstaining is performed then only nuclei are visible demonstrating chromatin patterns as deep magenta hues shown in Figure 5.6(a). The second is a silver-based technique that stains the intranuclear nucleolar organizing regions (NORs) (black intranuclear dots) and counterstained with nuclear fast red which uses kernechtrot that dyes nuclear chromatin red (Figure 5.6(b)). The third is the same silver-based staining for NORs without counterstaining (Figure 5.6(c)). All images used for analysis in this study were acquired using an Olympus BX51 microscope equipped with a 100X UIS2 objective (Olympus America, Central Valley, PA) and 2 mega pixel SPOT Insight camera (Diagnostic Instruments, Sterling Heights, MI). Image specifications were 24 bit RGB channels and 0.074 microns/pixel, $118 \times 89 \mu m$ field of view. More details pertaining to the image acquisition process for this dataset are available in [212].

5.3.2 Experimental setup

We note that our system is able to work with grayscale (single color) images as well as with color images. Equation (4), in the Appendix section, allows color images to be used, while the method can also be used to segment 3D data by defining the inter products and convolutions utilized in equations (4) and (5) in three dimensions. In addition, we mention that for color images, each color channel (R, G and B) is equally weighted in the approach we described above. This allows for segmentation even in the case when the optimal color transformation for detecting nuclei is not known precisely (as is the case in many of the images shown). In cases where this information is known precisely, the approach we proposed can be used with only the color channel that targets nuclei, or with the image after optimal color transformation. In each experiment, k sample nuclei (k is arbitrarily chosen as 20 in our experiments) were chosen by the authors for the training process. All but one of the parameters remained constant for all experiments. The percent of variance retained in PCA analysis was set to (95%), $\epsilon = 0.01$ for the calculation of the average template, the step size in the gradient ascent procedure κ was set as 5×10^4, the scale number in the multi-scale strategy s was set as 2. For smoothing parameter σ in the gradient ascent procedure, a higher σ value helps to smooth the contour, while a lower σ value helps to better capture the real border of nuclei. In this work, σ was experimentally set as 1.5 (pixels). The only parameter that varied from dataset to dataset was the detection threshold μ. While a higher value of μ may miss some nuclei (e.g. out of focus), a lower value of μ may confuse noise and clutter for actual nuclear candidates. There are two ways to determine an appropriate value for detection threshold μ. When the ground truth (e.g. manual delineation of nuclei) for the training images is provided, μ value can be selected automatically by maximizing the dice metric $2 \times \frac{\#(GT \cap Detect(\mu))}{\#(GT)+\#(Detect(\mu))}$ [127] between the detections and provided ground truth. Here $\#(\cdot)$ counts the number of nuclei in different results, GT corresponds to the ground truth result, while $Detect(\mu)$ corresponds to the nuclei detection result with respect to μ. When ground truth is not available, an appropriate μ value has to be empirically selected by the user in order to detect most nuclei in the training images for each application or dataset. In the experiments shown below, ground truth was not used for selecting μ. Rather μ was empirically determined for

each dataset based on empirical experimentation with a given field of view (containing multiple nuclei) from the corresponding dataset.

For comparison, we choose several different types of algorithms commonly used for cell nuclei segmentation. These include the level set method (Chan and Vese model [38]), an unsupervised learning method (color K-means [133]), and the direct seeded watershed method, which uses a shape based method to separate clumped nuclei (implemented by CellProfiler[18]). Since the CellProfiler software [18] is only able to process 2D grayscale images, a typical choice is to convert the color histopathology image to grayscale image by forming a weighted sum of R, G, and B channels, which keeps the luminance channel ($grayscale = 0.2989 \times R + 0.5870 \times G + 0.1140 \times B$) [8]. In addition, we prefer to take the general approach of normalizing all image data to fit the intensity range of [0, 1] by scaling the minimum and maximum of each image (discounting outliers set at 1% in our implementation). Since the level set method and the K-means method may not be able to separate clumped nuclei very well, a common solution is to apply seeded watershed algorithm on the binary masks segmented from level set method and K-means methods, in which seeds are defined as the local maxima in the distance transformed images of binary masks [213]. Note that H-dome maxima [213] are calculated on the distance transformed images in order to prevent over-segmentation, and for different dataset, the H value is arbitrarily selected for the best performance. These techniques were chosen since they are similar to several of the methods described in the literature for segmenting nuclei from microscopy images [177, 192]. In the following sections, we will show both qualitative and quantitative comparisons of these methods.

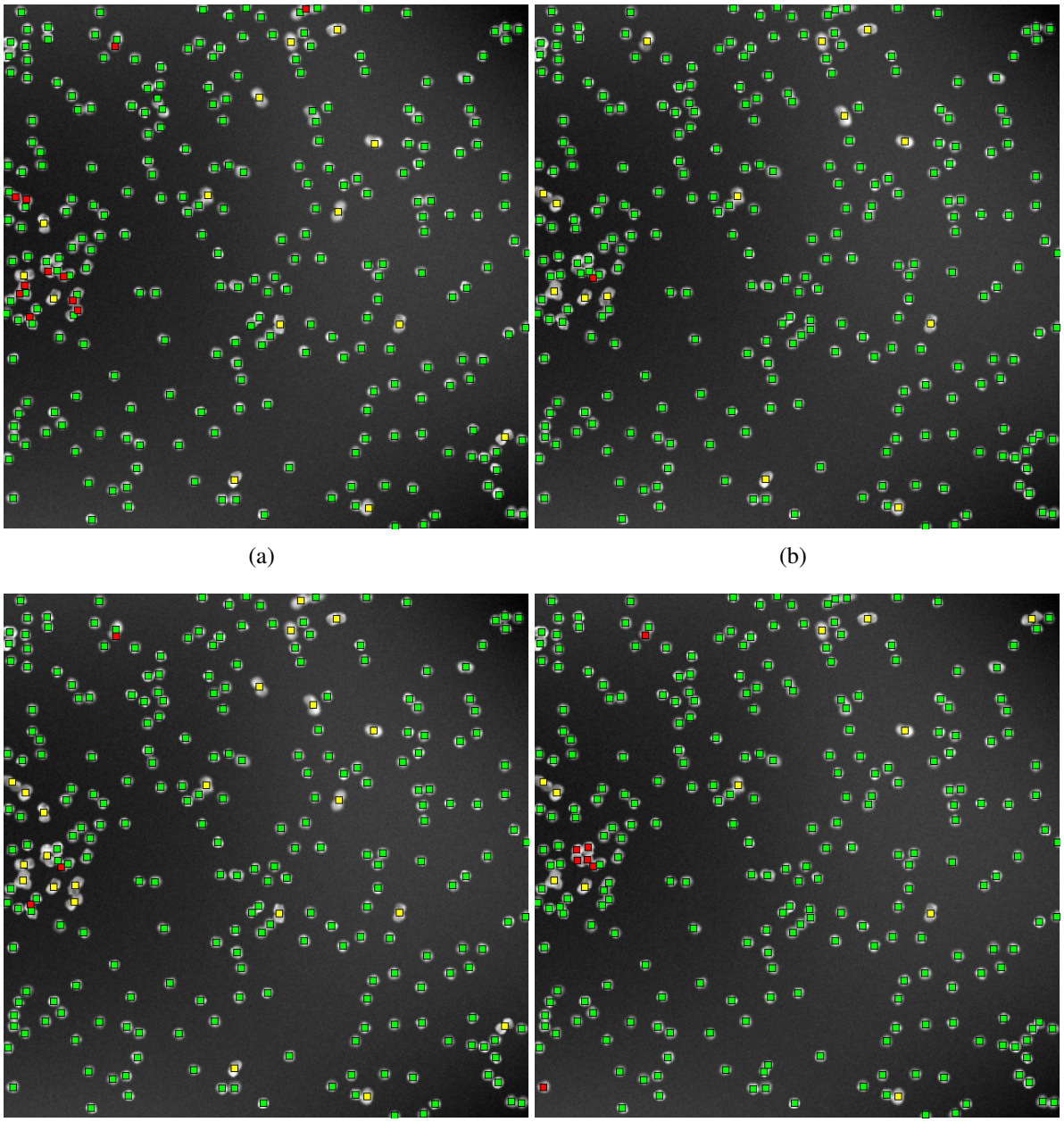
5.3.3 Qualitative evaluation

In Figure 5.4, Figure 5.5, Figure 5.6 and Figure 5.7, we compare the results for different types of datasets using different methods. In Figure 5.4, the results are obtained by different methods applied to the segmentation of synthetic nuclei with clustering probability set to 0.3. Note that we use green (in color), red, and yellow square dots to represent correct detections, missed detections, and spurious detections respectively. In Figure 5.5, the first column shows the sample segmentations of the U2OS data (under uniform illumination), and the second column shows the sample segmentations of the NIH3T3 data (under heterogeneous illumination), in which the white contours delineate the borders of segmented nuclei. The first row of Figure 5.5 corresponds to results computed using the approach we described in this chapter, the second row corresponds to level set method-based method, the third row corresponds to color K-means-based method, and the fourth row corresponds to direct seeded watershed method. In addition, we show the hand-labeled images of U2OS data and NIH3T3 data as the ground truth separately in Figure 5.5(i) and Figure 5.5(j) in the final row. In Figure 5.6, we show the segmentation results on sample histology images with different staining techniques, in which each column corresponds to a distinct staining technique (details have been described in the previous section), while each row corresponds to a distinct segmentation method (the row order is the same as Figure 5.5). In addition, in Figure 5.7, we show the result of yeast cells (obtained from Dr. Lages' group from UT Southwestern Medical Center [46]), which were observed with bright light under a 40x objective using a Nikon Ti-E2000 microscope. Cells are delimited either by a dark or white boundary.

From the comparison we can see that when nuclei samples are imaged clearly with distinct enough borders (for example, parts of image in Figure 5.4(a) and Figure 5.5(a)), all methods tested are able to achieve reasonable results. However, when noise or clutter is present, or when images are screened under uneven illumination (intensity inhomogeneities can be seen in Figure 5.5(a) and Figure 5.5(b)), most methods fail to segment nuclei very well (Figure 5.5(d), (f) and (h)). Comparatively, our template matching approach still performs well on these images. The improvements are pointed out by white arrows in Figure 5.5. In addition, our template matching approach can be naturally applied to higher dimensional data, as can other algorithms, such as RGB channel images (Figure 5.6(a), (b) and (c)), and achieve what can be visually confirmed as better segmentation results over the existing methods tested. The improvements are pointed out by black arrows in Figure 5.6. We note that in several locations (pointed out by arrows) our method performs better at segmenting cluttered nuclei. We also note that other methods often detect spurious locations as nuclei. Finally, we also note that our template matching approach is also much more likely (because it is constrained to do so) to produce contours that are more smooth and more realistic than the several other methods used for comparison.

5.3.4 Quantitative evaluation

We used the synthetic dataset described above to calculate the average count produced by each method. We also studied the performance as a function of the clustering probability for this simulated dataset. The result is shown in Table 5.1, where C.A. refers to "Count Accuracy", while O.P. refers to "Overlap Probability" of data at each column. For the fluorescence microscopy



(a) (b)

(c) (d)

Figure 5.4: Nuclei counting in synthetic images. Upper left: results of our template matching approach. Upper right: result obtained with level set method. Bottom left: results obtained with color K-means-based method. Bottom right: results obtained with seeded watershed method. Note that green square dots represent correct detections, red square dots represent missed detections, and yellow square dots represent spurious detections.

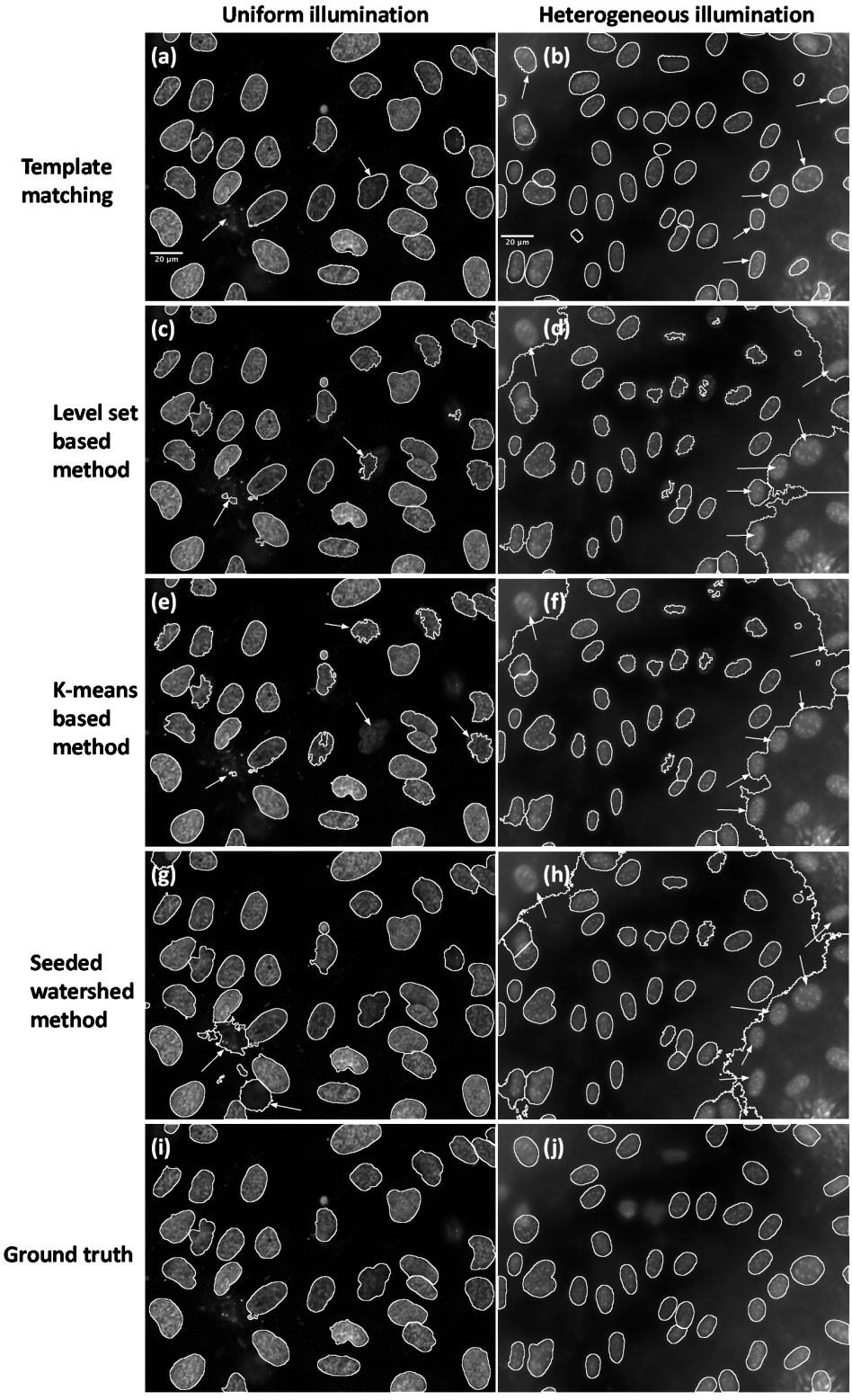


Figure 5.5: Nuclei detection and segmentation from different fluorescence images. Note that the improvements are pointed out by white arrows. First row: results obtained with our template matching approach. Second row: results obtained with level set based method. Third row: results of color K-means-based method. Fourth row: results of seeded watershed method. Last row: hand-labeled results as the ground truth. First column: results of U2OS fluorescence image under uniform illumination. Second column: results of NIH3T3 fluorescence image under heterogeneous illumination.

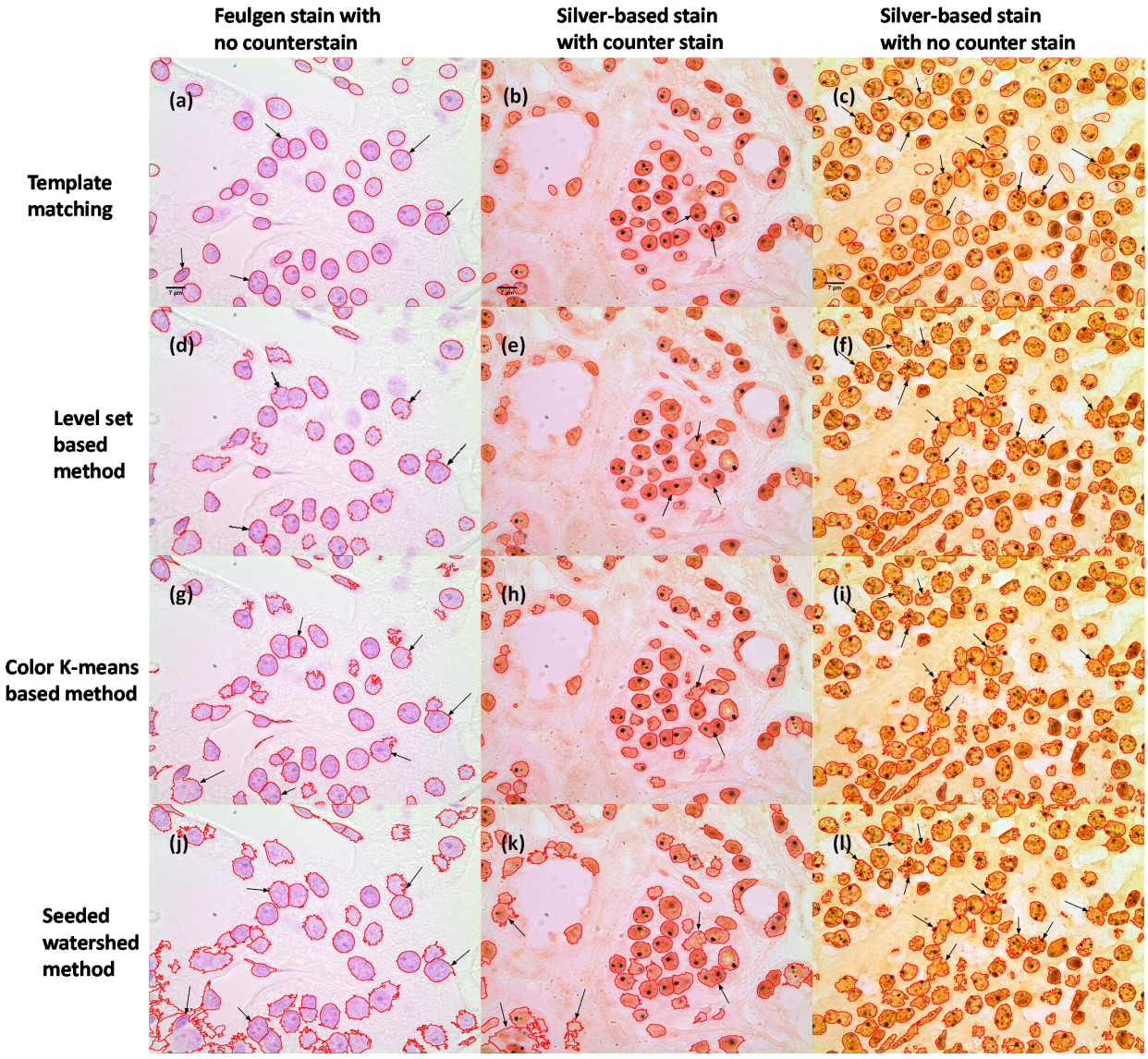


Figure 5.6: Nuclei segmentation from histopathology images with different staining. Note that the improvements are pointed out by black arrows. First row: results of our template matching approach. Second row: results of level set-based method. Third row: results of color K-means-based method. Fourth row: results of seeded watershed method.

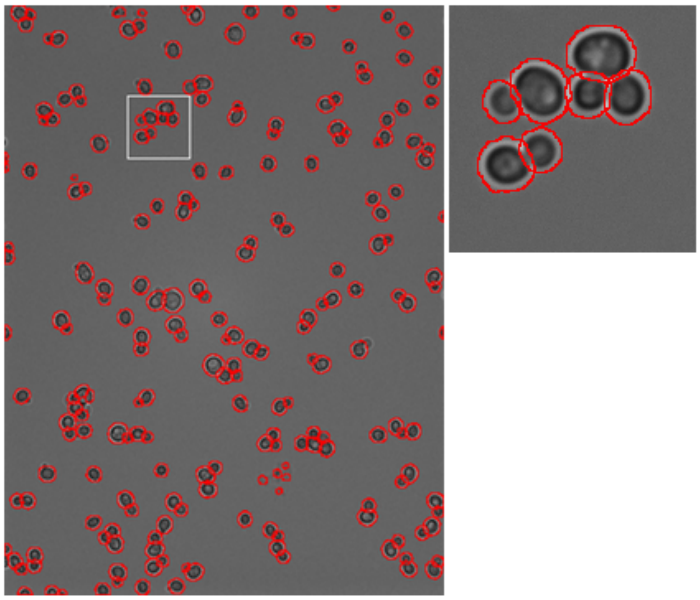


Figure 5.7: Segmentation result on cellular microscopy image

data (U2OS and NIH3T3), we follow the same evaluation procedure as documented in [135], including: (1) Rand and Jaccard indices (RI and JI), which are used to measure the fraction of the pairs where the two clusterings agree (higher means better); (2) two spatially-aware evaluation metrics: Hausdorff metric and normalized sum of distances (NSD) (smaller means better); (3) counting errors: split, merged, added, and missing (smaller means better).

We also compare the results of the methods discussed above together with two other methods: active masks [214] and a merging based algorithm [16], as well as a manual delineation result. In Table 5.2, for both U2OS and NIH3T3 data, we can see that although the Hausdorff metric values are quite high for our template matching approach, most segmentation metrics are comparable or better than many of the the existing algorithms. Our segmentation result also performs better than the manual delineation result (JI, Split for U2OS data, Split and Missing for NIH3T3 data) explored in [135]. More details pertaining to each method used in this comparison are available in [135]. The high Hausdorff metric can be explained by two reasons: (1) some bright noise regions are detected (especially in NIH3T3 dataset) and no morphological post-processing is used in our template matching approach and (2) we choose a relatively high threshold μ that discards some incomplete nuclei (small area) attached to the image border in some images. However, these incomplete nuclei are included in the manual delineated ground truth. The first reason may explain why the "Added" error for NIH3T3 dataset is much higher than for U2OS dataset. In addition, the second reason may also explain why, excluding the active masks method [214], our algorithm misses more of the U2OS cells (last column of this table). On the other hand, for the NIH3T3 image data which contains intensity heterogeneities, the method we propose misses the fewest nuclei. We also note that, for the Rand and Jaccard indices, the normalized sum of distances (NSD) metric, and splitting errors, for both U2OS and NIH3T3 dataset, our results are similar to or better than the best results produced by other methods (excluding the manual delineation result).

Table 5.1: Nuclei Counting Accuracy

Algorithms	C. A. (O.P.: 0)	C. A. (O.P.: 0.15)	C. A. (O.P.: 0.30)	C. A. (O.P.: 0.45)	C. A. (O.P.: 0.60)
Template matching	99.8%	86.5%	84.7%	80.6%	76.2%
Level Set [38]	99.4%	86.4%	83.1%	78.6%	71.2%
K-means [133]	**100.0%**	87.7%	84.3%	80.2%	72.5%
Seeded Watershed [181]	99.9%	**91.0%**	**88.1%**	**84.7%**	**78.4%**

Table 5.2: Quantitative Comparison of Nuclei Segmentation

Algorithm (U2OS/NIH3T3)	RI	JI	Hausdorff	NSD($\times$10)	Split	Merged	Added	Missing
Manual	95%/93%	2.4/3.4	9.7/12.0	0.5/0.7	1.6/1.0	1.0/1.2	0.8/0.0	2.2/3.2
Watershed (direct) [181]	91%/78%	1.9/1.6	34.9/19.3	3.6/3.7	13.8/2.9	1.2/2.4	2.0/11.6	3.0/5.5
Active Masks [214]	87%/72%	2.1/2.0	148.3/98.0	5.5/5.0	10.5/1.9	2.1/1.5	**0.4**/3.9	10.8/31.1
Merging Algorithm [16]	96%/83%	2.2/1.9	**12.9/15.9**	0.7/**2.5**	1.8/1.6	2.1/3.0	1.0/6.8	3.3/5.9
Level Set [38]	91%/81%	2.39/2.30	96.6/122.8	0.85/5.0	1.1/1.4	0.35/1.4	2.75/4.2	**0.85**/8.2
K-means [133]	90%/78%	2.36/2.35	94.6/100.6	1.05/6.15	1.56/**0.45**	**0.3/0.9**	2.6/**2.75**	1.6/17.4
Template matching	**95%/91%**	**2.50/2.72**	77.8/131.2	**0.64**/2.65	**0.58**/0.51	1.45/2.49	0.9/3.7	3.48/**2.8**

Finally, we also studied how the number of nuclei used for training affects the performance of the proposed method. This was done for both U2OS and NIH3T3 dataset, by randomly selecting nuclei (of different sample sizes), and then implementing the method as described above. We found that the performance of several different quantitative metrics, such as the Rand index, NSD, etc., do not vary significantly when different amounts/types of nuclei samples are used (data omitted for brevity).

5.4 Conclusion and discussion

In this chapter, we described a method for segmenting cell nuclei from several different modalities of images based on supervised learning and template matching. The method is suitable for a variety of imaging experiments given that it contains a training step that adapts the statistical model for the given type of data. In its simplest form, the method consists of building a statistical model for the texture and shape variations of the nuclei from the input of a user, and then segmenting arbitrary images by finding the instance in the model that best matches, in the sense of the NCC, local regions in the input images. We note that given an experimental setup, once

the training operation is completed, the method is able to segment automatically any number of images from the same modality. We have demonstrated the application of the method to several types of images, and results showed that the method can achieve comparable, and often times better, performance compared with the existing specifically designed algorithms. Our main motivation was to design a method for segmenting nuclei from microscopy images of arbitrary types (scalar, color, fluorescence, different staining, etc.). To our knowledge ours is the first method to apply a template matching approach which includes texture and shape variations to accurately delineating nuclei from microscopy images. In addition, to our knowledge ours is the first method to utilize a supervised learning strategy to build such a statistical model, that includes texture and shape variations in multiple channels, for detecting nuclei from microscopy images.

In a practical sense, our method provides three main contributions. First, its overall performance is robust across different types of data with little tuning of parameters. We have demonstrated this here by applying the same exact software (with the only difference in each test being the value for μ) to a total of six different imaging modalities and showing the method performs as well or better than all other methods we were able to compare against. The performance was compared quantitatively and qualitatively, using both real and simulated data. Comparison results with a total of six alternative segmentation methods are shown here. Other, simpler, segmentation methods were also used for comparison, including several thresholding schemes followed by morphological operations. The results of these were not comparable to many of the methods shown here. Therefore, we have omitted them for brevity. Secondly, amongst the methods we have tested in this manuscript, we show that our method is the only method (besides manual segmentation) that is capable of handling significant intensity inhomogeneities. This is due to the fact that we utilize the NCC metric in the registration-based segmentation process. The NCC metric is independent of the overall intensity of the local region of the image being segmented. Finally, we also mention that, amongst all methods tried, the template matching method we described produced noticeably more smooth and accurate borders with fewer spurious contours. This can be seen, for example, by close observation of Figure 5.6. The smoothness in the contours obtained by our method is primarily due to the fact that the statistical modeling we use includes only the main modes of variation in nuclear shape. These tend to be, topically, size, elongation, as well as bending (in addition to rotation). High order fluctuation in contours do occur in nuclei at times, but these do not occur as often as the ones already mentioned. We note that the method is still flexible enough to accurately segment nuclei that do not conform to these main modes of variation given the elastic matching procedure applied in the last step of the procedure.

We also note that our algorithm has several parameters including the percent of variance in PCA analysis, ϵ in the calculation of "average" template, σ, s, κ in the non-rigid registration procedure, and μ in the approximate segmentation procedure. The algorithm is not unduly sensitive to these, as the same fixed parameters were utilized in all six experiments (datasets) used in this chapter. The only parameter that was selected differently for each dataset was the detection threshold μ. When ground truth is available, we described a method to automatically choose the optimal μ for the given dataset. In addition, in our current implementation, we discard the top and bottom 1% (in size) of the generated templates, in an effort to reduce outlier detections. This percentage too could be made dataset dependent through a cross validation procedure, when the precise ground truth is present.

Finally, it is important to describe the computational cost of our template matching approach, which is also important in evaluating the performance of an algorithm. Our approach consists of a training stage and a testing stage, and is implemented in MATLAB 64 bit mode and tested on a PC laptop (CPU: Intel core i5 2.30GHz, memory: 8GB). The computational time for training a statistical model (560 simulated templates) from 20 nuclei samples (window size: 155×179), for example, is about 1.6 hours. Detecting and segmenting all cell nuclei (36 nuclei) from a fluorescence image (1030×1349) takes about 20 minutes (about half a minute per nucleus). We note however, that the computational time can often be significantly reduced by implementing the algorithm in a compiled language such as C, for example. In addition, we note that the computational time should be considered in context to alternative segmentation methods capable of producing results (albeit not as accurate) on similar datasets. The level set algorithm by Chan and Vese, which is used in a variety of other nuclear segmentation methods, takes even longer to compute on the same image (23 minutes) in our implementation (also in MATLAB). Finally, we note that the computational time of our algorithm can be decreased by utilizing a multi-scale framework. That is, instead of performing the filtering-based approach for detection in the original image space, we have also experimented with first reducing the size of the image (and templates) by two for the initial detection only (the remaining part of the method utilized the full resolution image). Thus we were able to reduce the total computation time for the same field of view to roughly 10.6 minutes. The accuracy of the final segmentation was not severely affected (data not shown for brevity). Future work will include improving the computational efficiency of this method by further investigation of multi-scale approaches, as well as faster optimization methods (e.g. conjugate gradient). Finally, we note again that the approach described above utilizes all color information contained in the training and test image. In cases where the nuclear stain color is known precisely, the approach can be easily modified to utilize only that color. In addition, many existing techniques for optimal color transformation [215] can also be combined with our proposed approach in the future for better performance.

Chapter 6

Conclusion and Discussion

The motivation for this Ph.D. research work comes out from the fact that although current segmentation methods have already performed adequately in the applications for which they were designed, these methods still have a couple of drawbacks: a large number of algorithms successfully used in real applications were specifically designed for the given application, thus, a significant amount of tuning and calibration is usually required before an algorithm that was specifically designed for one application can be used in another. Even then, in many cases, the chosen method may not perform satisfactorily. Therefore, a researcher faced with a new problem or application, must often spend considerable resources to modify (or develop a new) a reliable segmentation method capable of extracting the structures of interest for the given application.

The big goal is to propose a general purpose segmentation system, which can be applied to different biomedical applications with little tuning or calibration, and achieve comparable or even better results compared with the state-of-the-art methods specifically designed for each application. To be general, the proposed system should be able to segment a variety of biological objects with different structures (e.g. tissues, nuclei, cells, etc.) from different imaging modalities (MRI, CT, histology, etc). In addition, the system can be used with two or three-dimensional data, as well as scalar or vector valued (e.g. color) images. If the proposed aim can be achieved, such a system can be utilized as a general segmentation tool, which can be widely used in a variety of biomedical applications.

In this dissertation, we have proposed such a system that satisfies the requirements mentioned above based on supervised learning strategy. The system is general in the sense that, given sufficient training data, it can be used to segment a variety of biological structures, from different imaging modalities. First, a user provides the system with a few hand labeled/segmented images, which can be easily achieved without extensive customization in most cases, and then, the system proceeds by ”learning” a classification-based function that can be used to estimate the class of a given pixel (e.g. background vs. object). Once such a function is learned, it is then coupled with a region optimization algorithm (together with any geometric/shape bias terms available from the training data) for producing the segmentation of any unknown images. In summary, we have several innovations in this research work:

Our first innovation is to propose a general image segmentation framework based on supervised learning. We show that the feature vectors comprised from intensity neighborhoods, can be utilized as a simple and general feature, which provides decent segmentation performance across

different types of biomedical applications compared to the state-of-the-art methods specifically designed for these applications. However, we note that the intensity neighborhood features are quite sensitive to intensity variation, thus, data must be appropriately normalized for both training and testing data to guarantee the segmentation performance. In addition, the feature derived from intensity neighborhoods may fail due to the intensity heterogeneity problem (e.g. caused by uneven illumination condition) existing in the data, as we show some texture examples in Chapter 3. Thus, how to design a more robust feature that is insensitive to intensity heterogeneity problem is important and necessary, which will be one of our research directions in the future.

In addition, such a segmentation framework is designed to work across multiple scales. In order to prevent the disadvantage of using large neighborhoods that they amount to high dimensional spaces, making estimation and pixel classification difficult, we utilize small neighborhoods (e.g. N = 3 or 5) instead for maintaining reasonable computational costs. In order to capture the useful information of biological structures for different classes, we utilize multiple windows of the same size, but each window corresponds to a scale (obtained through different levels of blurring and sub-sampling). Therefore we can associate each pixel in the training set (as well as during actual classification of a test pixel) several sets of neighborhoods with the same size N but under different resolutions, comprising the multiple scales associated with that pixel. In this dissertation, weighted confidence voting is utilized to produce the final prediction, and how to better integrate classification information from each scale for prediction can be our next research direction in the future.

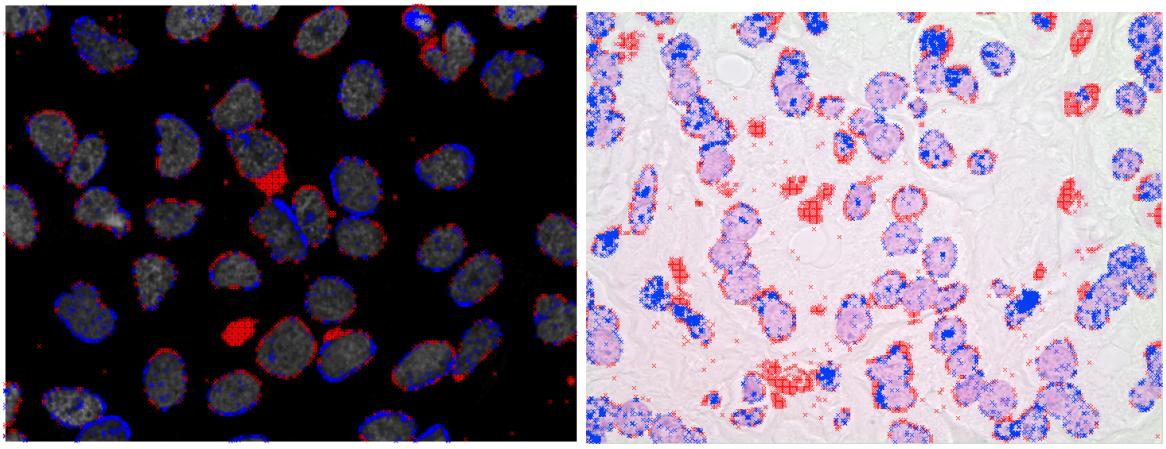
We compared the application of our system to several other segmentation approaches in three distinct biomedical image segmentation tasks: segmentation of tissues from 3D brain MR images, segmentation of tissues in color histology images, and segmentation of nuclei from gray-scale fluorescence microscopy images. We have chosen at least one other relatively modern segmentation method for comparison in each application. Overall, our general purpose segmentation system performed as well as (or at times better than) some of the best available custom tailored methods in each application, which validates the feasibility of this framework. We also note that the system consists of several modules, some of which can be replaced with advanced modules (e.g. classification module) later.

Our second innovation is to propose a fast classification algorithm based on data selection instead of SVM classifier previously used, where the basic idea is to select a small portion of subset from the original big training set, and use a very simple classifier (1-nearest neighbor classifier) to predict the label of each pixel. Since an obvious current drawback of our previous system is that it trades computation time for generality, the big challenge in developing a practical segmentation system is how to deal with big amount of training data so that both classification accuracy and computational speed can be guaranteed. We proposed a new majority-voting based data selection criterion, which works similarly to greedy algorithm in the way that at each iteration, one data point from the class with the lowest classification accuracy will be chosen and added to the current subset. We show that when the data is appropriately selected, even a weak classifier using a small portion of training data is able to achieve decent classification performance. We also show that the computational efficiency has been greatly improved at the same time. For example, in our previous implementation (all computation times reported were based on a single 2.0GHz Intel Xeon processor), the computing time for training the SVM classifier from 1.2×10^5 training samples for the histology image segmentation tests at one single scale

was 14.5 hours, and 6.4 hours for segmenting an image of size 1103×1421 pixels. In the new classification algorithm, for the same histology images of teratoma, it takes 2.11 hours to select a subset from 3.99×10^6 instances, and 0.41 hours to segment the same image.

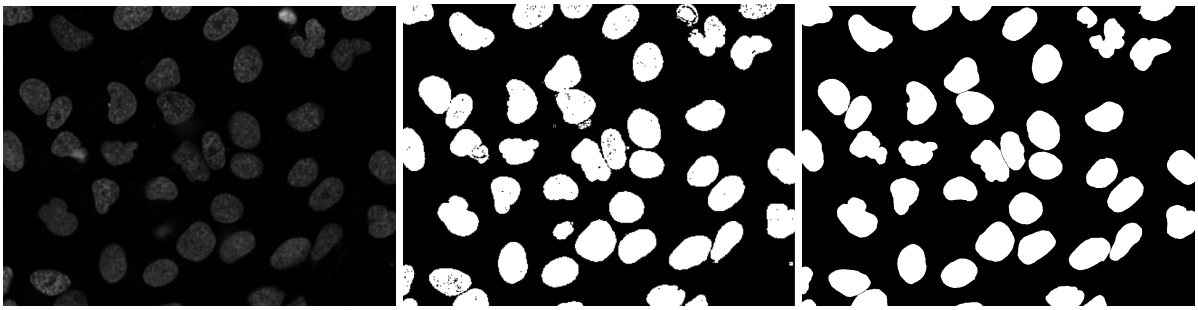
Meanwhile, we should note that this fast classification algorithm still has several limitations: (1) the proposed data selection criterion has been shown experimentally to outperform several other data selection criteria, however, we did not provide any theoretical proof to show why this criterion works, or how it can be improved. Thus, it is necessary to seek the theoretical support for the proposed data selection criterion, which helps the readers to better understand the essence of this criterion. (2) it is worth investigating what else information can be utilized to better select data instances. One hypothesis is to assume that pixels near the spatial border between different classes may contain more useful discriminant information for classification, and we provide two experimental examples to show that this hypothesis might be true. In the first experiment, we would like to show the spatial correspondences of support vector points that consist the optimal decision boundary. We test the hypothesis on the fluorescence microscopic images of nuclei and histological images of nuclei. Pixels from the two classes: nuclei and background, are randomly sampled to construct the training set, and the intensity neighborhood window feature is used as the feature vector for each pixel. Since the decision boundary with the maximal margin has been proved to be the optimal decision boundary [52], we will utilize the support vector machine (SVM) to find the optimal decision boundary by calculating the support vectors, which are the points near the decision boundary (optimal decision boundary) that maximize the margin between different classes. For validation, we plot these support vector points spatially on both fluorescence microscopic image and histological image in Figure 6.1 to show the relationship between the spatial position and the feature space position. In Figure 6.1, we use blue and red to represent the support vector points for nuclei class and background class respectively. Clearly, we can see that most support vector points exist near the spatial boundary between nuclei and background, which validates our hypothesis in some sense. In the second experiment, we would like to show that when we only choose the pixels near the spatial border between different classes for training, we can still achieve similar performance for classification. We choose the fluorescence nuclei images for testing, and the manually segmented result as the ground truth. In order to validate the hypothesis from an opposite direction, we only consider the pixels near the spatial borders between nuclei and background and again, randomly sample the pixels inside and outside the nuclei as from two classes, and implement the same pixel level classification procedure using SVM classifier. In Figure 6.2, we can see that although some noise exists, most nuclei can be segmented accurately. Again, the result proves that our hypothesis can be feasible. Thus, the correspondence between pixel space and feature space will be investigated as our future work. (3) Finally, it is necessary to investigate a faster implementation for distance calculation. Either in SVM or in 1-NN classifier (also the same for many other state-of-the-art classifiers), distances (e.g. L_2 distance) between feature vectors are to be calculated for classification purpose. In this research work, L_2 distance between feature vectors is calculated directly. However, we note that since each feature vector is constructed from a square window intensity neighborhood, if the L_2 distance between two feature vectors can be represented in the form of intensity neighborhoods, the L_2 distances between every feature vector in the training set with all feature vectors (all pixels) from the testing image can be implemented by convoluting the square window of that training feature vector with the whole testing image via fast convolution

trick, and thereby, the computational time can be significantly reduced.



(a) (b)

Figure 6.1: Support vectors plotted on the different modalities of images (a) fluorescence microscopic nuclei image (b) histological nuclei image



(a) (b) (c)

Figure 6.2: Segmentation result using the pixels near the spatial borders (a) original image (b) segmentation result only using the pixels near the borders for training (c) the ground truth

Our third innovation is that we proposed a deformable model for region optimization, which is able to delineate regions that encompass each single object in the image automatically, providing the segmented regions with smooth borders/surfaces and removing noise at the same time. Different from some state-of-the-art models [164], which require the probabilities of pixels belonging to different classes as the input, our model is more general in the sense that it takes the classification decision value/confidence directly as the input, and takes both region information and edge information into consideration for optimization. We note that the proposed model can be generally utilized as a standard post-processing step after pixel classification processes in different types of experiments to optimize the segmented regions. We also show experimentally that such a model is able to provide much better result rather than simply grouping the pixels with the same label as the segmented regions. In this Ph.D. research work, we provide the model

for the case of binary classes, and we will extend such a model to multiple classes case as our future work.

Our last innovation is to propose another general system based on supervised learning for segmenting biomedical objects with a strong shape prior information available, and we focus on segmentation of cell nuclei in this work. We note that this system is suitable for a variety of imaging experiments given that it contains a training step that adapts the statistical model for the given type of data, and a testing step that segments arbitrary images by finding the instance in the model that best matches, in the sense of the normalized cross correlation (NCC), local regions in the input images. Given an experimental setup, once the training operation is completed, the method is able to segment automatically any number of images from the same modality. To our knowledge, our system is the first method to apply a template matching approach which includes texture and shape variations to accurately delineating nuclei from microscopy images. In addition, to our knowledge our system is the first method to utilize a supervised learning strategy to build such a statistical model, that includes texture and shape variations in multiple channels, for detecting nuclei from microscopy images.

In a practical sense, the proposed system provides three main contributions. First, its overall performance is robust across different types of data with little tuning of parameters. We have demonstrated this here by applying the same exact software (with the only difference in each test being the value for μ) to a total of six different imaging modalities and showing the method performs as well or better than all other methods we were able to compare against. Secondly, amongst the methods we have tested in this manuscript, we show that our method is the only method (besides manual segmentation) that is capable of handling significant intensity heterogeneity. This is due to the fact that we utilize the NCC metric in the registration-based segmentation process, which is independent of the overall intensity of the local region of the image being segmented. Finally, we also mention that, amongst all methods tried, the template matching method we described produced noticeably more smooth and accurate borders with fewer spurious contours. The smoothness in the contours obtained by our method is primarily due to the fact that the statistical modeling we use includes only the main modes of variation in nuclear shape. These tend to be, topically, size, elongation, as well as bending (in addition to rotation). High order fluctuation in contours do occur in nuclei at times, but these do not occur as often as the ones already mentioned. We note that the method is still flexible enough to accurately segment nuclei that do not conform to these main modes of variation given the elastic matching procedure applied in the last step of the procedure.

Finally, it is necessary to discuss the computational cost of our template matching approach, which is also important in evaluating the performance of an algorithm. Our current system is implemented in MATLAB, which may take some time for segmenting nuclei from one image, for example, it takes about 20 minutes to detect and segment all cell nuclei (36 nuclei) from a fluorescence image (1030×1349), however, we note that the computational time can often be significantly reduced by implementing the algorithm in a compiled language such as C, for example. We also note that the computational time of our algorithm can be even decreased by utilizing a multi-scale framework. That is, instead of performing the filtering-based approach for detection in the original image space, we have also experimented with first reducing the size of the image (and templates) by two for the initial detection only (the remaining part of the method utilized the full resolution image). Thus we were able to reduce the total computation

time for the same field of view to roughly 10.6 minutes. The accuracy of the final segmentation was not severely affected (data not shown for brevity). Future work will include improving the computational efficiency of this method by further investigation of multi-scale approaches, as well as faster optimization methods (e.g. conjugate gradient). Finally, we note again that the approach described above utilizes all color information contained in the training and test image. In cases where the nuclear stain color is known precisely, the approach can be easily modified to utilize only that color. In addition, many existing techniques for optimal color transformation [215] can also be combined with our proposed approach in the future for better performance.

Appendix A

Here we describe the non-rigid registration algorithm used in both the training and segmentation steps outlined above. Let $T(\mathbf{x})$ represent a target image (usually a raw image to be segmented) to which a source image (in our case the template) $S(\mathbf{x})$ is to be deformed such that $T(\mathbf{x}) \approx S_u(\mathbf{x}) = S(\mathbf{x} - u(\mathbf{x}))$, with $u(\mathbf{x})$ representing the warping function to be computed. We wish to maximize the square of the multichannel normalized cross correlation (NCC) between the two images:

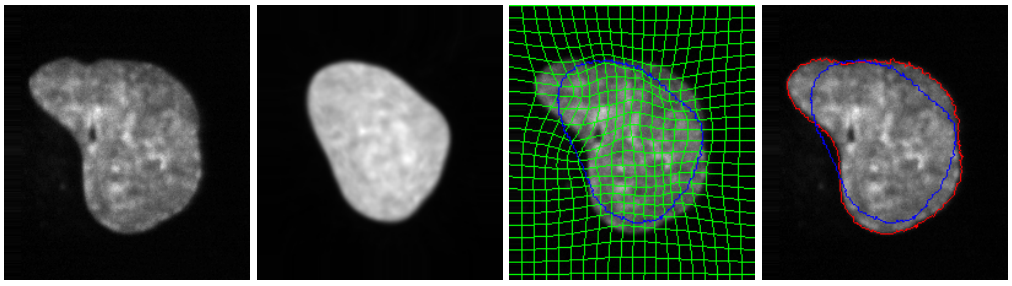
$$\Phi(u) = \frac{1}{N_{ch}} \sum_{j=1}^{N_{ch}} \left(\frac{\langle S_u^j, T^j \rangle}{\|S_u^j\| \|T^j\|} \right)^2 \tag{A.1}$$

where N_{ch} is the number of channels, and $\|T\|^2 = \langle T, T \rangle = \sum_{\mathbf{x}} T(\mathbf{x})T(\mathbf{x})$, where the sum is computed over the (fixed) image grid. We note that maximizing the squared NCC is equivalent to maximizing the NCC. We choose the squared NCC since it provides a more general framework, in which both positive and negative cross correlations can be optimized to the same effect. Equation (A.1) is maximized via steepest gradient ascent. The gradient of it is given by:

$$\begin{aligned} \nabla_u \Phi(u^k; \mathbf{x}) = \frac{2}{N_{ch}} \sum_{j=1}^{N_{ch}} \frac{\langle S_u^j, T^j \rangle}{\|S_u^j\|^4 \|T^j\|^2} \times \\ \left(\left\langle S_u^j, T^j \right\rangle S_u^j(\mathbf{x}) - \|S_u^j\|^2 T(\mathbf{x}) \right) \nabla S_u^j(\mathbf{x}) \end{aligned} \tag{A.2}$$

In practice we convolve $\nabla_u \Phi(u, \mathbf{x})$ with a radially symmetric Gaussian kernel of variance σ^2 in order to regularize the problem. Optimization is conducted iteratively starting with $u^0(\mathbf{x}) = 0$, and $u^{k+1}(\mathbf{x}) = u^k + \kappa G_\sigma(\mathbf{x}) * \nabla_u \Phi(u^k; \mathbf{x})$, where G_σ is the Gaussian kernel, $*$ represents the digital convolution operation, and κ is a small step size. Optimization continues until the increase in the NCC value falls below a chosen threshold.

In addition, we perform the maximization above in a multi scale framework. That is, we utilize a sequence of images $[T(\mathbf{x})]_2, [S(\mathbf{x})]_2$, $[T(\mathbf{x})]_1, [S(\mathbf{x})]_1$, and $[T(\mathbf{x})]_0, [S(\mathbf{x})]_0$, where $[T(\mathbf{x})]_2$ denotes the image T donwsampled by four (reduced to 1/8 of its size) after blurring, $[T(\mathbf{x})]_1$ denotes the image T donwsampled by two (reduced to 1/4 of its size) after blurring, and $[T(\mathbf{x})]_0$ denotes the original image being matched. The algorithm starts by obtaining an estimate for $u(\mathbf{x})$ (using the gradient ascent algorithm described above) using images $[T(\mathbf{x})]_2, [S(\mathbf{x})]_2$. The estimate of the deformation map u is then used to initialized the same gradient ascent algorithm using images $[T(\mathbf{x})]_2, [S(\mathbf{x})]_2$, and so on.



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

Figure A.1: Accurate segmentation process. (a) real image patch; (b) simulated image patch; (c) the warping mesh; (d) the final accurate segmentation result.

In Figure A.1, we show an example of non-rigid registration between a real nucleus and a template, in which Figure A.1(a) shows the original image of a nucleus, while Figure A.1(b) shows a nucleus template that best matches the real nucleus. Non-rigid registration process is implemented, which deforms the template to the real nucleus in Figure A.1(c), where the wrapping mesh is shown in green, and blue contours represent the border of best-matching template. After deformation, we shown the segmentation result in Figure A.1(d), in which red contours represent the border of the template after deformation. Apparently, we can see that the border of the template after deformation delineates the border of real nucleus very accurately.

Bibliography

[1] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285-296, pp. 23–27, 1975. 1.1, 2.3.4, 5.1

[2] DY. Huang and CH Wang, "Optimal multi-level thresholding using a two-stage otsu optimization approach," *Pattern Recognition Letters*, vol. 30, no. 3, pp. 275–284, 2009. 1.1

[3] SK. S. Fan and Y. Lin, "A multi-level thresholding approach using a hybrid optimal estimation algorithm," *Pattern recognition letters*, vol. 28, no. 5, pp. 662–669, 2007. 1.1

[4] Y.W. Lim and S.Uk. Lee, "On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques," *Pattern Recognition*, vol. 23, no. 9, pp. 935–952, 1990. 1.1

[5] A.KC. Wong and PK. Sahoo, "A gray-level threshold selection method based on maximum entropy principle," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 19, no. 4, pp. 866–871, 1989. 1.1

[6] T.N. Pappas, "An adaptive clustering algorithm for image segmentation," *Signal Processing, IEEE Transactions on*, vol. 40, no. 4, pp. 901–914, 1992. 1.1

[7] WN. Lie, "Automatic target segmentation by locally adaptive image thresholding," *Image Processing, IEEE Transactions on*, vol. 4, no. 7, pp. 1036–1041, 1995. 1.1

[8] WK Pratt, *Digital Image Processing*, John Wiley & Sons Inc., 1991. 1.1, 5.3.2

[9] R.C. Gonzalez and R.E. Woods, *Digital image processing*, Prentice Hall Upper Saddle River, NJ, second edition, 2002. 1.1, 5.1

[10] N.J Nilsson, "Principles of artificial intelligence," *Symbolic Computation, Berlin: Springer, 1982*, vol. 1, 1982. 1.1

[11] P.H. Winston, "Artificial intelligence," *Reading: Addison-Wesley*, 1984. 1.1

[12] R.O. Duda and P.E Hart, "Use of the hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972. 1.1

[13] A.P. Dhawan and A. Sim, "Segmentation of images of skin lesions using color and texture information of surface pigmentation," *Computerized Medical Imaging and Graphics*, vol. 16, no. 3, pp. 163–177, 1992. 1.1

[14] A.P. Dhawan and L. Arata, "Segmentation of medical images through competitive learning," *Computer Methods and Programs in Biomedicine*, vol. 40, no. 3, pp. 203–215, 1993. 1.1

[15] A.M Nazif and M.D Levine, "Low level image segmentation: an expert system," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, , no. 5, pp. 555–577, 1984. 1.1

[16] G. Lin, U. Adiga, K. Olson, J.F. Guzowski, C.A. Barnes, and B. Roysam, "A hybrid 3d watershed algorithm incorporating gradient cues and object models for automatic segmentation of nuclei in confocal image stacks," *Cytometry*, vol. 56, no. 1, pp. 23–36, 2003. 1.1, 2.3.4, 5.1, 5.3.4, 5.2

[17] X. Yang, H. Li, and X. Zhou, "Nuclei segmentation using marker-controlled watershed, tracking using mean-shift, and kalman filter in time-lapse microscopy," *Circuits System I, IEEE Transactions on*, vol. 53, no. 11, pp. 2405–2414, 2006. 1.1, 1.2.1, 5.1

[18] A.E Carpenter, T.R Jones, M.R Lamprecht, C. Clarke, I.H Kang, O. Friman, D. A Guertin, J.H Chang, R.A Lindquist, J. Moffat, et al., "Cellprofiler: image analysis software for identifying and quantifying cell phenotypes," *Genome biology*, vol. 7, no. 10, pp. R100, 2006. 1.1, 5.3.2

[19] L. Florack and A. Kuijper, "The topological structure of scale-space images," *Journal of Mathematical Imaging and Vision*, vol. 12, no. 1, pp. 65–79, 2000. 1.1

[20] J. M. Gauch and S. M. Pizer, "Multiresolution analysis of ridges and valleys in grey-scale images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 15, no. 6, pp. 635–646, 1993. 1.1

[21] O. Olsen and M. Nielsen, "Multi-scale gradient magnitude watershed segmentation," in *Image Analysis and Processing*. Springer, 1997, pp. 6–13. 1.1

[22] T. Lindeberg, "Detecting salient blob-like image structures and their scales with a scale-space primal sketch: a method for focus-of-attention," *International Journal of Computer Vision*, vol. 11, no. 3, pp. 283–318, 1993. 1.1

[23] T. Lindeberg, *Scale-space theory in computer vision*, Springer, 1993. 1.1

[24] E. Akbas and N. Ahuja, "From ramp discontinuities to segmentation tree," *Computer Vision–ACCV 2009*, pp. 123–134, 2010. 1.1

[25] C. Undeman and T. Lindeberg, "Fully automatic segmentation of mri brain images using probabilistic anisotropic diffusion and multi-scale watersheds," in *Scale Space Methods in Computer Vision*. Springer, 2003, pp. 641–656. 1.1

[26] T. McInerney and D. Terzopoulos, "Deformable models in medical image analysis," in *Mathematical Methods in Biomedical Image Analysis, 1996., Proceedings of the Workshop on*. IEEE, 1996, pp. 171–180. 1.1

[27] CY Xu, D.L. Pham, and J.L Prince, "Image segmentation using deformable models," *Handbook of medical imaging*, vol. 2, pp. 129–174, 2000. 1.1

[28] L.A. Vese and T.F. Chan, "A multiphase level set framework for image segmentation using the mumford and shah model," *International Journal of Computer Vision*, vol. 50, no. 3, pp. 271–293, 2002. 1.1

[29] C. Xu and J.L. Prince, "Gradient vector flow: A new external force for snakes," in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*. IEEE, 1997, pp. 66–71. 1.1

[30] C. Zimmer, E. Labruyere, V. Meas-Yedid, N. Guillen, and J-C Olivo-Marin, "Segmentation and tracking of migrating cells in videomicroscopy with parametric active contours: A tool for cell-based drug testing," *Medical Imaging, IEEE Transactions on*, vol. 21, no. 10, pp. 1212–1221, 2002. 1.1

[31] N. Ray, S.T. Acton, T. Altes, E.E De Lange, and J.R. Brookeman, "Merging parametric active contours within homogeneous image regions for mri-based lung segmentation," *Medical Imaging, IEEE Transactions on*, vol. 22, no. 2, pp. 189–199, 2003. 1.1

[32] N. Paragios and R. Deriche, "Coupled geodesic active regions for image segmentation: A level set approach," *Computer VisionECCV 2000*, pp. 224–240, 2000. 1.1

[33] J. Lie, M. Lysaker, and XC Tai, "A variant of the level set method and applications to image segmentation," *Mathematics of computation*, vol. 75, no. 255, pp. 1155–1174, 2006. 1.1

[34] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International journal of computer vision*, vol. 1, no. 4, pp. 321–331, 1988. 1.1, 1.2.1, 2.1, 4.1

[35] A. Yezzi Jr, S. Kichenassamy, A. Kumar, P. Olver, and A. Tannenbaum, "A geometric snake model for segmentation of medical imagery," *Medical Imaging, IEEE Transactions on*, vol. 16, no. 2, pp. 199–209, 1997. 1.1

[36] A. Hafiane, F. Bunyak, and K. Palaniappan, "Evaluation of level set-based histology image segmentation using geometric region criteria," in *Biomedical Imaging: From Nano to Macro, 2009. ISBI'09. IEEE International Symposium on*. IEEE, 2009, pp. 1–4. 1.1, 1.2.1

[37] C. Li, C. Xu, C. Gui, and M.D. Fox, "Level set evolution without re-initialization: A new variational formulation," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. IEEE, 2005, vol. 1, pp. 430–436. 1.1, 4.1, 4.3, 5.2.1

[38] T.F. Chan and L.A. Vese, "Active contours without edges," *Image Processing, IEEE Transactions on*, vol. 10, no. 2, pp. 266–277, 2001. 1.1, 1.2.1, 4.1, 5.1, 5.3.2, 5.1, 5.2

[39] J. Shi and J. Malik, "Normalized cuts and image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 888–905, 2000. 1.1

[40] L. Grady, "Random walks for image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 11, pp. 1768–1783, 2006. 1.1

[41] ZY Wu and R. Leahy, "An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 15, no. 11, pp. 1101–1113, 1993. 1.1

[42] L. Grady and E.L Schwartz, "Isoperimetric graph partitioning for image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 3, pp. 469–475, 2006. 1.1

[43] Y. Boykov and M.P. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in nd images," in *International Conference on Computer Vision*. Citeseer, 2001, vol. 1, pp. 105–112. 1.1, 2.1

[44] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient nd image segmentation," *International Journal of Computer Vision*, vol. 70, no. 2, pp. 109–131, 2006. 1.1, 1.2.1, 4.1

[45] C. Chen, H. Li, X. Zhou, and STC Wong, "Constraint factor graph cut–based active contour method for automated cellular image segmentation in rnai screening," *Journal of Microscopy*, vol. 230, no. 2, pp. 177–191, 2008. 1.1, 5.1

[46] C. Chen, W. Wang, J.A Ozolek, N. Lages, S.J Altschuler, L.F Wu, and G.K Rohde, "A template matching approach for segmenting microscopy images," in *Biomedical Imaging (ISBI), 2012 9th IEEE International Symposium on*. IEEE, 2012, pp. 768–771. 1.1, 1.4, 5, 5.3.3

[47] M. Prastawa, E. Bullitt, N. Moon, K. Van Leemput, and G. Gerig, "Automatic brain tumor segmentation by subject specific modification of atlas priors," *Academic Radiology*, vol. 10, no. 12, pp. 1341–1348, 2003. 1.1

[48] T. Heimann and HP Meinzer, "Statistical shape models for 3d medical image segmentation: a review.," *Medical image analysis*, vol. 13, no. 4, pp. 543, 2009. 1.1

[49] B. Van Ginneken, A. F Frangi, J. J Staal, B. M ter Haar Romeny, and M. A. Viergever, "Active shape model segmentation with optimal features," *Medical Imaging, IEEE Transactions on*, vol. 21, no. 8, pp. 924–933, 2002. 1.1

[50] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 6, pp. 681–685, 2001. 1.1

[51] S. C. Mitchell, J. G. Bosch, B. P. F. Lelieveldt, R. J van der Geest, J. H. C. Reiber, and M. Sonka, "3-d active appearance models: segmentation of cardiac mr and ultrasound images," *Medical Imaging, IEEE Transactions on*, vol. 21, no. 9, pp. 1167–1178, 2002. 1.1

[52] C.M. Bishop, *Pattern recognition and machine learning*, vol. 4, Springer New York, 2006. 1.1, 2.2, 2.2.3, 2.2.5, 2.2.6, 4.2, 6

[53] G. B Coleman and H. C Andrews, "Image segmentation by clustering," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 773–785, 1979. 1.1

[54] J. Winn and N. Jojic, "Locus: Learning object classes with unsupervised segmentation," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*. IEEE, 2005, vol. 1, pp. 756–763. 1.1

[55] H. Zhang, J.E. Fritts, and S.A. Goldman, "Image segmentation evaluation: a survey of unsupervised methods," *Computer Vision and Image Understanding*, vol. 110, no. 2, pp. 260–280, 2008. 1.1

[56] KZ Mao, Peng Zhao, and Puay-Hoon Tan, "Supervised learning-based cell image segmentation for p53 immunohistochemistry," *Biomedical Engineering, IEEE Transactions on*, vol. 53, no. 6, pp. 1153–1163, 2006. 1.1, 5.1

[57] C. Chen, J.A Ozolek, W. Wang, and G.K Rohde, "A general system for automatic biomedical image segmentation using intensity neighborhoods," *International Journal of Biomed-*

ical Imaging, vol. 2011, pp. 8, 2011. 1.1, 1.2.1, 1.4, 2.1, 3, 3.3, 3.3.3, 3.3.3, 3.3.4

[58] S. Ray and R.H. Turi, "Determination of number of clusters in k-means clustering and application in colour image segmentation," in *Proceedings of the 4th international conference on advances in pattern recognition and digital techniques*, 1999, pp. 137–143. 1.1

[59] HP Ng, SH Ong, KWC Foong, PS Goh, and WL Nowinski, "Medical image segmentation using k-means clustering and improved watershed algorithm," in *Image Analysis and Interpretation, 2006 IEEE Southwest Symposium on*. IEEE, 2006, pp. 61–65. 1.1

[60] KS. Chuang, HL. Tzeng, S. Chen, J. Wu, and TJ. Chen, "Fuzzy c-means clustering with spatial information for image segmentation," *computerized medical imaging and graphics*, vol. 30, no. 1, pp. 9–15, 2006. 1.1

[61] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: Image segmentation using expectation-maximization and its application to image querying," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 8, pp. 1026–1038, 2002. 1.1

[62] C. Chen, J.A. Ozolek, W. Wang, and G.K Rohde, "A pixel classification system for segmenting biomedical images using intensity neighborhoods and dimension reduction," in *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*. IEEE, 2011, pp. 1649–1652. 1.1, 1.4, 2.1

[63] KS. Cheng, JS. Lin, and CW. Mao, "The application of competitive hopfield neural network to medical image segmentation," *Medical Imaging, IEEE Transactions on*, vol. 15, no. 4, pp. 560–567, 1996. 1.1

[64] S. Li, T. Fevens, A. Krzyżak, and S. Li, "Automatic clinical image segmentation using pathological modeling, pca and svm," *Engineering Applications of Artificial Intelligence*, vol. 19, no. 4, pp. 403–410, 2006. 1.1

[65] S. Belongie, C. Carson, H. Greenspan, and J. Malik, "Color-and texture-based image segmentation using em and its application to content-based image retrieval," in *Computer Vision, 1998. Sixth International Conference on*. IEEE, 1998, pp. 675–682. 1.2

[66] I. Bankman, *Handbook of medical imaging: processing and analysis management*, Academic Press, 2000. 1.2

[67] M. Brejl and M. Sonka, "Object localization and border detection criteria design in edge-based image segmentation: automated learning from examples," *Medical Imaging, IEEE Transactions on*, vol. 19, no. 10, pp. 973–985, 2000. 1.2

[68] J. Im, JR Jensen, and JA Tullis, "Object-based change detection using correlation image analysis and image segmentation," *International Journal of Remote Sensing*, vol. 29, no. 2, pp. 399–423, 2008. 1.2

[69] L. Vese, "Multiphase object detection and image segmentation," *Geometric Level Set Methods in Imaging, Vision, and Graphics*, pp. 175–194, 2003. 1.2

[70] Y. Zhang, M. Brady, and S. Smith, "Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm," *Medical Imaging, IEEE Transactions on*, vol. 20, no. 1, pp. 45, 2001. 1.2.1, 2.1

[71] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International journal of computer vision*, vol. 22, no. 1, pp. 61–79, 1997. 1.2.1, 4.1

[72] C. Li, C. Xu, C. Gui, and M.D. Fox, "Level set evolution without re-initialization: A new variational formulation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Washington, DC, USA, 2005, vol. 1, pp. 430–436, IEEE Computer Society. 1.2.1

[73] C. Li, C.Y. Kao, J.C. Gore, and Z. Ding, "Implicit active contours driven by local binary fitting energy," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–7. 1.2.1

[74] WM Wells III, WEL Grimson, R. Kikinis, and FA Jolesz, "Adaptive segmentation of MRI data," *Medical Imaging, IEEE Transactions on*, vol. 15, no. 4, pp. 429–442, 1996. 1.2.1

[75] K. Van Leemput, F. Maes, D. Vandermeulen, and P. Suetens, "Automated model-based bias field correction of mr images of the brain," *Medical Imaging, IEEE Transactions on*, vol. 18, no. 10, pp. 885–896, 1999. 1.2.1, 2.2.1, 2.3.1, 2.3.2

[76] S.P. Awate, T. Tasdizen, N. Foster, and R.T. Whitaker, "Adaptive, nonparametric markov modeling for unsupervised, mri brain-tissue classification," *Medical Image Analysis*, vol. 10, no. 5, pp. 726–739, 2006. 1.2.1, 2.2.2, 2.3.1, 2.3.2, 2.2, 2.3.5

[77] C.A. Davatzikos and J.L. Prince, "An active contour model for mapping the cortex," *Medical Imaging, IEEE Transactions on*, vol. 14, no. 1, pp. 65–80, 1995. 1.2.1

[78] R. Valdés-Cristerna, V. Medina-Bañuelos, and O. Yáñez-Suárez, "Coupling of radial-basis network and active contour model for multispectral brain MRI segmentation," *Biomedical Engineering, IEEE Transactions on*, vol. 51, no. 3, pp. 459–470, 2004. 1.2.1

[79] Y. Artan, D.L. Langer, M.A. Haider, T.H. van der Kwast, A.J. Evans, M.N. Wernick, and I.S. Yetik, "Prostate cancer segmentation with multispectral mri using cost-sensitive conditional random fields," in *Biomedical Imaging: From Nano to Macro, 2009. ISBI'09. IEEE International Symposium on*. IEEE, 2009, pp. 278–281. 1.2.1

[80] R. Toth and A. Madabhushi, "Multifeature landmark-free active appearance models: Application to prostate mri segmentation," *Medical Imaging, IEEE Transactions on*, vol. 31, no. 8, pp. 1638–1650, 2012. 1.2.1

[81] L. Zhukov, ZS Bao, I. Guskov, J. Wood, and D. Breen, "Dynamic deformable models for 3d mri heart segmentation," in *SPIE Medical Imaging*, 2002, vol. 4684, pp. 1398–1405. 1.2.1

[82] D. Metaxas, T. Chen, X. Huang, and L. Axel, "Cardiac segmentation from mri-tagged and ct images," in *8th WSEAS International Conf. on Computers, special session on Imaging and Image Processing of Dynamic Processes in biology and medicine*, 2004. 1.2.1

[83] O. Kubassova, "Automatic segmentation of blood vessels from dynamic mri datasets," *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2007*, pp. 593–600, 2007. 1.2.1

[84] C.A. Castro, A. Ben-Yehudah, J.A. Ozolek, P.H. Mills, C.J. Redinger, J.D. Mich-Basso, D.A. McFarland, S.L. Oliver, E.T. Ahrens, and G. Schatten, "Semiquantitative

histopathology and 3d magnetic resonance microscopy as collaborative platforms for tissue identification and comparison within teratomas derived from pedigreed primate embryonic stem cells," *Stem cell research*, vol. 5, no. 3, pp. 201–211, 2010. 1.2.1, 2.3.3, 3.3, 3.3.3

[85] C. W. Pouton and J. M Haynes, "Embryonic stem cells as a source of models for drug discovery," *Nature Reviews Drug Discovery*, vol. 6, no. 8, pp. 605–616, 2007. 1.2.1, 2.3.3

[86] H. Thomson, "Bioprocessing of embryonic stem cells for drug discovery," *Trends in biotechnology*, vol. 25, no. 5, pp. 224–230, 2007. 1.2.1, 2.3.3

[87] S. Naik, S. Doyle, M. Feldman, J. Tomaszewski, and A. Madabhushi, "Gland segmentation and computerized gleason grading of prostate histology by integrating low-, high-level and domain specific information," in *MIAAB Workshop*. Citeseer, 2007. 1.2.1

[88] P.S Karvelis, D.I Fotiadis, I. Georgiou, and M. Syrrou, "A watershed based segmentation method for multispectral chromosome images classification," in *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE*. IEEE, 2006, pp. 3009–3012. 1.2.1

[89] S. Petushi, F.U Garcia, M.M Haber, C. Katsinis, and A. Tozeren, "Large-scale computations on histology images reveal grade-differentiating parameters for breast cancer," *BMC Medical Imaging*, vol. 6, no. 1, pp. 14, 2006. 1.2.1

[90] O. Sertel, J. Kong, U.V Catalyurek, G. Lozanski, J.H Saltz, and M.N Gurcan, "Histopathological image analysis using model-based intermediate representations and color texture: Follicular lymphoma grading," *Journal of Signal Processing Systems*, vol. 55, no. 1, pp. 169–183, 2009. 1.2.1

[91] J. Kong, O. Sertel, H. Shimada, KL. Boyer, JH. Saltz, and MN. Gurcan, "Computer-aided evaluation of neuroblastoma on whole-slide histology images: Classifying grade of neuroblastic differentiation," *Pattern Recognition*, vol. 42, no. 6, pp. 1080–1092, 2009. 1.2.1

[92] O. Sertel, J. Kong, H. Shimada, UV. Catalyurek, J.H Saltz, and M.N Gurcan, "Computer-aided prognosis of neuroblastoma on whole-slide images: Classification of stromal development," *Pattern Recognition*, vol. 42, no. 6, pp. 1093–1103, 2009. 1.2.1

[93] R. Bhagavatula, M. Fickus, W. Kelly, C. Guo, J.A. Ozolek, C.A. Castro, and J. Kovačević, "Automatic identification and delineation of germ layer components in h&e stained images of teratomas derived from human and nonhuman primate embryonic stem cells," in *Proceedings of the 2010 IEEE international conference on Biomedical imaging: from nano to Macro*. IEEE Press, 2010, pp. 1041–1044. 1.2.1, 2.1, 2.3.3, 2.4

[94] M. N Gurcan, L. E Boucheron, A. Can, A. Madabhushi, N. M Rajpoot, and B. Yener, "Histopathological image analysis: A review," *Biomedical Engineering, IEEE Reviews in*, vol. 2, pp. 147–171, 2009. 1.2.1

[95] G. Xiong, X. Zhou, and L. Ji, "Automated segmentation of drosophila rnai fluorescence cellular images using deformable models," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 53, no. 11, pp. 2415–2424, 2006. 1.2.1, 2.2.1

[96] P. Yan, X. Zhou, M. Shah, and S.T.C. Wong, "Automatic segmentation of high-throughput rnai fluorescent cellular images," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 12, no. 1, pp. 109–117, 2008. 1.2.1, 5.1

[97] C. Chen, W. Wang, J.A Ozolek, and G.K Rohde, "A flexible and robust approach for segmenting cell nuclei from 2d microscopy images using supervised learning and template matching," *Cytometry Part. A*, 2013. 1.4, 5

[98] T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, and A.Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, 2002. 2.1, 2.2.3

[99] J.A. Sethian et al., "Level set methods and fast marching methods," *Journal of Computing and Information Technology*, vol. 11, no. 1, pp. 1–2, 2003. 2.1, 4.1

[100] N. Malpica, C. Ortiz de Solorzano, J.J. Vaquero, A. Santos, I. Vallcorba, J.M. Garcia-Sagredo, and F. del Pozo, "Applying watershed algorithms to the segmentation of clustered nuclei," *Cytometry*, vol. 28, no. 4, pp. 289–297, 1997. 2.1, 2.3.4, 5.1

[101] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, pp. 583–598, 1991. 2.1, 2.3.4

[102] O. Debeir, C. Decaestecker, J.L. Pasteels, I. Salmon, R. Kiss, and P. Van Ham, "Computer-assisted analysis of epiluminescence microscopy images of pigmented skin lesions," *Cytometry Part A*, vol. 37, no. 4, pp. 255–266, 1999. 2.1

[103] A. Madabhushi, M.D. Feldman, D.N. Metaxas, J. Tomaszeweski, and D. Chute, "Automated detection of prostatic adenocarcinoma from high-resolution ex vivo mri," *Medical Imaging, IEEE Transactions on*, vol. 24, no. 12, pp. 1611, 2005. 2.1, 2.3.3, 2.4

[104] E.B. Dam and M. Loog, "Efficient segmentation by sparse pixel classification," *Medical Imaging, IEEE Transactions on*, vol. 27, no. 10, pp. 1525–1534, 2008. 2.1

[105] R.D. Nowak, "Wavelet-based rician noise removal for magnetic resonance imaging," *IEEE Transactions on Image Processing*, vol. 8, no. 10, 1999. 2.2.1

[106] M. Varma and A. Zisserman, "Texture classification: Are filter banks necessary?," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003*, 2003, vol. 2. 2.2.2

[107] S. Awate, T. Tasdizen, and R. Whitaker, "Unsupervised texture segmentation with nonparametric neighborhood statistics," *Computer Vision–ECCV 2006*, pp. 494–507, 2006. 2.2.2

[108] A. Buades, B. Coll, and J.M. Morel, "A non-local algorithm for image denoising," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2005*, 2005, pp. 60–65. 2.2.2

[109] T. Tasdizen, "Principal neighborhood dictionaries for nonlocal means image denoising," *IEEE Transactions on Image Processing*, vol. 18, no. 12, pp. 2649–2660, 2009. 2.2.2

[110] T. Lindeberg, "Scale-space for discrete signals," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 3, pp. 234–254, 1990. 2.2.2

[111] T. Hastie, R. Tibshirani, and J.H. Friedman, *The elements of statistical learning: data mining, inference, and prediction: with 200 full-color illustrations*, Springer New York:, 2001. 2.2.4

[112] C.C. Chang and C.J. Lin, "Libsvm: a library for support vector machines," 2001. 2.2.4

[113] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995. 2.2.5, 2.2.5

[114] B.E. Boser, I.M. Guyon, and V.N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*. ACM, 1992, pp. 144–152. 2.2.5

[115] S. Knerr, L. Personnaz, G. Dreyfus, and et al., "Single-layer learning revisited: A stepwise procedure for building and training a neural network," *Optimization Methods and Software*, vol. 1, pp. 23–34, 1990. 2.2.5

[116] C. Chih-Chung and L. Chih-Jen, "LIBSVM: a library for support vector machines," *Software available at http://www. csie. ntu. edu. tw/˜ cjlin/libsvm*, 2001. 2.2.5

[117] T. Hastie, R. Tibshirani, and J.H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, Springer Verlag, 2009. 2.2.5

[118] D. Haussler, M. Kearns, and R.E. Schapire, "Bounds on the sample complexity of bayesian learning using information theory and the vc dimension," *Machine Learning*, vol. 14, no. 1, pp. 83–113, 1994. 2.2.6

[119] P. Domingos, "Bayesian averaging of classifiers and the overfitting problem," in *International Conference on Machine Learning*. Citeseer, 2000, pp. 223–230. 2.2.6

[120] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996. 2.2.6

[121] Y. Freund and R. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *Computational learning theory*. Springer, 1995, pp. 23–37. 2.2.6

[122] T. Dietterich, "Ensemble methods in machine learning," *Multiple classifier systems*, pp. 1–15, 2000. 2.2.6

[123] A. Madabhushi, J. Shi, M. Feldman, M. Rosen, and J. Tomasezweski, "Comparing classification performance of feature ensembles: Detecting prostate cancer from high resolution mri," *Computer Vision Methods in Medical Image Analysis (In conjunction with ECCV)*, pp. 25–36, 2006. 2.2.6

[124] A. Madabhushi, J. Shi, M. Feldman, M. Rosen, and J. Tomaszewski, "Comparing ensembles of learners: Detecting prostate cancer from high resolution mri," *Computer Vision Approaches to Medical Image Analysis*, pp. 25–36, 2006. 2.2.6

[125] T.F. Wu, C.J. Lin, and R.C. Weng, "Probability estimates for multi-class classification by pairwise coupling," *The Journal of Machine Learning Research*, vol. 5, pp. 975–1005, 2004. 2.2.6

[126] A.J. Worth, "The Internet brain segmentation repository (IBSR)," *http://www. cma. mgh.*

Harvard. edu/ibsr. 2.3.1, 2.3.2

[127] L.R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, 1945. 2.3.1, 5.3.2

[128] T. Kapur, W.E.L. Grimson, W.M. Wells III, and R. Kikinis, *Segmentation of brain tissue from magnetic resonance images*, Citeseer, 1995. 2.3.1, 2.3.2

[129] K. Held, E.R. Kops, B.J. Krause, W.M. Wells III, R. Kikinis, and H.W. Muller-Gartner, "Markov random field segmentation of brain mr images," *Medical Imaging, IEEE Transactions on*, vol. 16, no. 6, pp. 878–886, 2002. 2.3.1, 2.3.2

[130] R.P. Lanza, *Essentials of stem cell biology*, Academic Press, 2006. 2.3.3

[131] MATLAB Image Processing ToolBox, "Color-based segmentation using k-means clustering," *http://www.mathworks.com/products/demos/image/color_seg_k/ipexhistology.html*. 2.3.3

[132] A.Z. Chitade and S.K. Katiyar, "Colour based image segmentation using k-means clustering," *International Journal of Engineering Science and Technology*, , no. 10, pp. 5319–5325. 2.3.3

[133] KS Ravichandran and B. Ananthi, "Color skin segmentation using k-means cluster," *Int. J. Comput. Appl. Math.*, vol. 4, no. 2, pp. 153–157, 2009. 2.3.3, 5.3.2, 5.1, 5.2

[134] S. Naik, S. Doyle, S. Agner, A. Madabhushi, M. Feldman, and J. Tomaszewski, "Automated gland and nuclei segmentation for grading of prostate and breast cancer histopathology," in *Proceedings of the 2008 IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI)*. IEEE, 2008, pp. 284–287. 2.3.3

[135] L.P. Coelho, A. Shariff, and R.F. Murphy, "Nuclear segmentation in microscope cell images: a hand-segmented dataset and comparison of algorithms," in *IEEE Int. Sym. Biomed. Imag., 2009.*, 2009, pp. 518–521. 2.3.4, 2.3.4, 2.3.5, 5.3.1, 5.3.4

[136] T.W. Ridler and S. Calvard, "Picture thresholding using an iterative selection method," *System, Man, Cybernetics, IEEE Transactions on*, vol. 8, no. 8, pp. 630–632, 1978. 2.3.4, 5.1

[137] G. Srinivasa, M. Fickus, M.N. Gonzalez-Rivero, S.Y. Hsieh, Y. Guo, A.D. Linstedt, and J. Kovacevic, "Active mask segmentation for the cell-volume computation and Golgi-body segmentation of HeLa cell images," in *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*. IEEE, 2008, pp. 348–351. 2.3.4

[138] D.P. Panda and A. Rosenfeld, "Image segmentation by pixel classification in (gray level, edge value) space," *Computers, IEEE Transactions on*, vol. 100, no. 9, pp. 875–879, 1978. 3

[139] M. Topi, P. Matti, and O. Timo, "Texture classification by multi-predicate local binary pattern operators," in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*. IEEE, 2000, vol. 3, pp. 939–942. 3, 3.3, 3.3.2, 3.1

[140] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Discriminative learned dictionaries for local image analysis," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8. 3, 3.3.2, 3.1

[141] X.Y. Wang, T. Wang, and J. Bu, "Color image segmentation using pixel wise support vector machine classification," *Pattern Recognition*, vol. 44, no. 4, pp. 777–787, 2011. 3

[142] X.Y. Wang, X.J. Zhang, H.Y. Yang, and J. Bu, "A pixel-based color image segmentation using support vector machine and fuzzy c-means," *Neural Networks*, 2012. 3

[143] J.A. Olvera-López, J.A. Carrasco-Ochoa, J.F. Martínez-Trinidad, and J. Kittler, "A review of instance selection methods," *Artificial Intelligence Review*, vol. 34, no. 2, pp. 133–143, 2010. 3.1, 3.1

[144] D.R. Wilson and T.R. Martinez, "Reduction techniques for instance-based learning algorithms," *Machine learning*, vol. 38, no. 3, pp. 257–286, 2000. 3.1

[145] H. Brighton and C. Mellish, "Advances in instance selection for instance-based learning algorithms," *Data mining and knowledge discovery*, vol. 6, no. 2, pp. 153–172, 2002. 3.1

[146] S. Garcia, J. Derrac, J.R. Cano, and F. Herrera, "Prototype selection for nearest neighbor classification: Taxonomy and empirical study," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 3, pp. 417–435, 2012. 3.1

[147] P. Hart, "The condensed neighborhood rule," *Information Theory, IEEE Transactions on*, vol. 14, pp. 515–516, 1968. 3.1

[148] C.H. Chou, B.H. Kuo, and F. Chang, "The generalized condensed nearest neighbor rule as a data reduction method," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*. IEEE, 2006, vol. 2, pp. 556–559. 3.1

[149] D.L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," *Systems, Man and Cybernetics, IEEE Transactions on*, , no. 3, pp. 408–421, 1972. 3.1

[150] F. Angiulli, "Fast nearest neighbor condensation for large data sets classification," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 19, no. 11, pp. 1450–1464, 2007. 3.1, 3.2.1, 3.3.1, 3.3.4

[151] J.C. Riquelme, J.S. Aguilar-Ruiz, and M. Toro, "Finding representative patterns with ordered projections," *Pattern Recognition*, vol. 36, no. 4, pp. 1009–1018, 2003. 3.1

[152] T. Raicharoen and C. Lursinsap, "A divide-and-conquer approach to the pairwise opposite class-nearest neighbor (poc-nn) algorithm," *Pattern recognition letters*, vol. 26, no. 10, pp. 1554–1567, 2005. 3.1

[153] A. Lumini and L. Nanni, "A clustering method for automatic biometric template selection," *Pattern Recognition*, vol. 39, no. 3, pp. 495–497, 2006. 3.1

[154] J.C. Bezdek and L.I. Kuncheva, "Nearest prototype classifier designs: An experimental study," *International Journal of Intelligent Systems*, vol. 16, no. 12, pp. 1445–1473, 2001. 3.1

[155] RA Mollineda, FJ Ferri, and E. Vidal, "An efficient prototype merging strategy for the condensed 1-nn rule through class-conditional hierarchical clustering," *Pattern Recognition*, vol. 35, no. 12, pp. 2771–2782, 2002. 3.1

[156] C.J. Veenman and D.M.J. Tax, "A weighted nearest mean classifier for sparse subspaces," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society*

Conference on. IEEE, 2005, vol. 2, pp. 1171–1176. 3.1

[157] P. Brodatz, *Textures: a photographic album for artists and designers*, vol. 66, Dover New York, 1966. 3.3, 3.3.2

[158] T. Randen and J.H. Husoy, "Filtering for texture classification: A comparative study," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, no. 4, pp. 291–310, 1999. 3.3, 3.3.2, 3.1

[159] L. Liu, P. Fieguth, D. Clausi, and G. Kuang, "Sorted random projections for robust rotation-invariant texture classification," *Pattern Recognition*, 2011. 3.3

[160] K. Skretting and J.H. Husøy, "Texture classification using sparse frame-based representations," *EURASIP journal on applied signal processing*, vol. 2006, pp. 102–102, 2006. 3.3.2, 3.1

[161] A. Di Lillo, G. Motta, and J.A. Storer, "Texture classification based on discriminative features extracted in the frequency domain," in *Image Processing, 2007. ICIP 2007. IEEE International Conference on*. IEEE, 2007, vol. 2, pp. II–53. 3.3.2, 3.1

[162] C. Xu and J.L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Trans Image Process*, vol. 7, no. 3, pp. 359–369, 1998. 4.1, 5.1

[163] V. Caselles, F. Catté, T. Coll, and F. Dibos, "A geometric model for active contours in image processing," *Numerische Mathematik*, vol. 66, no. 1, pp. 1–31, 1993. 4.1

[164] N. Paragios and R. Deriche, "Geodesic active regions and level set methods for supervised texture segmentation," *International Journal of Computer Vision*, vol. 46, no. 3, pp. 223–247, 2002. 4.1, 6

[165] R. Valdés-Cristerna, V. Medina-Bañuelos, and O. Yáñez-Suárez, "Coupling of radial-basis network and active contour model for multispectral brain MRI segmentation," *Biomedical Engineering, IEEE Transactions on*, vol. 51, no. 3, pp. 459–470, 2004. 4.1

[166] L.R. Ford Jr, D.R. Fulkerson, and A. Ziffer, "Flows in networks," *Physics Today*, vol. 16, pp. 54, 1963. 4.1

[167] C. Chen, J.A. Ozolek, W. Wang, and G.K. Rohde, "A general system for automatic biomedical image segmentation using intensity neighborhoods," *International Journal of Biomedical Imaging*, vol. 2011, 2011. 4.2, 4.3

[168] LC Evans, "Partial differential equations (graduate studies in mathematics vol 19)(providence, ri: American mathematical society)," 1998. 4.2

[169] A.I. Dow, S.A. Shafer, J.M. Kirkwood, R.A. Mascari, and A.S. Waggoner, "Automatic multiparameter fluorescence imaging for determining lymphocyte phenotype and activation status in melanoma tissue sections," *Cytometry*, vol. 25, no. 1, pp. 71–81, 1996. 5.1

[170] E. Bengtsson, C. Wahlby, and J. Lindblad, "Robust cell image segmentation methods," *Pattern Recognition Image Analysis*, vol. 14, no. 2, pp. 157–167, 2004. 5.1

[171] X. Chen, X. Zhou, and S.T.C. Wong, "Automated segmentation, classification, and tracking of cancer cell nuclei in time-lapse microscopy," *Biomedical Engineering, IEEE Trans-*

action on, vol. 53, no. 4, pp. 762–766, 2006. 5.1

[172] U. Adiga, R. Malladi, R. Fernandez-Gonzalez, and C.O. de Solorzano, "High-throughput analysis of multispectral images of breast cancer tissue," *Image Processing, IEEE Transaction on*, vol. 15, no. 8, pp. 2259–2268, 2006. 5.1

[173] C. Jung and C. Kim, "Segmenting clustered nuclei using h-minima transform-based marker extraction and contour parameterization," *Biomedical Engineering, IEEE Transaction on*, vol. 57, no. 10, pp. 2600–2604, 2010. 5.1

[174] B. Nielsen, F. Albregtsen, and H.E. Danielsen, "Automatic segmentation of cell nuclei in feulgen-stained histological sections of prostate cancer and quantitative evaluation of segmentation results," *Cytometry*, vol. 81, no. 7, pp. 588–601, 2012. 5.1

[175] S.J. Lockett, D. Sudar, C.T. Thompson, D. Pinkel, and J.W. Gray, "Efficient, interactive, and three-dimensional segmentation of cell nuclei in thick tissue sections," *Cytometry*, vol. 31, no. 4, pp. 275–286, 1998. 5.1

[176] C. Ortiz de Solorzano, R. Malladi, SA Lelievre, and SJ Lockett, "Segmentation of nuclei and cells using membrane related protein markers," *Journal of Microscopy*, vol. 201, no. 3, pp. 404–415, 2001. 5.1

[177] A. Dufour, V. Shinin, S. Tajbakhsh, N. Guillén-Aghion, J.C. Olivo-Marin, and C. Zimmer, "Segmenting and tracking fluorescent cells in dynamic 3-d microscopy with coupled active surfaces," *Image Processing, IEEE Transaction on*, vol. 14, no. 9, pp. 1396–1410, 2005. 5.1, 5.3.2

[178] G. Lin, M.K. Chawla, K. Olson, C.A. Barnes, J.F. Guzowski, C. Bjornsson, W. Shain, and B. Roysam, "A multi-model approach to simultaneous segmentation and classification of heterogeneous populations of cell nuclei in 3d confocal microscope images," *Cytometry*, vol. 71, no. 9, pp. 724–736, 2007. 5.1

[179] E. Hodneland, N.V. Bukoreshtliev, T.W. Eichler, X.C. Tai, S. Gurke, A. Lundervold, and H.H. Gerdes, "A unified framework for automated 3-d segmentation of surface-stained living cells and a comprehensive segmentation evaluation," *Medical Imaging, IEEE Transaction on*, vol. 28, no. 5, pp. 720–738, 2009. 5.1

[180] G. Li, T. Liu, A. Tarokh, J. Nie, L. Guo, A. Mara, S. Holley, and S. Wong, "3d cell nuclei segmentation based on gradient flow tracking," *BMC cell biology*, vol. 8, no. 1, pp. 40, 2007. 5.1

[181] C. Wählby, 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," *Journal of Microscopy*, vol. 215, no. 1, pp. 67–76, 2004. 5.1, 5.1, 5.2

[182] F. Li, X. Zhou, J. Ma, and S. Wong, "Multiple nuclei tracking using integer programming for quantitative cancer cell cycle analysis," *Medical Imaging, IEEE Transactions on*, vol. 29, no. 1, pp. 96–105, 2010. 5.1

[183] M.E. Plissiti, C. Nikou, and A. Charchanti, "Combining shape, texture and intensity features for cell nuclei extraction in pap smear images," *Pattern Recognition Letter*, vol. 32, no. 6, pp. 838–853, 2011. 5.1

[184] K. Nandy, P.R. Gudla, R. Amundsen, K.J. Meaburn, T. Misteli, and S.J. Lockett, "Automatic segmentation and supervised learning-based selection of nuclei in cancer tissue images," *Cytometry*, vol. 81, no. 9, pp. 743–754, 2012. 5.1

[185] F. Leymarie and M.D. Levine, "Tracking deformable objects in the plane using an active contour model," *Pattern Analysis and Machine Intelligence, IEEE Transaction on*, vol. 15, no. 6, pp. 617–634, 1993. 5.1

[186] P. Bamford and B. Lovell, "Unsupervised cell nucleus segmentation with active contours," *Signal. Pr.*, vol. 71, no. 2, pp. 203–213, 1998. 5.1

[187] A. Garrido and N. Pérez de la Blanca, "Applying deformable templates for cell image segmentation," *Pattern Recognition*, vol. 33, no. 5, pp. 821–832, 2000. 5.1

[188] L. Yang, P. Meer, and D.J. Foran, "Unsupervised segmentation based on robust estimation and color active contour models," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 9, no. 3, pp. 475–486, 2005. 5.1

[189] J. Cheng and J.C. Rajapakse, "Segmentation of clustered nuclei with shape markers and marking function," *Biomedical Engineering, IEEE Transaction on*, vol. 56, no. 3, pp. 741–748, 2009. 5.1

[190] D.P. Mukherjee, N. Ray, and S.T. Acton, "Level set analysis for leukocyte detection and tracking," *Image Processing, IEEE Transaction on*, vol. 13, no. 4, pp. 562–572, 2004. 5.1

[191] M. Plissiti and C. Nikou, "Overlapping cell nuclei segmentation using a spatially adaptive active physical model," *Image Processing, IEEE Transaction on*, 2012. 5.1

[192] O. Dzyubachyk, W.A. van Cappellen, J. Essers, W.J. Niessen, and E. Meijering, "Advanced level-set-based cell tracking in time-lapse fluorescence microscopy," *Medical Imaging, IEEE Transaction on*, vol. 29, no. 3, pp. 852–867, 2010. 5.1, 5.3.2

[193] A. Dufour, R. Thibeaux, E. Labruyere, N. Guillen, and JC Olivo, "3d active meshes: Fast discrete deformable models for cell tracking in 3d time-lapse microscopy," *Image Processing, IEEE Transaction on*, vol. 20, no. 7, pp. 1925–1937, 2011. 5.1

[194] P. Quelhas, M. Marcuzzo, A. M. Mendonca, and A. Campilho, "Cell nuclei and cytoplasm joint segmentation using the sliding band filter.," *Medical Imaging, IEEE Transaction on*, vol. 29, no. 8, pp. 1463–1473, 2010. 5.1

[195] T. Esteves, P. Quelhas, A.M. Mendonça, and A. Campilho, "Gradient convergence filters and a phase congruency approach for in vivo cell nuclei detection," *Machine Vision and Applications*, vol. 23, no. 4, pp. 623–638, 2012. 5.1

[196] P.R. Gudla, K. Nandy, J. Collins, KJ Meaburn, T. Misteli, and SJ Lockett, "A high-throughput system for segmenting nuclei using multiscale techniques," *Cytometry*, vol. 73, no. 5, pp. 451–466, 2008. 5.1

[197] D.P. McCullough, P.R. Gudla, B.S. Harris, J.A. Collins, K.J. Meaburn, M.A. Nakaya, T.P. Yamaguchi, T. Misteli, and S.J. Lockett, "Segmentation of whole cells and cell nuclei from 3-d optical microscope images using dynamic programming," *Medical Imaging, IEEE Transaction on*, vol. 27, no. 5, pp. 723–734, 2008. 5.1

[198] B.L. Luck, K.D. Carlson, A.C. Bovik, and R.R. Richards-Kortum, "An image model and

segmentation algorithm for reflectance confocal images of in vivo cervical tissue," *Image Processing, IEEE Transaction on*, vol. 14, no. 9, pp. 1265–1276, 2005. 5.1

[199] V.T. Ta, O. Lézoray, A. Elmoataz, and S. Schüpp, "Graph-based tools for microscopic cellular image segmentation," *Pattern Recognition*, vol. 42, no. 6, pp. 1113–1125, 2009. 5.1

[200] Y. Al-Kofahi, W. Lassoued, W. Lee, and B. Roysam, "Improved automatic detection and segmentation of cell nuclei in histopathology images.," *Biomedical Engineering, IEEE Transaction on*, vol. 57, no. 4, pp. 841–852, 2010. 5.1

[201] K.M. Lee and W.N. Street, "Model-based detection, segmentation, and classification for image analysis using on-line shape learning," *Machine Vision Application*, vol. 13, no. 4, pp. 222–233, 2003. 5.1

[202] K.M. Lee and W.N. Street, "An adaptive resource-allocating network for automated detection, segmentation, and classification of breast cancer nuclei topic area: image processing and recognition," *Neural Network, IEEE Transaction on*, vol. 14, no. 3, pp. 680–687, 2003. 5.1

[203] J. Fehr, O. Ronneberger, H. Kurz, and H. Burkhardt, "Self-learning segmentation and classification of cell-nuclei in 3d volumetric data using voxel-wise gray scale invariants," *Pattern Recognion*, pp. 377–384, 2005. 5.1

[204] C. Jung, C. Kim, S.W. Chae, and S. Oh, "Unsupervised segmentation of overlapped nuclei using bayesian classification," *Biomedical Engineering, IEEE Transaction on*, vol. 57, no. 12, pp. 2825–2832, 2010. 5.1

[205] W. D. Penny, K. J. Friston, J. T. Ashburner, S. J. Kiebel, and T. E. Nichols, Eds., *Statistical Parametric Mapping: The Analysis of Functional Brain Images*, Academic Press, 2006. 5.1

[206] MATLAB, *version 7.12.0 (R2011a)*, The MathWorks Inc., Natick, Massachusetts, 2011. 5.1

[207] G. K. Rohde, A. J. S. Ribeiro, K. N. Dahl, and R. F. Murphy, "Deformation-based nuclear morphometry: capturing nuclear shape variation in hela cells.," *Cytometry*, vol. 73, no. 4, pp. 341–350, 2008. 5.2.1

[208] G. Heitz, T. Rohlfing, and C.R. Maurer Jr, "Statistical shape model generation using nonrigid deformation of a template mesh," in *P. SPIE. Medical Imaging*. Citeseer, 2005, vol. 5747, pp. 1411–1421. 5.2.1

[209] J. Shlens, "A tutorial on principal component analysis," *Systems Neurobiology Laboratory, University of California at San Diego*, 2005. 5.2.1

[210] A. Lehmussola, P. Ruusuvuori, J. Selinummi, H. Huttunen, and O. Yli-Harja, "Computational framework for simulating fluorescence microscope images with cell populations," *Medical Imaging, IEEE Transaction on*, vol. 26, no. 7, pp. 1010–1016, 2007. 5.3.1

[211] A. Lehmussola, P. Ruusuvuori, J. Selinummi, T. Rajala, and O. Yli-Harja, "Synthetic images of high-throughput microscopy for validation of image analysis methods," *Proceeding of IEEE.*, vol. 96, no. 8, pp. 1348–1360, 2008. 5.3.1

[212] W. Wang, J. A. Ozolek, and G. K. Rohde, "Detection and classification of thyroid follicular lesions based on nuclear structure from histopathology images.," *Cytometry*, vol. 77, no. 5, pp. 485–494, 2010. 5.3.1

[213] 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," *Image Processing, IEEE Transaction on*, vol. 14, no. 9, pp. 1288–1299, 2005. 5.3.2

[214] G. Srinivasa, M.C. Fickus, Y. Guo, A.D. Linstedt, and J. Kovacevic, "Active mask segmentation of fluorescence microscope images," *Image Processing, IEEE Transaction on*, vol. 18, no. 8, pp. 1817–1829, 2009. 5.3.4, 5.2

[215] J. Deng, J. Hu, and J. Wu, "A study of color space transformation method using nonuniform segmentation of color space source," *Journal of Computers*, vol. 6, no. 2, pp. 288–296, 2011. 5.4, 6