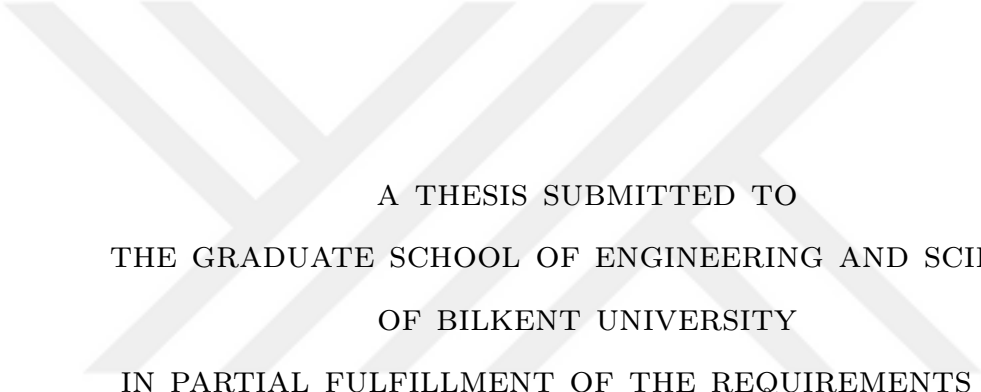


# **RULE BASED SEGMENTATION OF COLON GLANDS**



A THESIS SUBMITTED TO  
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE  
OF BILKENT UNIVERSITY  
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR  
THE DEGREE OF  
MASTER OF SCIENCE  
IN  
COMPUTER ENGINEERING

By  
Simge Yücel  
September 2018

# RULE BASED SEGMENTATION OF COLON GLANDS

By Simge Yücel

September 2018

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



---

Çiğdem Gündüz Demir(Advisor)

---

Can Alkan

---

Aybar Can Acar

Approved for the Graduate School of Engineering and Science:

---

Ezhan Karaşan  
Director of the Graduate School

# ABSTRACT

## RULE BASED SEGMENTATION OF COLON GLANDS

Simge Yücel

M.S. in Computer Engineering

Advisor: Çiğdem Gündüz Demir

September 2018

Colon adenocarcinoma, which accounts for more than 90 percent of all colorectal cancers, originates from epithelial cells that form colon glands. Thus, for its diagnosis and grading, it is important to examine the distortions in the organizations of these epithelial cells, and hence, the deformations in the colon glands. Therefore, localization of the glands within a tissue and quantification of their deformations is essential to develop an automated or a semi-automated decision support system. With this motivation, this thesis proposes a new structural segmentation algorithm to detect glands in a histopathological tissue image. This structural algorithm proposes to transform the histopathological image into a new representation by locating a set of primitives using the Voronoi diagram, to generate gland candidates by defining a set of rules on this new representation, and to devise an iterative algorithm that selects a subset of these candidates based on their fitness scores. The main contribution of this thesis is the following: The representation introduced by this proposed algorithm enables us to better encode the colon glands by defining the rules and the fitness scores with respect to the appearance of the glands in a colon tissue. This representation and encoding have not been used by the previous studies. The experimental results of our algorithm show that this proposed algorithm improves the segmentation results of its pixel-based and structural counterparts without applying any further processing.

*Keywords:* Histopathological image analysis, gland segmentation, Voronoi diagram, structural method.

# ÖZET

## KALIN BAĞIRSAK BEZLERİNİN KURALA DAYANARAK BÖLÜTLENMESİ

Simge Yücel  
Bilgisayar Mühendisliği, Yüksek Lisans  
Tez Danışmanı: Çiğdem Gündüz Demir  
Eylül 2018

Tüm kolorektal kanserlerin yüzde doksandan fazlasını oluşturan kolon adenokarsinomu, kolon bezlerini oluşturan epitel hücrelerden kaynak almaktadır. Dolayısıyla, bu kanserin tanı ve derecelendirmesinde, epitel hücrelerin organizasyonlarındaki bozuklukların, bundan dolayı da kolon bezlerindeki deformasyonların incelenmesi önemlidir. Bu nedenle, kolon dokusundaki bezlerin yerlerinin tespit edilmesi ve deformasyonlarının nicelenmesi, otomatik veya yarı otomatik karar destek sistemlerinin geliştirilmesi için esastır. Bu motivasyonla, bu tez, histopatolojik doku görüntülerindeki bezleri saptamak için yeni bir yapısal bölütleme algoritması önermektedir. Bu yapısal algoritma, Voronoi diyagramı kullanarak histopatolojik görüntü üzerinde bir temel öge kümesi yerleştirmeyi ve bu şekilde görüntüyü yeni bir gösterime dönüştürmeyi; bu yeni gösterim üzerinde kurallar tanımlayarak bez adaylarını üretmeyi; ve uygunluk skorlarına göre bu adaylar arasından alt küme seçen tekrarlı bir algoritma tasarlamayı önermektedir. Bu tezin başlıca katkısı; önerilen algoritma ile ortaya konan gösterimin, bezlerin kolon dokusundaki görünümüne göre kural ve uygunluk skoru tanımlayarak, kolon bezlerinin daha iyi kodlanmasına olanak sağlamasıdır. Bu gösterim ve kodlama daha önceki çalışmalarda kullanılmamıştır. Algoritmamızın deneysel sonuçları, önerilen bu algoritmanın, ek bir işlem uygulamadan, piksel tabanlı ve yapısal benzerlerinin bölütleme sonuçlarını iyileştirdiğini göstermiştir.

*Anahtar sözcükler:* Histopatolojik görüntü analizi, bez bölütlemesi, Voronoi diyagramı, yapısal yöntem.



## Acknowledgement

First of all, I would like to thank my advisor igdem Gündüz Demir for guiding and supporting me through this study. It was an honor for me to work with her. I learned lots of things from her, and without her support, I could not able to complete this study. I also would like to thank my jury members Can Alkan and Aybar Can Acar for their valuable feedback to improve this study.

I owe my deepest gratitude to my parents, Feray and Bilal, and my lovely sister Selin. This thesis would not be possible without their support, help, and patience. I am very lucky to have such a family. I think me and my sister have successfully completed a difficult year for both of us. When we look back, we will remember the Sundays that we study together with a smile.

I would like to indicate my special thanks to Ebru Ateş for being a sister to me and making my master days unforgettable. She always supported me in my every decision and most importantly tried to protect me in any case. We had very enjoyable days and I am sure we will add more beautiful memories to the existing ones.

Especially I would like to thank three valuable people. I want to start with Troya ağıl Köylü for his endless motivation, friendship, and help through my study. He was a great lab pier and I will never forget his moral support for this year. Secondly, I want to thank Seda Gülkesen who became a sister and a friend through my master study. I will never forget the joyful Friday nights that we met for studying. Finally, I want to present my special thanks to İstemi Rahman Bahçeci for his great support, especially during the last month of my study. I am very happy to know him personally. He is one of the most virtuous and honest person I have ever met and I hope he will never change.

It has been eight years since I started studying at Bilkent University, and during this time I met many great people. I would like to thank Cemil Kaan Akyol, Miray Ayşen, Gülfem Demir, Can Fahrettin Koyuncu, Caner Mercan, Muhsin

Can Orhan, Ali Burak Ünal, Başak Ünsal and Muhammed Ali Yeşilyaprak for their friendship, collaboration, and support. Likewise, I would also like to thank all the professors who make a contribution to my education so far.

I want to thank my team leader, teammates and especially Naciye Öztürk for their understanding and support during this hard year.

Finally, I would like to thank two valuable and highly reputable person. My deepest thanks to our great leader Mustafa Kemal Atatürk. If I received this education and became an engineer today as a girl, I owe to him. I also would like to thank the founder of our university dear İhsan Doğramacı for giving importance to science and leaving us such a great educational establishment.

# Contents

|          |                                      |           |
|----------|--------------------------------------|-----------|
| <b>1</b> | <b>Introduction</b>                  | <b>1</b>  |
| 1.1      | Contributions . . . . .              | 5         |
| 1.2      | Outline . . . . .                    | 6         |
| <b>2</b> | <b>Background</b>                    | <b>8</b>  |
| 2.1      | Domain Description . . . . .         | 8         |
| 2.2      | Related Work . . . . .               | 9         |
| 2.2.1    | Pixel Based Methods . . . . .        | 9         |
| 2.2.2    | Structural Methods . . . . .         | 14        |
| <b>3</b> | <b>Methodology</b>                   | <b>17</b> |
| 3.1      | Voronoi Representation . . . . .     | 18        |
| 3.2      | Gland Candidate Generation . . . . . | 21        |
| 3.2.1    | Candidate Set-1 . . . . .            | 21        |
| 3.2.2    | Candidate Set-2 . . . . .            | 21        |

|          |                                               |           |
|----------|-----------------------------------------------|-----------|
| 3.3      | Iterative Gland Localization . . . . .        | 27        |
| 3.3.1    | Metric Definition . . . . .                   | 27        |
| 3.3.2    | Iterative Gland Selection Algorithm . . . . . | 30        |
| <b>4</b> | <b>Experiments</b>                            | <b>36</b> |
| 4.1      | Dataset . . . . .                             | 36        |
| 4.2      | Evaluation . . . . .                          | 37        |
| 4.3      | Parameter Selection . . . . .                 | 37        |
| 4.4      | Results . . . . .                             | 38        |
| <b>5</b> | <b>Conclusion</b>                             | <b>47</b> |

# List of Figures

|     |                                                                       |    |
|-----|-----------------------------------------------------------------------|----|
| 1.1 | Structure of a gland. . . . .                                         | 2  |
| 1.2 | Sample colon tissue image which contains white gaps. . . . .          | 4  |
| 2.1 | Gland structure. . . . .                                              | 9  |
| 2.2 | Examples of colon tissues containing gland structures. . . . .        | 11 |
| 3.1 | Steps of our algorithm. . . . .                                       | 19 |
| 3.2 | Steps of generating the Voronoi representation. . . . .               | 20 |
| 3.3 | White connected components . . . . .                                  | 22 |
| 3.4 | Connected components generated from nuclear Voronoi polygons. . . . . | 23 |
| 3.5 | Illustration for the conversion method. . . . .                       | 25 |
| 3.6 | Steps of the candidate set generation. . . . .                        | 26 |
| 3.7 | Examples for the eliminated candidates. . . . .                       | 29 |
| 3.8 | Candidate examples that share Voronoi polygons. . . . .               | 31 |
| 3.9 | Eliminated candidate examples through area thresholding. . . . .      | 33 |

|      |                                                         |    |
|------|---------------------------------------------------------|----|
| 3.10 | The glands selected by the iterative algorithm. . . . . | 35 |
| 4.1  | Example results from the training set. . . . .          | 40 |
| 4.2  | Example results from the test set. . . . .              | 41 |



# List of Tables

|     |                                                                                                  |    |
|-----|--------------------------------------------------------------------------------------------------|----|
| 4.1 | Quantative results obtained on the training images when $n=40$ . . .                             | 42 |
| 4.2 | Quantative results obtained on the training images when $n=60$ . . .                             | 43 |
| 4.3 | Quantative results obtained on the training images when $n=80$ . . .                             | 44 |
| 4.4 | The results obtained by our method and the pixel-based methods,<br>for the training set. . . . . | 45 |
| 4.5 | The results obtained by our method and the pixel-based methods,<br>for the test set. . . . .     | 45 |
| 4.6 | Results of the object-graph method and our proposed method, for<br>the training set. . . . .     | 46 |
| 4.7 | Results of the object-graph method and our proposed method, for<br>the test set. . . . .         | 46 |

# Chapter 1

## Introduction

Colorectal cancer is the third most commonly diagnosed cancer type in Western Europe and North America [1, 2]. It is ranked in the fourth place among all cancer-related deaths and 1.1 million deaths are expected by 2030 [3]. The most common type of colorectal cancer is adenocarcinoma, accounting for more than 90 percent of all colorectal cancers. Colon adenocarcinoma originates from epithelial cells, which line up the luminal surface of colon and form colon glands together with the lumina (see Figure 1.1). Thus, for diagnosis and grading of this cancer type, pathologists examine the distortions in the organizations of these epithelial cells, and hence, deformations in the colon glands.

Similar to all cancer types, the colon adenocarcinoma is diagnosed and graded by a histopathological examination of fixed and stained colon tissue sections. In this process, pathologists examine a colon tissue under a microscope and identify whether or not there exist deformations in the glands, and if any, the degree of these deformations. Thus, in order to develop an automated or a semi-automated decision support system, which will help the pathologists make faster and more objective decisions, it is essential to identify gland locations within a tissue and to quantify their deformations. With this motivation, this thesis focuses on the first part, for which it proposes a new gland segmentation algorithm.



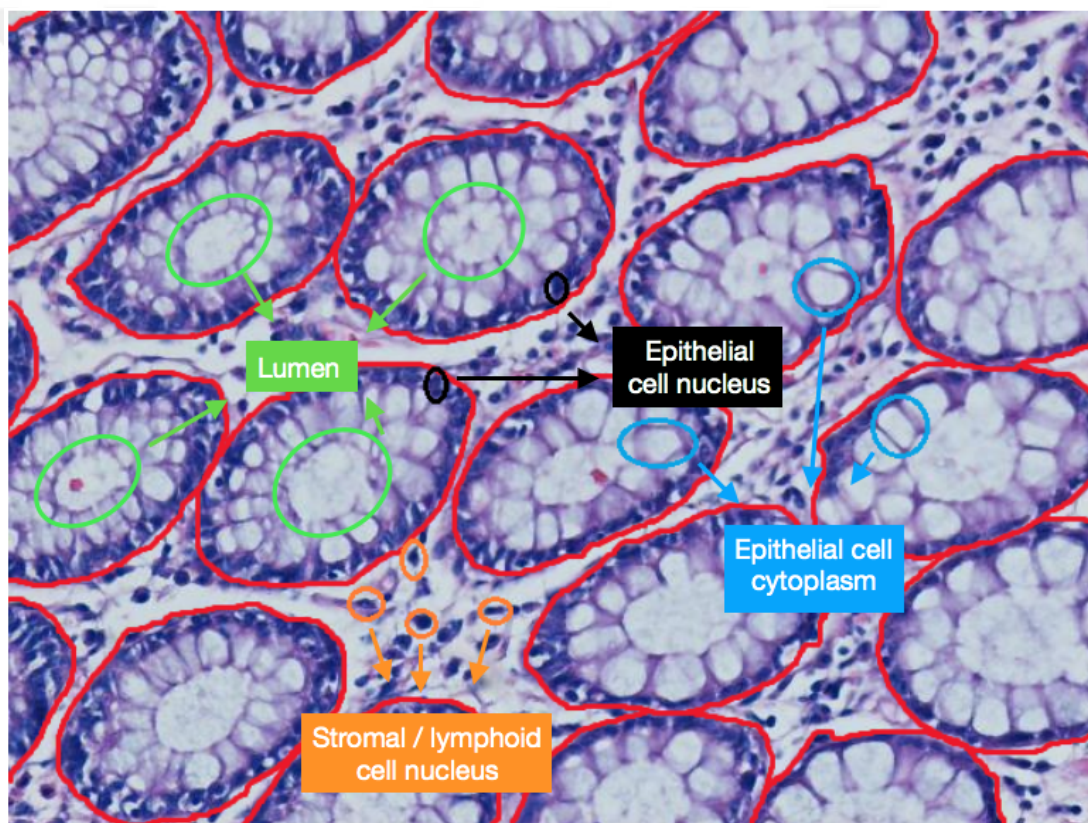


Figure 1.1: Sample colon tissue image, on which tissue components are illustrated. Additionally, gland boundaries are drawn with red in the image.

In the literature, the existing gland segmentation methods can mainly be categorized into two: pixel-based and structural methods. The pixel-based methods commonly assign the predefined labels (e.g., nucleus and lumen labels) to image pixels and then apply a post-processing technique on these image labels to form individual glands. These methods indeed rely on the fact that, in a colon tissue image, a typical gland has a large enough whitish luminal region and this region is surrounded by a thick enough line of purplish nuclear pixels. The earlier methods assign labels to the pixels based on their intensities, usually by thresholding [4] or using a simple classifier [5]. On the other hand, these labeling methods may lead to incorrect class labels especially when there exist variations and noise in the pixel intensities. Additionally, the commonly used post-processing techniques (e.g., region growing) applied afterwards may fail when there exist pixel-level variations and imperfections in the gland pixels (e.g., when there exist a considerable large white gap in the line of nuclear pixels due to the imperfections in the tissue preparation steps, see Figure 1.2). More recently, the use of deep learning classifiers [6], [7], [8] alleviates the problems related with incorrect pixel labeling. However, they are still prone to the pixel-level imperfections in the image, since they usually require a post-processing step (e.g., applying small area elimination followed by morphological operations) that is applied directly on the pixel labels or their probabilities. Additionally, the deep learning methods necessitate a considerable number of annotated images, which sometimes is difficult to obtain.

The structural methods typically decompose an image into a set of multi-type primitives (e.g., nuclear and lumen primitives), for instance by locating circles [9] and superpixels [10], and then use the neighborhood information of these primitives to form the glands. Since they run on the primitives (i.e., use the spatial organization of these primitives instead of directly using that of the pixels), these methods are expected to be less susceptible to noise and variations observed at the pixel level (e.g., the one illustrated in Figure 1.2). In this thesis, we propose a new gland segmentation algorithm of this kind.

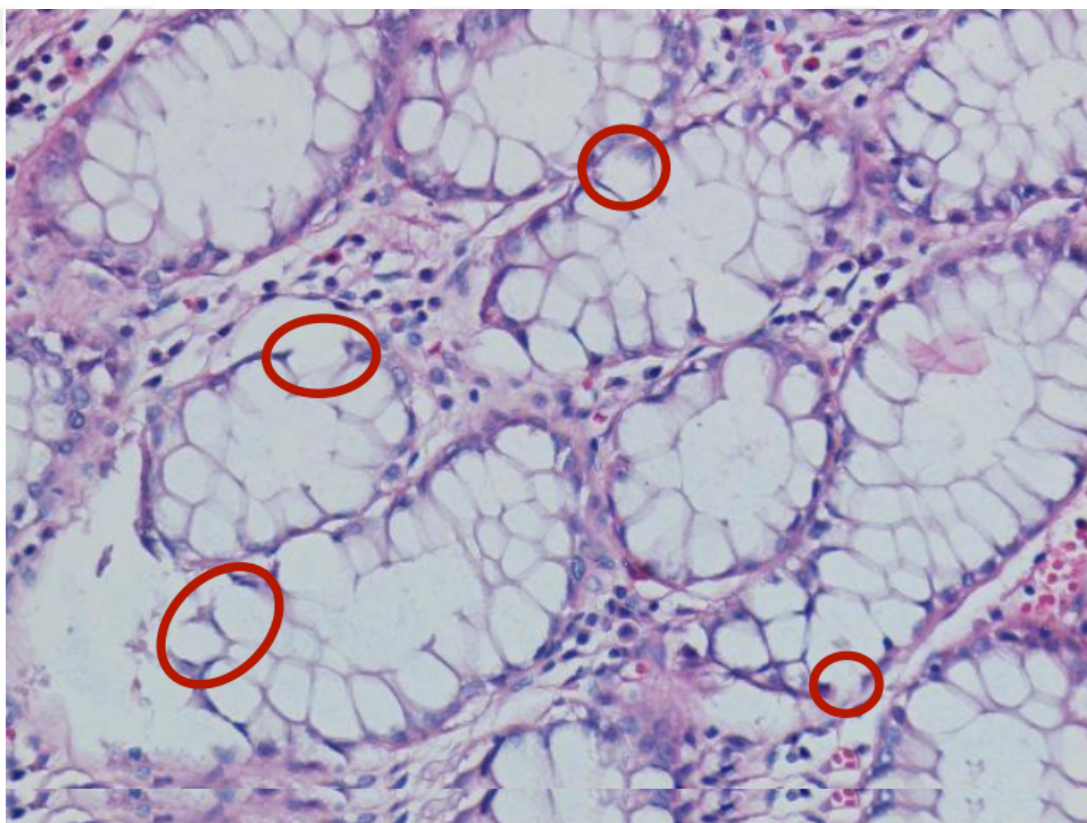


Figure 1.2: Sample colon tissue image which contains white gaps in between epithelial nuclei due to imperfection in the tissue preparation process.

## 1.1 Contributions

The proposed algorithm 1.) defines a representation by locating a set of new primitives, 2.) generates gland candidates by defining a set of rules on this representation according to the spatial distribution of these primitives, and 3.) devises an iterative gland selection algorithm based on the fitness score it proposes.

In particular, the proposed representation decomposes an image into nuclear and white circles, similar to our previous work [9], but then it constructs a Voronoi diagram on the centroids of these circles. Compared to the case where only the circles are used, this Voronoi diagram representation makes easier to define rules for generating the gland candidates, especially when there exist noise and variations.

This thesis defines the gland generation rules and its fitness score, which we call *GlandScore*, with the motivation of mimicking the appearance of glands in a colon tissue. Particularly, it uses the observations given below. The proposed Voronoi diagram representation facilitates the encoding of these observations as well as the quantification of the fitness measures that they use. These encodings and the definition of their corresponding fitness measures have not been used by the previous methods, and thus, constitute the main contribution of this thesis.

- A colon gland should contain a white luminal region but no nuclei inside. In our representation, this corresponds to finding a large enough connected component of white Voronoi polygons. This component should contain no nuclear polygon inside (or only a few of them when there exist artifacts in the image).

To encode this observation, a gland candidate is formed by identifying each connected component that contains white polygons more than the threshold and combining it with its adjacent nuclear polygons. Then, to quantify the fitness of this candidate, the *precision* measure is defined as the ratio of the number of the nuclear polygons located on the candidate's border to the number of all nuclear polygons belonging to the same candidate.

- A colon gland should be surrounded by epithelial nuclei. In our representation, this corresponds to having a row of nuclear Voronoi polygons that confine the white connected component. This connected component should be confined by the nuclear polygons entirely (or only a few of them could be missing when there exist noise in the image).

To encode this observation, a gland candidate is formed similarly. Its fitness is quantified by the *recall* measure that is defined as the ratio of the number of the nuclear polygons located on the candidate's border to the number of all polygons (nuclear and white ones) on the same border. Then, by taking the harmonic mean of its *precision* and *recall* measures, the *Fscore* of the candidate is calculated.

- A colon gland should of the tubular shape. In our representation, this corresponds to selecting the compact candidates.

After generating a gland candidate in a similar way, its compactness is quantified by the *ConvexRatio* measure, which is the ratio of the candidate's area to that of its convex hull. At the end, for the same gland candidate, all these measures are combined by the *GlandScore*, which is the product of its *Fscore* and *ConvexRatio* measures.

After generating gland candidates and calculating their *GlandScores*, this thesis defines an iterative algorithm that selects a subset of these candidates based on their *GlandScores*. In each iteration, it selects the best candidate and updates the remaining ones that share some polygons with the selected one. This iterative algorithm continues until there remains no candidate with a high enough *GlandScore*.

## 1.2 Outline

The structure of the remaining of this thesis is as follows. Chapter 2 gives a brief information about the domain and reviews the existing gland segmentation methods in the literature. Then, Chapter 3 provides the details of the proposed gland

segmentation algorithm, explaining its image representation, candidate gland generation, and iterative gland selection steps. This chapter explicitly explains the rules used for gland generation and gives the definitions of the measures used for quantification. Next, Chapter 4 presents experimental settings as well as the results of the proposed algorithm. Finally, Chapter 5 concludes the thesis and presents its future work.



# Chapter 2

## Background

In the first part of this chapter, we will give general information about glands, their structure, and the staining method. In the second part, we will explicate the existing gland segmentation methods and discuss the feasible and weak sides of these methods.

### 2.1 Domain Description

The epithelial tissue is made up of glands which produce and secrete hormones, enzymes, and chemical substances that regulate the activity of cells or organs in the body. Glands exist in many different parts of the human body such as lungs, pancreas, and colon. The glands located in colon are in a tubular shape. Figure 2.1 shows the tubular structure of a gland and Figure 1.1 explicitly indicates the parts of the glands on a tissue sample [11]. At the center of a gland, there is a big white area which is called lumen. This area is surrounded by columnar epithelial cells. Outer side of these cells, epithelial cell nuclei exist which compose the boundary of a gland. In the stromal part of tissues, stromal cell nuclei are scattered among the glands.

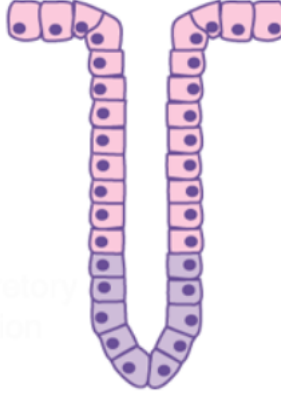


Figure 2.1: Gland structure.

In this thesis, we used images of the colon tissues stained with the hematoxylin-and-eosin (H&E) technique. It is the routinely used staining technique in histopathology. This method stains nucleus with hematoxylin and cytoplasm with eosin [12]. As a result of this staining, the nuclei in images look purple and the cytoplasm and stroma look pink.

## 2.2 Related Work

Gland segmentation is an important step in the automated or semiautomated detection of colon adenocarcinoma. In the literature, there exist two main approaches to this problem: pixel-based and structural.

### 2.2.1 Pixel Based Methods

Earlier studies of gland segmentation include pixel-based methods which make use of the intensities of pixels. However, after deep learning, these studies gain another perspective. For this reason, existing pixel-based methods can be categorized into two main classes such as ‘Before Deep Learning’ and ‘After Deep Learning’.



Before deep learning, many studies dealt with the gland detection problem by focusing on lumen detection. They utilize the colors' intensity difference to differentiate luminal and nuclear areas. The study by Wu *et al.*, proposes a method takes an image and applies a simple thresholding to convert it to a binary image [4]. Black parts of the binary image correspond to nuclei and white parts to lumina and cytoplasms. Then, in order to define the initial seeds, big white areas are found, which most probably correspond to the lumina. To find the luminal parts, a window of radius  $R_0$  is moved on the image and the white areas, which completely cover this window, are chosen as potential seeds.

After finding potential seeds, they are expanded by applying iterative morphological dilation operation with a circular structuring element. At each iteration, dilated parts are added to the seed's area if their pixels are white. The growth of regions is terminated when the pixels reach the nucleus chains. There are some false regions in stroma which form similar structures as glands by growing. In order to prevent the growth of these false regions, an iteration threshold is put and the areas are eliminated if the growth of the region does not stop until this threshold. Then, in order to find the false detected luminal regions, which are not eliminated in the previous step, the thickness of the boundaries are checked by looking at the ratio between the number of nucleus pixels and the number of total pixels on the boundary. If the ratio is very small, this gland is considered false and eliminated.

In another study by Naik *et al.*, a Bayesian classifier is used which classifies pixels based on their intensity values into three classes: nucleus, lumen, and cytoplasm [5]. Manually labeled pixels are used for training of each class. According to the color values of the training dataset's pixels, the priors and the probability density functions are generated. Then, for each pixel of a given image, the posterior probability of a pixel belonging to a particular class is calculated by using these priors and likelihoods using the Bayes theorem. By thresholding the posterior probabilities with empirically determined values, potential lumen, cytoplasm, and nuclear regions are detected. From these detected lumens, the ones that are most probably noise are eliminated according to the lumen size, which is defined during the training process by using size histograms. After getting

the possible starting points, a level-set algorithm is performed to find the gland regions. A level-set curve is fitted to the detected lumen areas and the algorithm runs until the difference in the contours is below an empirically determined threshold between two iterations. When the difference is higher than this predefined threshold, it means that lumen pixels reach to nucleus pixels, and the algorithm stops. The false glands are eliminated by looking at the contours of their boundaries.

Banwari *et al.* propose a similar study for the purpose of gland segmentation [13]. Their study consists of three steps which are preprocessing, area thresholding, and closing. In the preprocessing part, the red channel of the given image is taken and a histogram equalization is applied in order to enhance the contrast. The output of these steps gives the brightest part of the image. Then an average filtering is applied for smoothing the borders of these parts and possible lumen areas are obtained by thresholding according to pixels' intensities. In the second step of the study, they apply an area thresholding in order to eliminate the small areas. To the remaining areas, the morphological dilation operation is applied. Finally, in the third step, the closing operation is applied to the binary image in order to get the output.

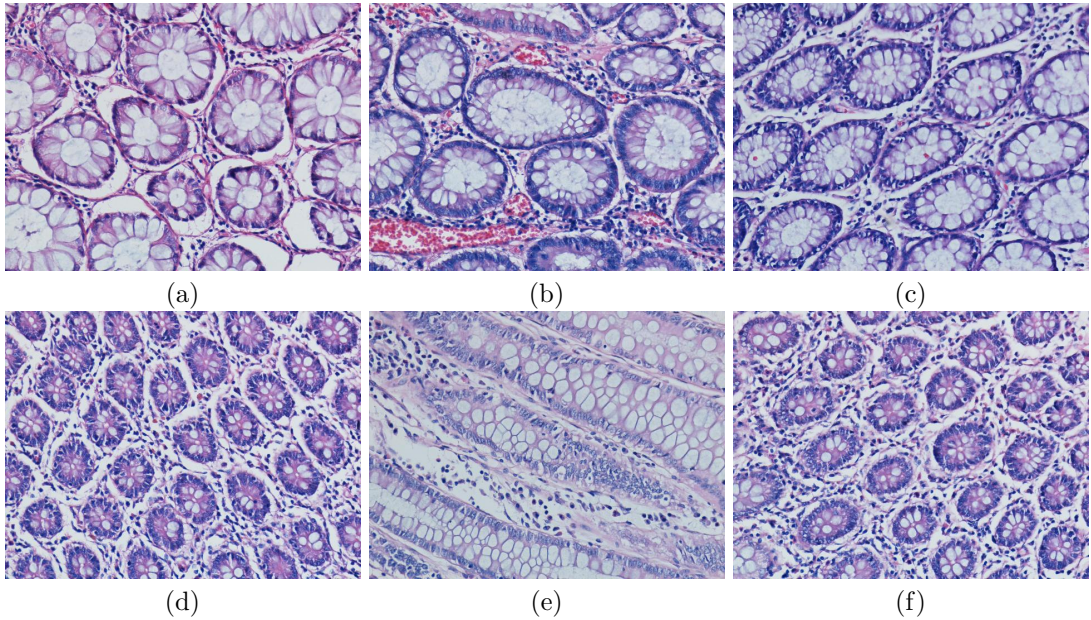


Figure 2.2: Examples of colon tissues containing gland structures.

As a result of these studies, it is seen that finding the lumen area, and expanding the area with a region growing algorithm is not always appropriate to successfully find glands. These studies use empirically determined area thresholds to distinguish the possible lumen areas from the potential noise. Although this idea may give satisfying results for the images in Figures 2.2 (a), 2.2 (b), and 2.2 (c), it will not give good results for the images in Figures 2.2 (d), 2.2 (e), and 2.2 (f), since their lumen areas are not big enough. Secondly, labeling methods may lead to incorrect class labels especially when there exist variations and noise in the pixel intensities. Also for the same reason, the post-processing techniques (e.g., region growing) may fail on such images.

More recently, researchers start to use deep learning classifiers to alleviate the problems related with incorrect pixel labeling. One of the latest studies by Chen *et al.* proposes a deep contour-aware network to solve gland segmentation problem [6]. This study uses a fully convolutional neural network for multi-level feature representation. The network takes an image as input and outputs the segmentation probability map and depicted contours of gland objects simultaneously. The network’s architecture consists of down-sampling and up-sampling paths. First of all, the network down-samples the image with convolutional and max-pooling layers. The results of the down-sampled layers are up-sampled with two different branches to give the segmented object and contour masks separately as outputs. Instead of implementing separate networks for the segmentation of objects and the detection of contours, they design a single network for both purposes. Finally, both results are fused together to obtain the final output image. Then, post-processing steps are applied to finalize the resulting image. These steps include small area elimination, smoothing with a disk filter, and filling the holes.

In another study by Manivannan *et al.*, a method is presented which uses exemplar label patches for structure prediction [7]. In the training process, label patches are extracted from the ground truths of the training images. In order to get a set of label exemplars, these patches are clustered with the k-means clustering algorithm. After obtaining K exemplars, linear classifiers are defined for each of them to learn their structures. In the test process, for each pixel

of a given test image, the classifiers output  $K$  separate probabilities, each of which corresponds to each exemplar and the pixel is labeled with the one whose probability is the highest. The higher values of the output map indicate the possible gland locations. In order to get glands from the output map, they apply a fixed threshold, which is defined in the training process. Then, for separating the adjacent glands, they apply the morphological erosion operation with a circular structuring element, and after this step, eliminate small connected components. To finalize the result, they apply the morphological dilation operation with a circular structuring element whose radius is twice the size of the structuring element's radius used in erosion.

Besides all these studies, Xu *et al.* propose an algorithm for gland segmentation using deep multichannel neural networks [8]. In this approach, they aim to detect the pixels of glands and recognize individual glands at the same time. In the proposed algorithm, they make use of multichannel learning; one channel for foreground segmentation, one channel for edge detection, and one channel for individual gland recognition. The foreground segmentation channel distinguishes glands from the background by using a fully convolutional neural network. The edge detection channel uses a Holistically-nested Edge Detector (HED) to detect boundaries between glands [14]. HED is a convolutional neural network based solution for edge detection. The individual gland recognition channel makes use of Faster Regional CNN (R-CNN) to get location information and Region Proposal Network (RPN) to form the final gland detection result [15]. After receiving the outputs of these three channels, which contain the information about the region, location, and boundary, a fusion algorithm is applied to combine them. In order to fuse these three channels, a convolutional neural network is used. To conclude, four different neural networks are used in this study, three of them for extracting the features, and the remaining one is to combine the outputs of the first three ones.

Although using neural networks for gland segmentation yields promising results, it has the disadvantage of necessitating a considerable number of annotated images and requiring a relatively larger computational time.

### 2.2.2 Structural Methods

In the literature, there are few studies which handle gland segmentation via structural methods. The common approach of these methods is decomposing the image into primitives and forming gland structures via using the neighborhood information of these primitives.

Gunduz-Demir *et al.* present one of the first studies in this field [9]. In the first step of their proposed algorithm, they quantize the image into three clusters (nucleus, stroma, and lumen clusters) by using the k-means algorithm. Then, they locate circular objects on the nucleus and lumen clusters, separately. They construct an object graph on lumen objects, and extract their local features on this graph. For each lumen object, they extract structural features, considering the neighborhood information between this object and its closest nuclear and lumen objects. Then, these features are used to group the lumen objects into two classes as gland and non-gland, by using the k-means algorithm. In the second part of the algorithm, another object-graph is constructed on nucleus circles. Edges are drawn between each nucleus object and its closest objects. Finally, a region growing algorithm is applied to the gland objects. The algorithm stops when a gland structure reaches to the edges of the nucleus-graph. At the end of the region growing process, small regions are eliminated, and a decision tree classifier is used to eliminate false glands.

In another study, Sirinukunwattana *et al.* propose a Random Polygons Model for modeling glandular structures on images [10]. In this model, first of all, they decompose the image into superpixels by using the Simple Linear Iterative Clustering (SLIC) algorithm, and for each superpixel a feature vector is extracted which contains color and texture information [16]. Then, a random forest classifier is trained on these features in order to get a glandular probability for each superpixel. By using superpixels' probabilities, a map is obtained which indicates the glandular probability of each pixel on the image. Then, initial seed areas are obtained by thresholding the probability map, and in order to get more reliable results, the morphological erosion and smoothing operations are applied to these

areas.

In the second part of the algorithm, by applying Otsu’s threshold to the hematoxylin channel of the image, the approximate locations of nuclei are obtained [17]. On those locations, randomly drawn vertices are located. However, the distance between any two vertices should be at least a pre-defined distance threshold  $d$ , otherwise, the vertex is rejected. Then, for each seed area, the closest vertices are found and by drawing edges between these vertices a polygon is obtained. Moreover, each seed area is expanded by  $m_{expand}$  pixels from all sides of its boundary. Finally, the Reversible-Jump Markov Chain Monte Carlo (RJMCMC) method takes the obtained polygons, the vertices which are located on the expanded pixels of these polygons and the probability map as inputs and outputs a sequence of polygons for each seed area [18]. Among these polygons they use *maximum a posteriori* polygon to estimate the glandular structure. For the elimination of false glands, they defined two criteria. Firstly, the polygons are eliminated whose number of vertices less than or equal to a pre-defined vertex number. Secondly, if the square root of the polygon’s area is less than or equal to the area threshold, this polygon is also eliminated.

Fu *et al.* propose an unconventional modeling for gland detection [19]. First of all, they convert the image from the Cartesian space to the polar space. Then, to infer the possible gland boundaries they introduce a random field model. They propose to infer the glands by using the knowledge that they form closed shapes. Their idea is, if they place the co-ordinate’s center inside a gland and then convert it to the polar space, they expect to see a line structure in the converted image. According to this idea, if they observe a line this means there is a gland. Therefore, they formulate the gland detection problem as detecting the lines in the polar space.

In the Cartesian space, each point of the given image is subsampled by a circular window with a radius of  $r_{max}$ . Then each window is converted to the polar space in order to get a polar image. A Conditional Random Fields (CRF) model is defined which consists of random variables  $X$  and  $Y$  [20]. Each row of the polar image is represented with  $X_i$  and they are labeled with  $Y_i$  where  $Y$

corresponds to the gland's boundary position at that row. In order to infer  $Y$ , a graph is constructed for each  $X_i$ . To make an inference from these graphs, two chain structures are used. To obtain the chain structures, they generate two polar images, one is with  $\theta$  ranging from 0 to  $\Pi$  and the other is with  $\theta$  ranging from  $\Pi$  to  $2\Pi$ . After generating the chain structures, the Viterbi algorithm is applied on each structure to infer the optimal  $Y$  [21]. Then, the results are combined with a heuristic method which is presented by them.

In this thesis, a new structural method is proposed for the purpose of gland segmentation. Similar to the existing methods, we obtain structures from our primitives. However, our proposed algorithm enables to encode the appearance of glands by using the predefined gland generation rules and their fitness scores, which have not been used by these previous methods.

## Chapter 3

# Methodology

Our proposed method transforms a histopathological image into a new representation that allows us to devise an iterative gland localization algorithm. In order to transform the image to this representation, Voronoi diagrams are used which represent the given image's subregions as Voronoi polygons. In the generation process of these polygons, they are labeled as white and nuclear according to the pixels' colors that they are placed onto. Then, considering these polygons' locations relative to each other, a candidate set is generated.

A typical structure of a gland consists of a white area known as *lumen*, which is surrounded by a black border that corresponds to epithelial cell nuclei. In order to generate a candidate set, connected components on white Voronoi polygons are first found. Then, to mimic the structural organization of a gland, one-row Voronoi polygons are added to each white connected component and candidates are obtained. After getting the one-row added version of the candidates, we select them according to their fitness scores. To quantitatively define this fitness score, the *GlandScore* metric is introduced.

Our iterative algorithm selects the glands by starting from the one with the highest *GlandScore*. At each selection, the candidates that share Voronoi polygons with the selected candidate are updated. The algorithm stops when there



is not any candidate whose *GlandScore* is higher than the predefined threshold. Figure 3.1 indicates the steps of our proposed algorithm.

### 3.1 Voronoi Representation

The Voronoi diagram representation of a histopathological image is obtained by following three substeps: pixel labeling, circle localization, and Voronoi diagram construction. For the first substep, a deconvolution operation is first applied to the image, which is stained with the hematoxylin-and-eosin technique [22]. This deconvolution operation is used to emphasize the hematoxylin channel of the given image. Then, the hematoxylin channel of the image is taken and according to this channel’s average intensity value, pixels are divided into two groups. If the intensity value of a pixel is greater than the average, we label the pixel as white, otherwise, we label it as nuclear.

After separating pixels into two groups, the next substep locates circles on each of these groups using the algorithm that was implemented in our research group [23]. In this algorithm, circles are located on the given pixels’ group starting from the largest circle to the one whose radius is at least  $r_{min}$ .

Then the last step takes the centers of these circles and locates a Voronoi diagram onto them. Figure 3.2 indicates the steps of getting the Voronoi representation for a given image. At the first step, Voronoi polygons are located on the given image. In order to make them more noticeable, each Voronoi polygon is painted with another color in Figure 3.2 (b). The Voronoi polygons are labeled as white and nuclear according to the labels of the circles that they are placed onto. In other words, if a Voronoi polygon is generated for the centroid of a circle which is located on white pixels, then this Voronoi polygon is labeled as white, otherwise, it is labeled as nuclear. Figure 3.2 (c) shows the white and nuclear Voronoi polygons for the given image.

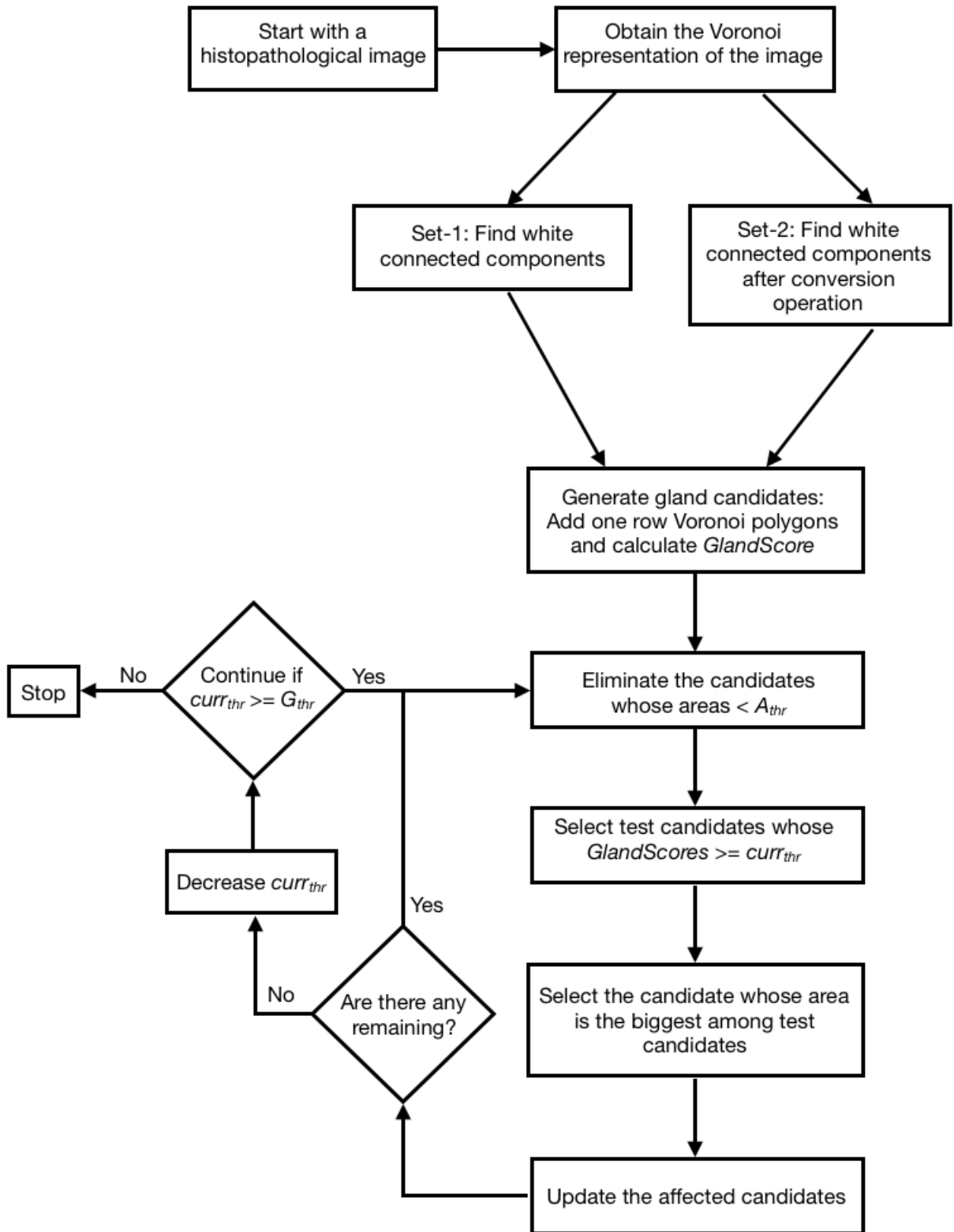


Figure 3.1: Steps of our algorithm.

After generating and labeling all Voronoi polygons, white connected components are found. Figure 3.2 (d) indicates the white connected components. By means of this representation, we are able to define gland candidates easily. Instead of working on pixels, we use Voronoi polygons and aim to generate possible gland candidates on them.

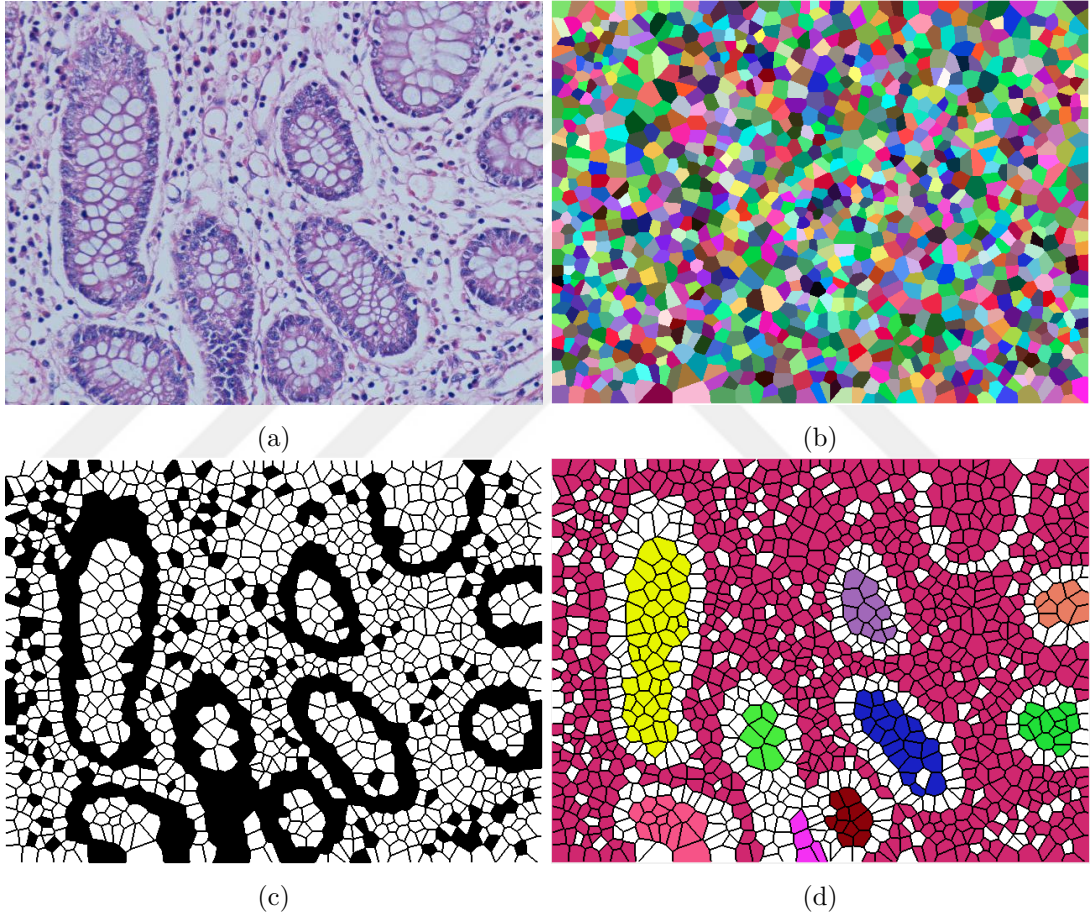


Figure 3.2: Steps of generating the Voronoi representation: (a) Histopathological image, (b) Voronoi diagram of the given image, (c) labeled Voronoi polygons, and (d) connected components generated from white Voronoi polygons.

## 3.2 Gland Candidate Generation

After getting nuclear and white connected components, we generate gland candidates by utilizing this representation. In the ideal case, no gaps should be found in between the Voronoi polygons that correspond to nuclei belonging to a single gland. However, there are some gaps between them and due to this situation we are not able to catch some candidates. Thus, we define two candidate sets, one for the ideal case and the other for the nonideal case. Then, both of these candidate sets are merged before selection.

### 3.2.1 Candidate Set-1

In order to obtain the candidates of the first set, we use the white connected components. Each connected component of this representation acts as a gland candidate. The logic behind directly using the connected components is to catch the glands for the ideal case, in other words, the glands that are successfully separated by nuclear polygons. In Figure 3.3, the connected components which we want to catch are specified with black circles. Note that one row of Voronoi polygons is added to each white connected component since this component only represents the luminal part of a gland but not its epithelial cell nuclei.

### 3.2.2 Candidate Set-2

The purpose of defining the second set is to obtain the possible gland candidates which are not able to be caught in the first set. For some images especially containing pixel level noise and variations, the connected components generated from white polygons for the first candidate set can output one large component for multiple glands. By defining a rule, we aim to separate these large components to their corresponding glands. These are the candidates for the nonideal case. The green connected component in Figure 3.3 is an example for this case.

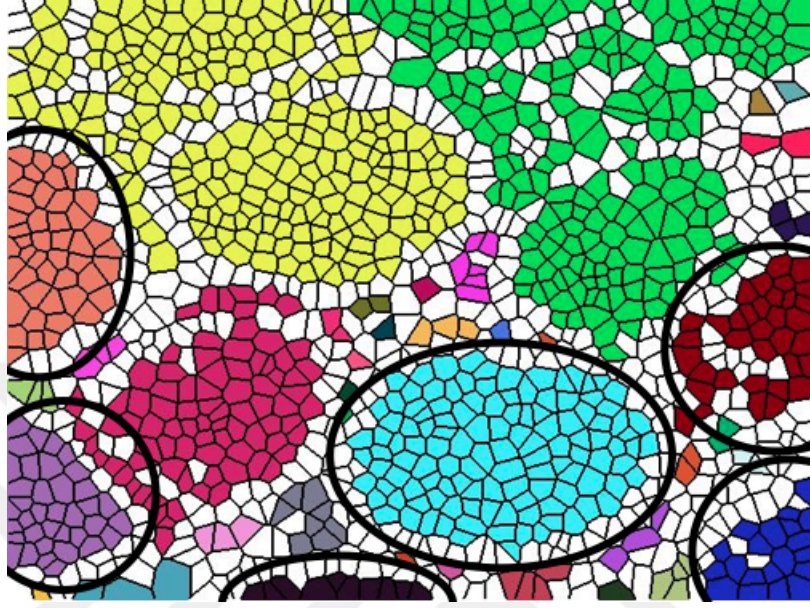


Figure 3.3: White connected components which are used in the generation of Candidate Set-1.

In a tissue image, luminal regions appear as white, as well as some stromal part of the region. Thus, some white regions can exist in between the glands. In order to differentiate white Voronoi polygons that correspond to white regions inside a gland and those in between the multiple glands, white Voronoi polygons are converted to nuclear which are located between nuclear Voronoi polygons. By doing so, we aim to obtain a closed region for the lumen that belongs to a single gland. In Figure 3.4, the places which should be converted are marked with red circles. Afterward, the white connected components are generated on these converted polygons. These white connected components' one row of Voronoi polygons added versions constitute our second candidate set.

### 3.2.2.1 Conversion Method for Nuclear Voronoi Polygons

In order to obtain the second candidate set, the white Voronoi polygons which are located between two or more nuclear Voronoi polygons, are converted to nuclear polygons. The reason behind this conversion is that some of the epithelial cell nuclei contain gaps between them. However, we expect that the nuclei belonging



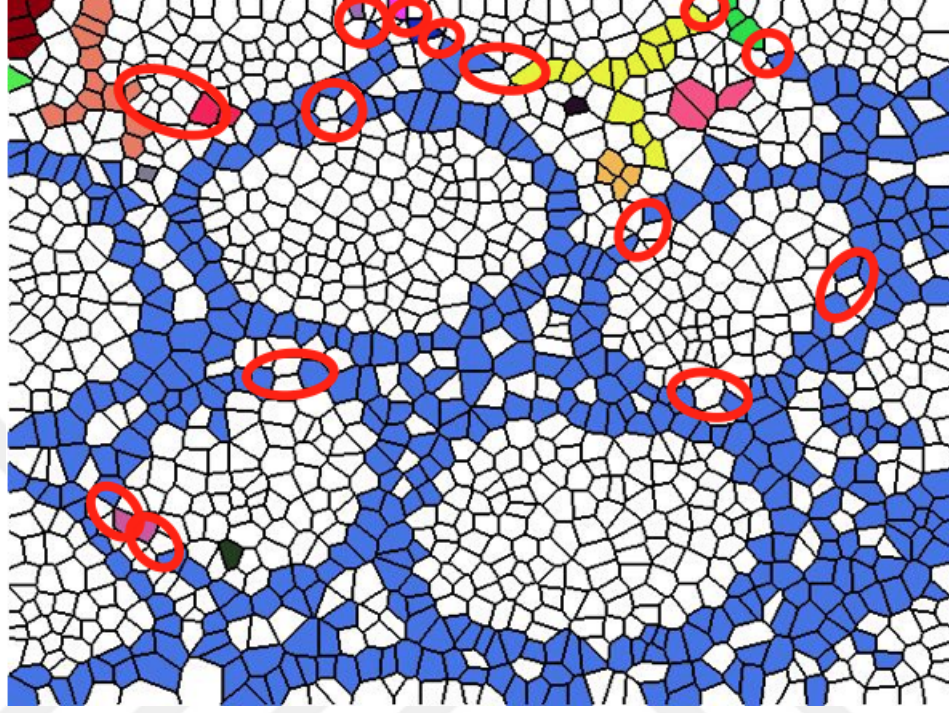


Figure 3.4: This representation shows the connected components which are generated from nuclear Voronoi polygons, for the given image. Red circles indicate the gaps between nuclear polygons, which should be converted.

to an individual gland should be adjacent to each other. Thus, by converting white Voronoi polygons, we aim to fill the gaps which are located on the boundary and obtain a path that includes all nuclear Voronoi polygons of a gland. The important point is that if we converted all of such white Voronoi polygons to nuclear ones, the nuclear connected components would start to close inward of the gland.

In order to minimize this inward closing and prevent unnecessary conversions, we define the following rule: For a white Voronoi polygon, we take its nuclear neighbors which are one step away from this white Voronoi polygon and look at the neighborhood relationships among these neighbors. If all these nuclear neighbors are reachable from each other using only themselves (i.e., if there is a single path among these nuclear neighbors), we do not change the type of this white Voronoi polygon. However, if there is not a path among any of two, the type of the white Voronoi polygon is changed to nuclear in order to obtain a path

among all nuclear Voronoi polygons.

During the conversion process, there is an exceptional case for the white Voronoi polygons which are located on the border of the image. If a white Voronoi polygon, which is on the border, has only one nuclear neighbor that is not on the border of the image, then this white Voronoi polygon is also converted to nuclear. This conversion enables to catch the glands on these locations, by getting closed structures between the nuclear Voronoi polygons and the border of the image.

Figure 3.5 demonstrates two different examples to clarify the conversion method. Figure 3.5 (a) shows the nuclear Voronoi polygons of the given image. In this figure, the parts specified with black circles are examined in detail in Figures 3.5 (b) and 3.5 (c). In order to make them more noticeable, all nuclear polygons are shown with blue in Figure 3.5 (b) and the converted polygons are indicated with red in Figure 3.5 (c).

In Figure 3.5 (b), we focus on four different polygons (numbered from 1 to 4) on each selected part. It is seen that Polygon 1 is not converted to nuclear in Figure 3.5 (c) since there is a path between its nuclear neighbors. However, if we separately look at the neighborhood relationships of Polygon 2's, Polygon 3's, and Polygon 4's one step away nuclear Voronoi polygons, it is seen that there is not a single path among them. Therefore, these white Voronoi polygons are converted to nuclear, in order to obtain a continuous path among the nuclear Voronoi polygons on the same white polygon.

After applying the conversion method to the white Voronoi polygons, new white connected components are found. Similar to the first set, one row of Voronoi polygons is added to each of these connected components and they compose of the second candidate set. Figure 3.6 shows the generation process of the candidate sets step by step.

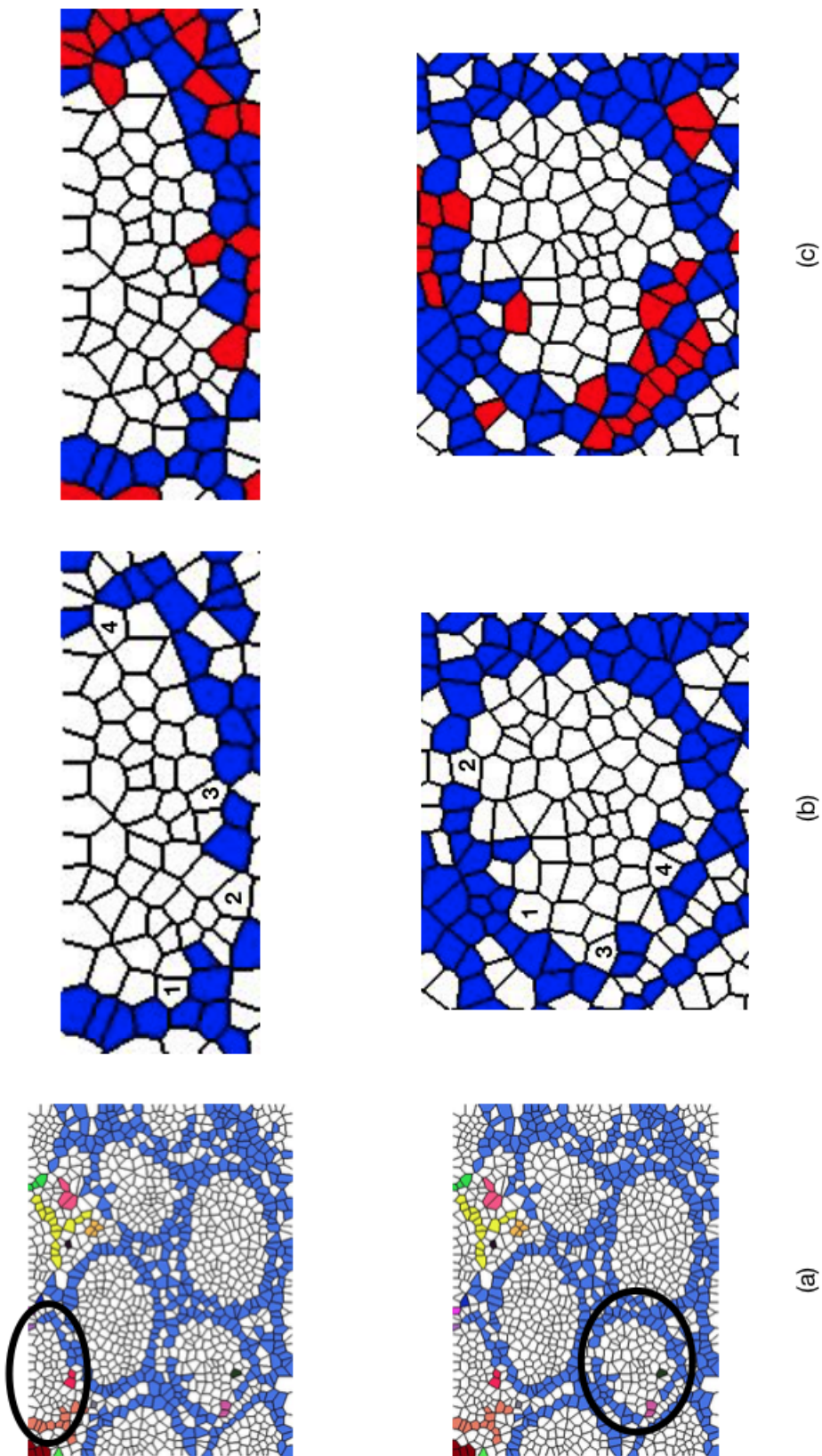


Figure 3.5: Illustration for the conversion method. (a) Voronoi polygons of the nuclear type, (b) before conversion, and (c) after conversion. In (b) and (c), nuclear polygons are shown with blue. The white polygons that do not change their types in this conversion are shown with white whereas those that change their types are shown with red. Here the numbers in the polygons are used in the text for explaining the steps of this conversion.



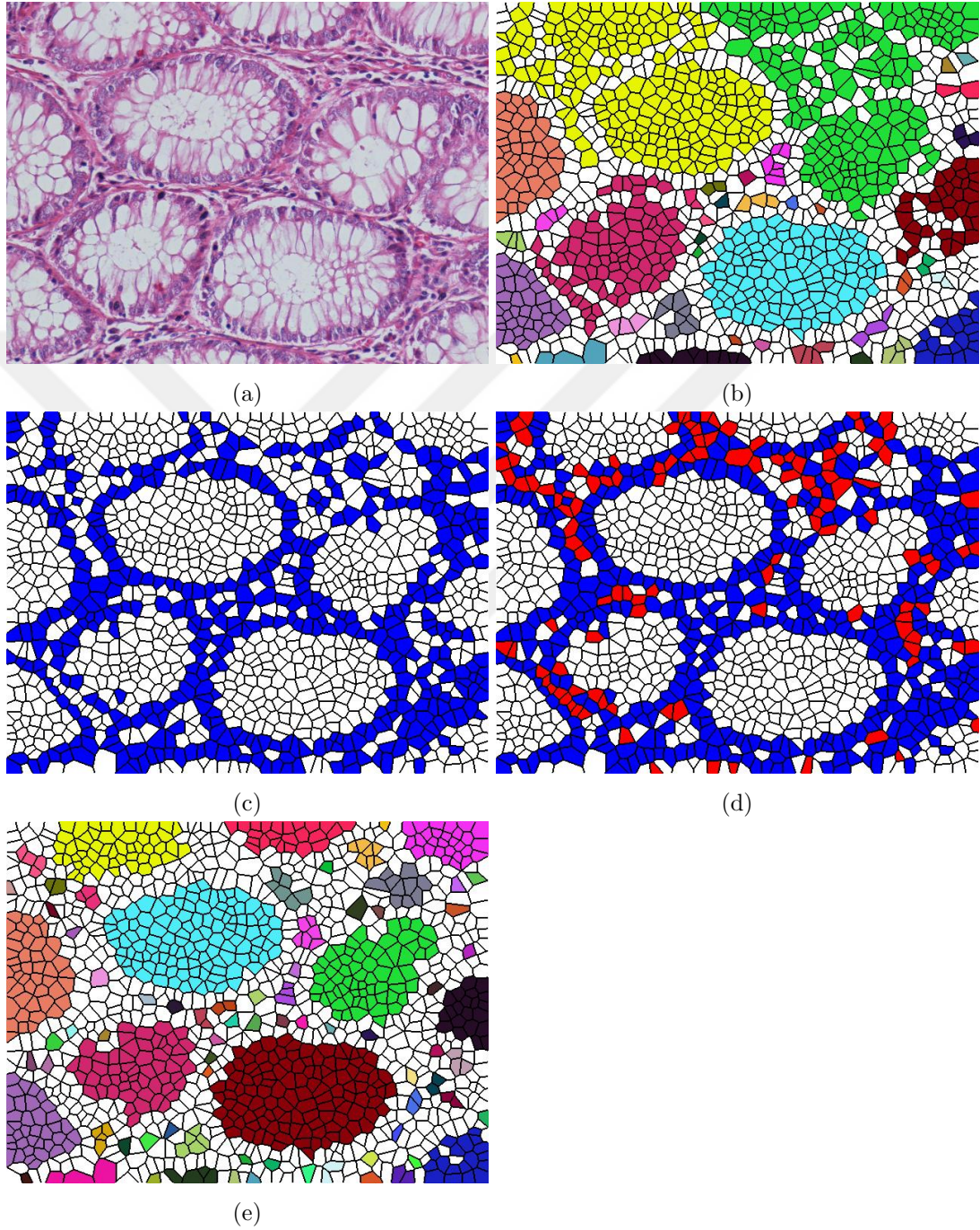


Figure 3.6: Steps of the candidate set generation: (a) Histopathological image, (b) white connected components, (c) nuclear Voronoi polygons before the conversion method, (d) nuclear Voronoi polygons after the conversion method, and (e) white connected components after the conversion operation.

### 3.3 Iterative Gland Localization

After defining the first and second candidate sets, all candidates are gathered in a single set and the gland selection process is started. In order to differentiate true gland candidates from the false ones, a metric is defined which is called *GlandScore*. For each candidate, the *GlandScore* is calculated. Our algorithm selects the candidates iteratively. For each iteration, a current threshold  $curr_{thr}$  is used, and the algorithm considers only the candidates whose *GlandScores* are larger than this threshold. Among these candidates, the one whose area is the biggest is selected and affected candidates' *GlandScores* are updated. When there exists no candidate whose *GlandScore* is greater than  $curr_{thr}$ , this threshold is decreased by 0.05 and the same process is repeated. In our method, we start our iterations with  $curr_{thr}=1$  and continues until it becomes less than the predefined threshold  $G_{thr}$ .

#### 3.3.1 Metric Definition

A gland is characterized by a large luminal area surrounded by epithelial cell nuclei. In our representation, the circular white luminal area is represented as a large enough connected component of white Voronoi polygons and epithelial cell nuclei are represented with the connected components of nuclear Voronoi polygons. Thus, a gland in our representation should have an elliptical convex shape. In addition to its shape, the nuclear Voronoi polygons' distribution is also important. According to our definition, a gland should have enough nuclear Voronoi polygons on its boundary, whereas it does not have much inside.

To mimic the structure of a gland, we define the *GlandScore* metric. This *GlandScore* metric is a product of two separate metrics, which are the *ConvexRatio* and the *Fscore*. The first one is the convex ratio which is defined for quantifying the elliptical convex shape. In order to get the *ConvexRatio*, we use the area and the convex area of a candidate. The second one is the *Fscore*, which quantifies the nuclear polygons on the boundary and inside of the candidate. The

logic behind defining the *Fscore* is that the nuclear Voronoi polygons' ratio both on the boundary and inside of a candidate will give us a value that quantifies the likelihood of that candidate being a gland.

The structure of a gland consists of white Voronoi polygons which are encircled with nuclear Voronoi polygons. In order to catch the defined structure and approximate the candidates' closeness to glands, one-row Voronoi polygons is added to white connected components' boundaries in the candidate generation process. Furthermore, while adding one-row Voronoi polygons, the candidates that contain other candidates inside are eliminated. These correspond to the candidates containing big gaps that another candidate can fit into. However, our glands do not consist of hollow structures, and therefore, those false candidates are directly eliminated at this step. Figure 3.7 shows two different examples for the eliminated candidates.

After obtaining the gland candidates, their *GlandScores* are calculated according to the following formulas. For a candidate  $C$ , let  $X(C)$  be the number of nuclear Voronoi polygons on  $C$ 's boundary,  $Y(C)$  be the total number of nuclear Voronoi polygons belonging to  $C$ , and  $Z(C)$  be the total number of Voronoi polygons on  $C$ 's boundary (nuclear as well as white ones). Then, for this candidate  $C$ ,

$$\text{Precision}(C) = \frac{X(C)}{Y(C)} \quad (3.1)$$

$$\text{Recall}(C) = \frac{X(C)}{Z(C)} \quad (3.2)$$

$$\text{ConvexRatio}(C) = \frac{\text{Area}(C)}{\text{ConvexArea}(C)} \quad (3.3)$$

$$\text{Fscore}(C) = \frac{2 \times \text{Precision}(C) \times \text{Recall}(C)}{\text{Precision}(C) + \text{Recall}(C)} \quad (3.4)$$

$$\text{GlandScore}(C) = \text{ConvexRatio}(C) \times \text{Fscore}(C) \quad (3.5)$$

Note that, when we add one-row Voronoi polygons, the original labels of these polygons are used in the calculation of the *GlandScore*. This means if a white Voronoi polygon is converted to nuclear during the conversion process, the *GlandScore* is calculated by considering the original white label of this polygon.

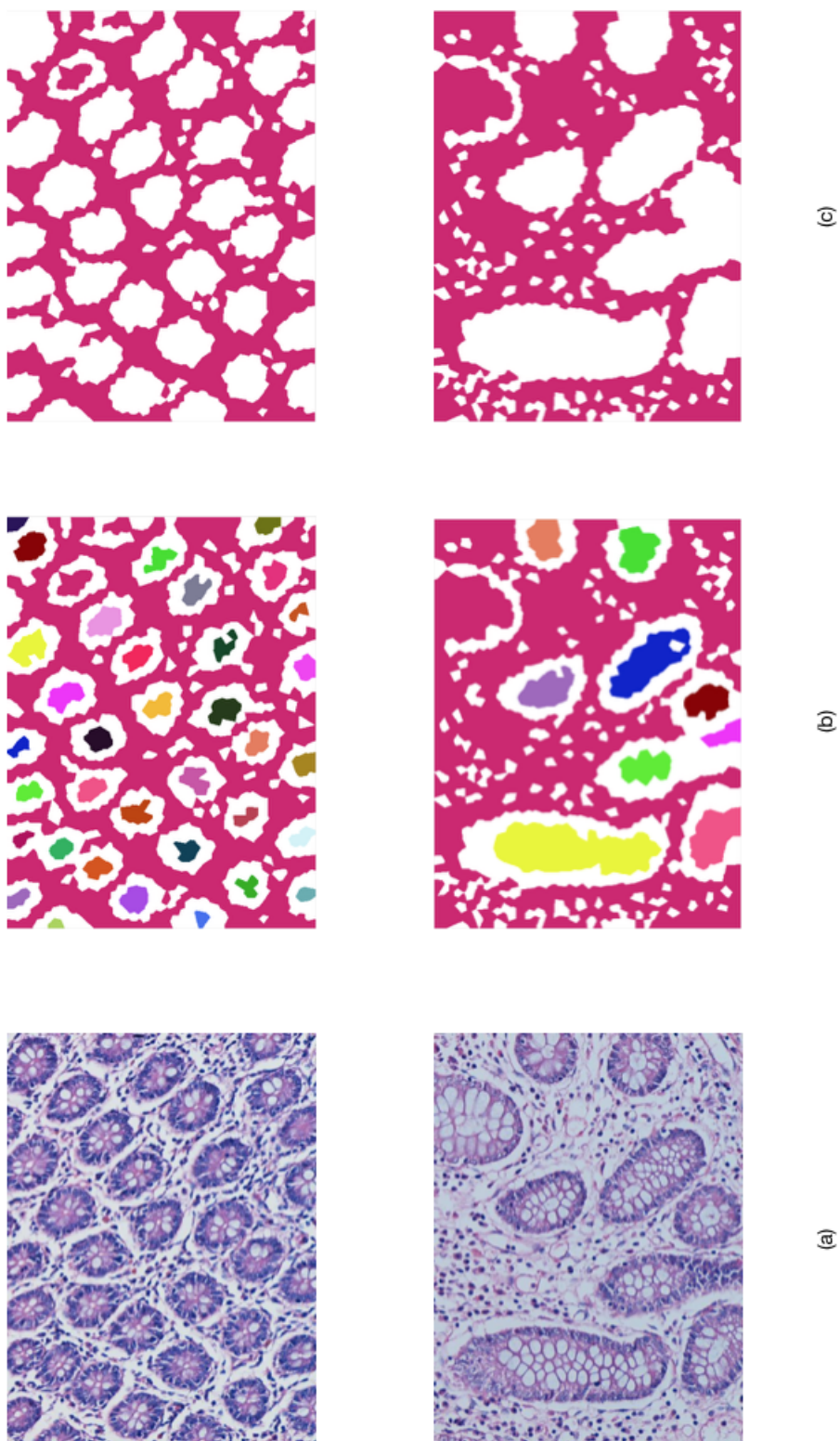


Figure 3.7: Examples for the eliminated candidates: (a) Histopathological images, (b) candidates of the first set for the given images, (c) eliminated candidates which contain big gaps that another candidate can fit into it.



There is an important point that needs to be drawn attention. While adding one-row Voronoi polygons to each candidate’s boundary, some of these Voronoi polygons can be used by more than one candidate. Each candidate’s *GlandScore* is calculated considering these shared Voronoi polygons. However, in the selection process, the one that is first selected takes the shared Voronoi polygons and these polygons are removed from the other candidates’ structures. As a result, the affected candidates’ *GlandScores* are recalculated according to the remaining Voronoi polygons. Figure 3.8 shows three candidates that share Voronoi polygons. The Voronoi polygons shared by Candidate-1 and Candidate-2 are marked with blue circles, whereas the polygons shared by Candidate-2 and Candidate-3 are marked with orange circles. In the *GlandScore* calculation process, each candidate uses these Voronoi polygons as if they were their own. These candidates’ calculated *GlandScores* are indicated under the pictures in Figure 3.8. Among the three of them, Candidate-1 is to be first selected. When it is selected, the Voronoi polygons which are specified with the blue circle are eliminated from Candidate-2 and its *GlandScore* is recalculated (it drops from 0.72 to 0.61). Then when Candidate-3 is selected, this time Candidate-2 loses the polygons that are indicated with the orange circle and its *GlandScore* drops to 0.50.

### 3.3.2 Iterative Gland Selection Algorithm

This algorithm uses an iterative approach to localize the glands based on their *GlandScore*. After generating candidate sets and calculating each candidate’s *GlandScore*, our algorithm starts with selecting the “best” candidate according to the *GlandScore* metric. At each iteration, the affected candidates’ *GlandScores* are updated. The pseudocode of this procedure is given in Algorithm 1.

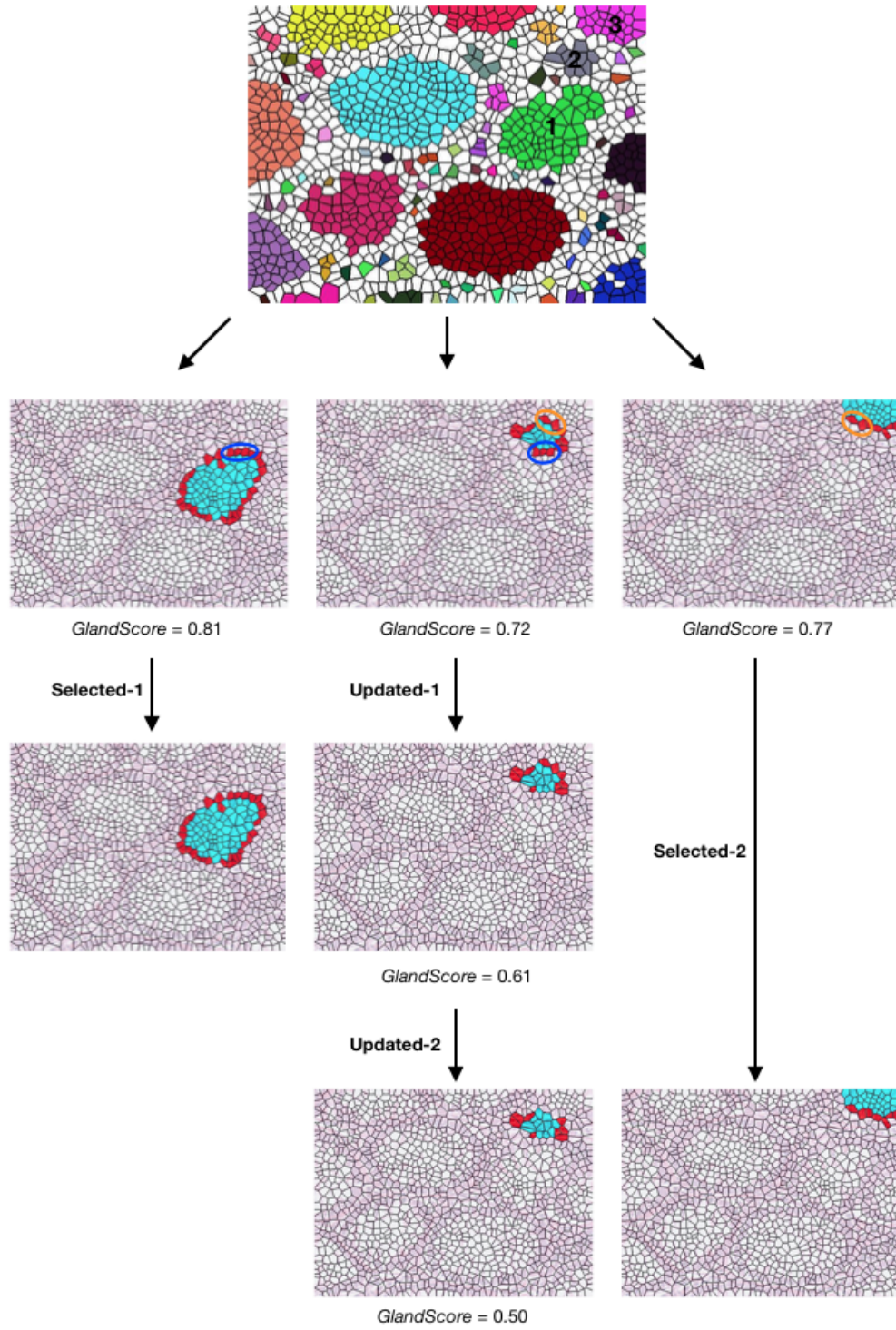


Figure 3.8: Candidate examples that share Voronoi polygons.

---

**Algorithm 1** Iterative Gland Selection Algorithm

---

**Input:** candidates  $\mathcal{C}$ , area threshold  $\mathcal{A}_{thr}$ , global *GlandScore* threshold  $G_{thr}$

**Output:** detected glands  $\mathcal{D}$

```
1:  $\mathcal{D} = \emptyset$ 
2:  $curr_{thr} \leftarrow 1$ 
   //  $curr_{thr}$  defines a local threshold for each iteration
3: while  $curr_{thr} \geq G_{thr}$  do
4:   while  $\mathcal{C}_{subset} = \{c_i \in \mathcal{C}_{subset} | GlandScore(c_i) \geq curr_{thr}$ 
5:     and  $area(c_i) \geq \mathcal{A}_{thr}\} \neq \emptyset$  do
6:     find  $C^* \in \mathcal{C}_{subset}$  whose area is largest
7:      $\mathcal{D} = \mathcal{D} \cup C^*$ 
8:     update  $\forall c_i \in \mathcal{C}$ 
9:   end while
10:   $curr_{thr} \leftarrow curr_{thr} - 0.05$ 
11: end while
```

---

Before selection, the candidates whose areas are below a predefined area threshold  $\mathcal{A}_{thr}$  are eliminated. The reason behind this elimination is that, after adding one-row Voronoi polygons to each white connected component, the *GlandScores* of some false candidates whose areas are very small become very high. These candidates especially belong to the second set, as a result of the conversions. This elimination prevents our algorithm to be misled by these small white connected components. Figure 3.9 shows some eliminated candidates which have been obtained from Candidate Set-2.

In our method, we select  $\mathcal{A}_{thr}$  to be proportional to the largest area among the candidates whose *GlandScores* are higher than the global *GlandScore* threshold  $G_{thr}$ . The reason of this is the following: Due to the different sizes of each histopathological image's glands, defining a fixed  $\mathcal{A}_{thr}$  is not feasible. Thus, at the beginning of the algorithm, we first find the largest candidate whose *GlandScore* is higher than the global threshold  $G_{thr}$  and set  $\mathcal{A}_{thr}$  as the  $p$  percent of its area.

Then, we start the iterative procedure by setting  $curr_{thr} = 1$  and considers only

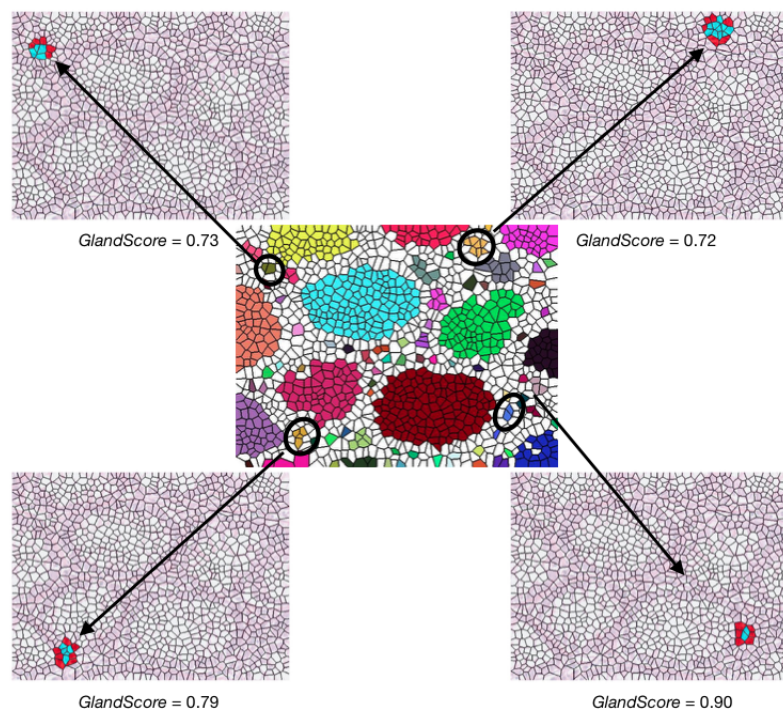


Figure 3.9: These are some of the examples which are eliminated through area thresholding. It is seen that their *GlandScores* are high enough to be considered when one-row of nuclear polygons is added.



the candidates whose *GlandScores* are greater than or equal to this  $curr_{thr}$ . (Here note that  $G_{thr}$  is the global threshold until which the selection continues. On the other hand,  $curr_{thr}$  is a local threshold whose value changes among different iterations.) Among these candidates, our algorithm selects the one whose area is the biggest. This step iteratively continues until all candidates whose *GlandScores* are greater than or equal to  $curr_{thr}$  are selected. Then, the local threshold  $curr_{thr}$  is decreased by 5 percent. Again, only the candidates whose *GlandScores* are greater than or equal to the newly set  $curr_{thr}$  are considered, and the selection is done similarly. The selection process stops when the value of  $curr_{thr}$  is less than the global threshold  $G_{thr}$ . The results of our iterative algorithm are illustrated on an example image in Figure 3.10.

After each selection, the selected candidate's Voronoi polygons are eliminated from the others'. Those affected candidates' *GlandScores* and areas are recalculated and updated before the next iteration. Due to these changes, the ones whose areas less than  $\mathcal{A}_{thr}$  are also eliminated and not considered in the next iterations. The reason behind using multiple thresholds and decreasing  $curr_{thr}$  by 5 percent at each iteration is to give priority to the candidates whose *GlandScores* are greater than a value, and among those, to select the biggest first. If we just considered the area as a criterion and started selecting the one whose area is the largest, candidates whose *GlandScores* are very high would not be selected at the beginning of the algorithm, because they are small in size even though their *GlandScores* are high.

When the algorithm stops and returns the detected glands, in order to smooth the shapes of these glands a post-processing step is applied. In this post-processing step, for a gland, its boundary pixels are taken, and from each of the selected pixels, a line is drawn to the  $n^{th}$  pixel which comes after itself and the pixels in between the lines of all pixels are included in the gland's region.

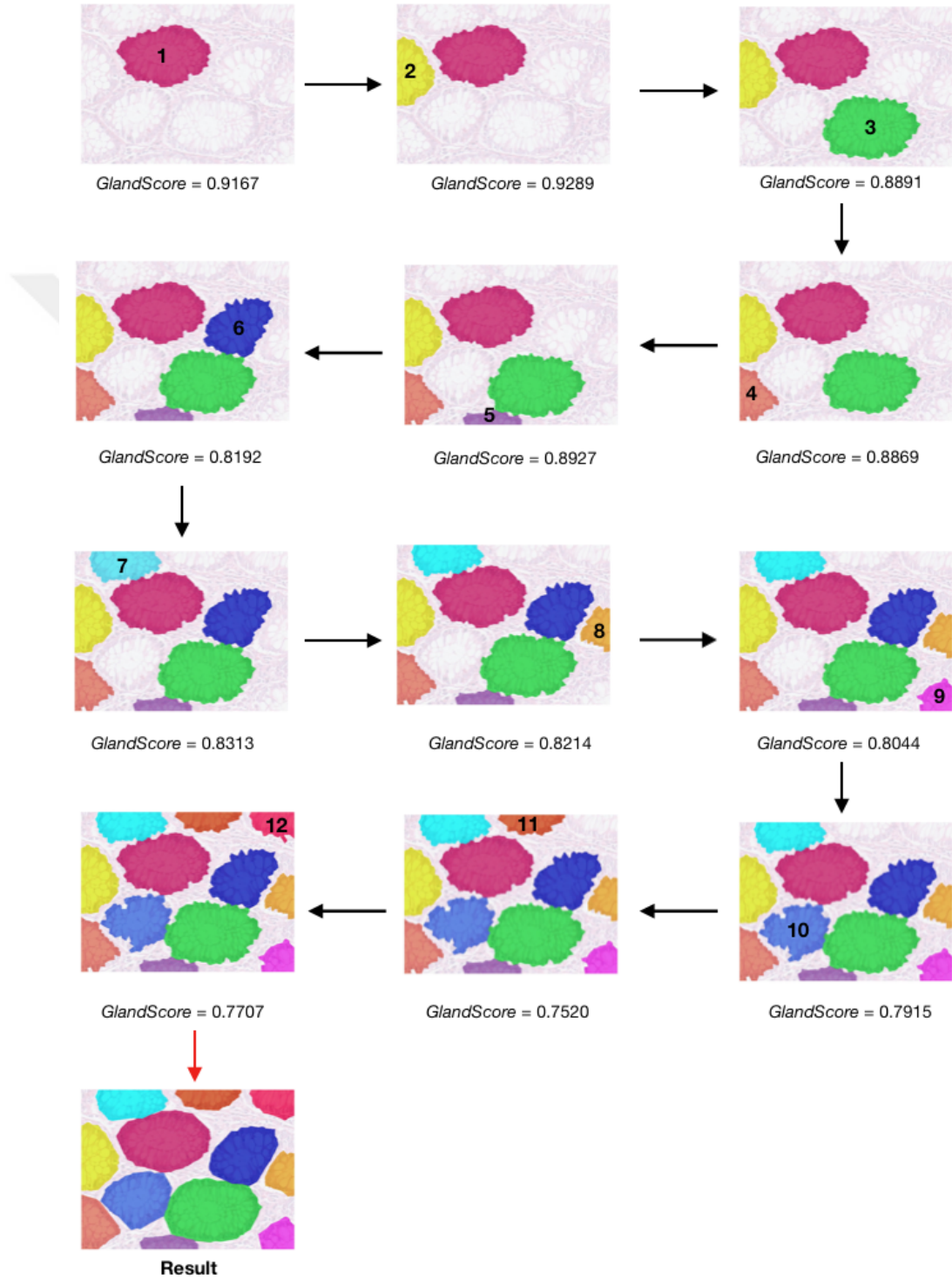


Figure 3.10: The glands selected by the iterative algorithm and their selection order.

# Chapter 4

## Experiments

In this chapter, firstly we give the details of the dataset that is used for training and testing our algorithm. Secondly, our evaluation technique is described in a detailed way. After the descriptions about the dataset and the evaluation technique, the selected parameters and their ranges are discussed. Finally, the results of our experiments and the results of the existing methods are presented with a discussion about the comparisons between them.

### 4.1 Dataset

We conduct our experiments on the dataset which contains 72 microscopic images of colon tissues taken from 36 patients. These images are taken from the Pathology Department Archives of Hacettepe University School of Medicine. They are acquired by a Nikon Coolscope Digital Microscope, by using a  $20\times$  microscope objective lens and their resolution is  $480 \times 640$ . The first 24 images which are taken from randomly selected 12 patients, constitute the training set. The remaining 48 images that are taken from the remaining 24 patients compose of the test set. All the images in our dataset are stained with the hematoxylin-and-eosin technique.

## 4.2 Evaluation

We evaluate our segmentation results by using the gold standards of the given images. In the gold standard, foregrounds and backgrounds are annotated. This annotation was performed by our pathologist collaborator.

To quantitatively measure the success of our segmentation method, true positive, false positive, true negative, and false negative values are calculated at the pixel level. Let  $A$  be the gland pixels in the gold standard and  $C$  be the gland pixels identified by the algorithm. Then, a pixel  $P$  is considered as true positive if  $P \in A$  and  $P \in C$ . However, if  $P \in A$  but  $P \notin C$ , this pixel is considered as false negative. Likewise,  $P$  is considered as false positive if  $P \notin A$  but  $P \in C$  and true negative if  $P \notin A$  and  $P \notin C$ .

Then accuracy, dice index, sensitivity, and specificity evaluation metrics are calculated by using these four values. In particular, they are computed as:

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.1)$$

$$\text{dice index} = \frac{2 \times TP}{2 \times TP + FP + FN} \quad (4.2)$$

$$\text{sensitivity} = \frac{TP}{TP + FN} \quad (4.3)$$

$$\text{specificity} = \frac{TN}{TN + FP} \quad (4.4)$$

## 4.3 Parameter Selection

Our rule based segmentation algorithm has the following four external parameters:

- $r_{min}$  is the minimum radius for the circles that are located on the given pixel groups in the Voronoi diagram generation process. After separating pixels

as black and white, circles are located on each of these groups starting from the largest circle to the one whose radius is at least  $r_{min}$ . Then, the centers of these circles are taken and a Voronoi diagram is located onto them.

- $G_{thr}$  is the global *GlandScore* threshold that we use for the selection of candidates. Our iterative algorithm starts from  $curr_{thr} = 1$ . Then,  $curr_{thr}$  is decreased by 5 percent at each iteration, until none of the candidates' *GlandScores* are greater than or equal to  $G_{thr}$ .
- $\mathcal{A}_{thr}$  is the area threshold for the elimination of candidates. Candidates whose area is smaller than this threshold are eliminated. Since an image may consist of glands of different sizes, defining a fixed  $\mathcal{A}_{thr}$  for all images may not be possible. Therefore, for each image, the candidate which has the largest area among the candidates whose *glandScore* is greater than  $G_{thr}$  is found.  $\mathcal{A}_{thr}$  is equal to the  $p$  percent of the area of this largest candidate.
- $n$  is the parameter that is used for smoothing the selected glands. At the end of our iterative algorithm, a post-processing step is applied to the selected glands. In this process, the pixels on the boundary are taken for each gland, and a line is drawn from each pixel to the  $n^{th}$  pixel which comes after it. At the end, the regions confined with these lines are included to the gland's region.

These parameteres are obtained using the images of the training set. The ranges of our parameters are as follow:  $r_{min} = \{3, 4, 5\}$ ,  $G_{thr} = \{0.5, 0.6, 0.7, 0.8\}$ ,  $p = \{0.10, 0.15, 0.20\}$ , and  $n = \{40, 60, 80\}$ . Considering all possible combinations of these parameters, we select the combination that gives the highest accuracy on the training set.

## 4.4 Results

In Figures 4.1 and 4.2, the segmentation results on example images are visually presented. In these figures, the first columns contain original images, the second

columns correspond to their gold standards, and the third columns contain our segmentation results. The images in Figure 4.1 belong to the training set and the ones in Figure 4.2 belong to the test set.

The quantitative results obtained on the training set images are reported in Tables 4.1, 4.2, and 4.3. These tables include the results of all possible combinations of the considered parameters. The average performances and their standard deviations are reported and the best results based on accuracy are indicated in bold.

We compare the segmentation results of our algorithm with three different methods. The first two of them are pixel-based methods and the last one is a structural method. Wu *et al.* present two different pixel-based methods for gland segmentation. In the first study, a region growing algorithm is presented for detecting the glands [4]. In order to find the luminal parts, a window of radius  $R_0$  is moved on the binary image, which is obtained by thresholding, and the white areas which completely cover this window are chosen as potential seeds. Then, the seeds are expanded by applying an iterative morphological dilation operation with a circular structuring element. In the second study, a directional filtering based method is presented [24]. They design directional two-dimensional linear filters with different orientations, in order to lower the pixel intensities. For each pixel, the lowest output value of these filters is selected. Then, epithelial cell nuclei are identified by using an intensity threshold. After finding the nucleus pixels the gaps which are surrounded by these pixels are filled with a circular structuring element. The first study approaches gland detection as a lumen-identification problem; on the other hand, the second study approaches it as a nucleus-identification problem.

Both of these methods were previously implemented in our research group. Our proposed algorithm's and the pixel-based methods' average quantitative results and their standard deviations are reported in Tables 4.4 and 4.5 for the training and test sets, respectively. According to the results, it is seen that our proposed algorithm improves the segmentation results of these pixel-based algorithms.



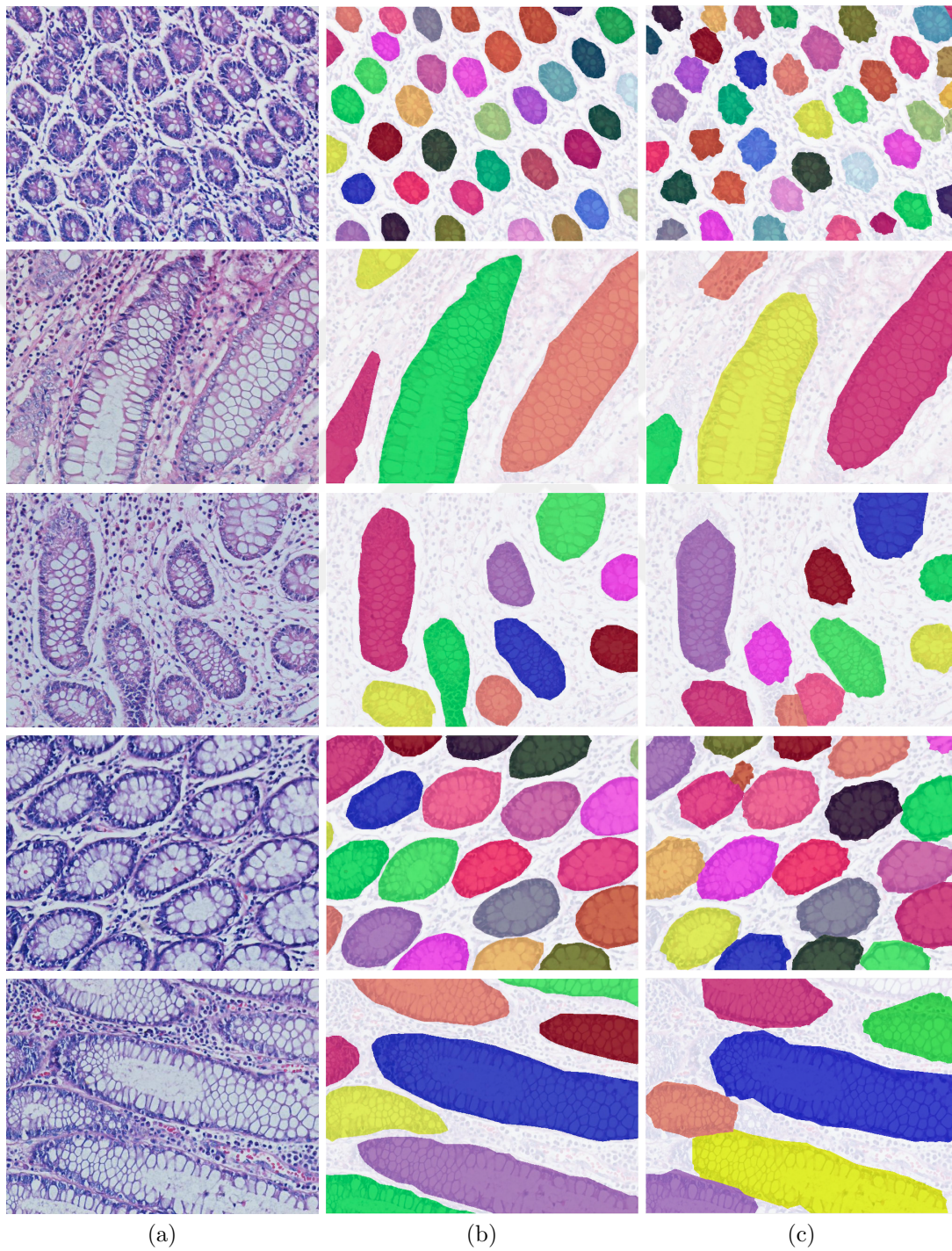


Figure 4.1: Example results from the training set: (a) original images, (b) gold standards of these images, and (c) our segmentation results.



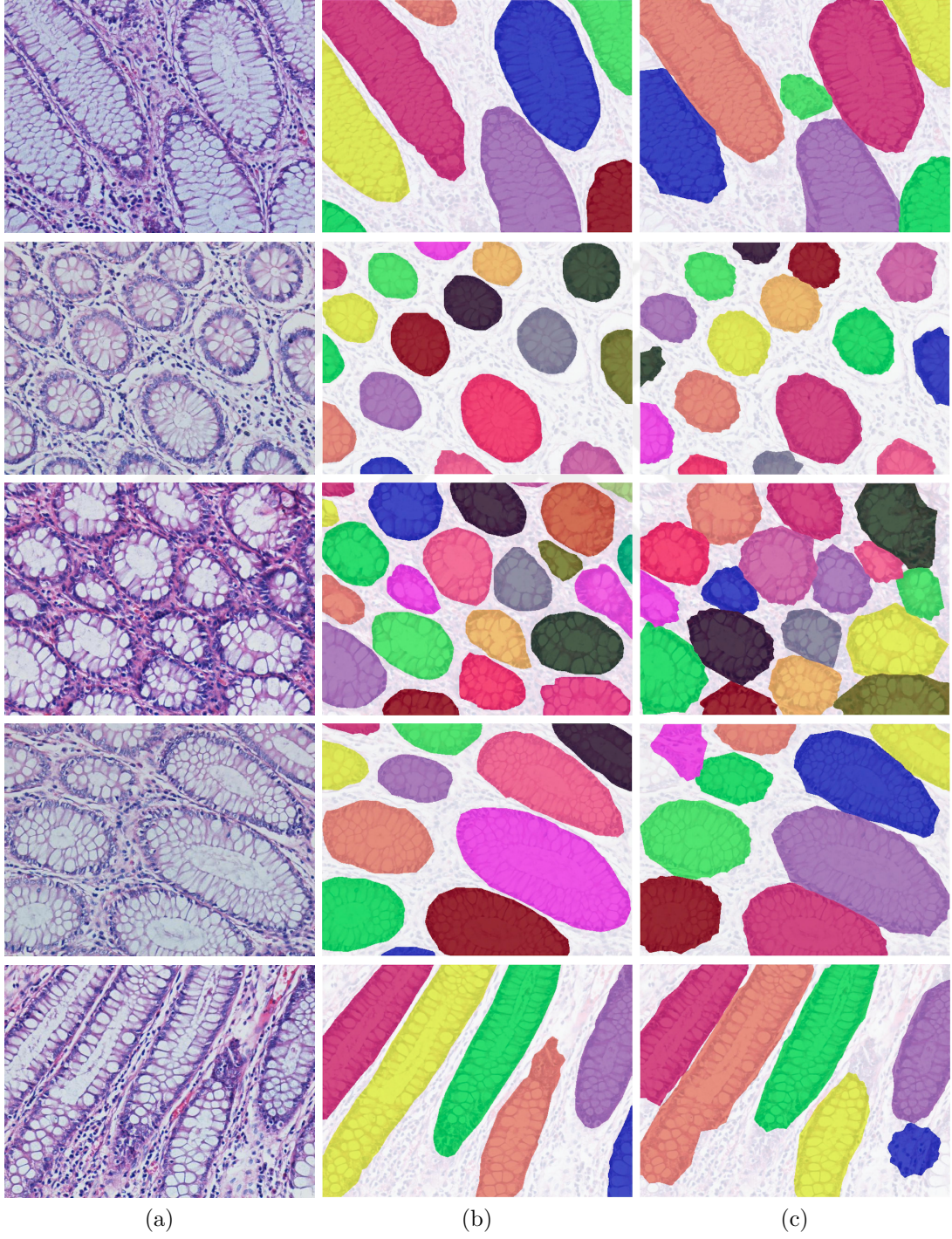


Figure 4.2: Example results from the test set: (a) original images, (b) gold standards of these images, and (c) our segmentation results.



Table 4.1: Quantative results obtained on the training images when n=40. Average results and their standard deviations are reported.

| n = 40    |      |             | $G_{thr}$         |                   |                   |                   |
|-----------|------|-------------|-------------------|-------------------|-------------------|-------------------|
| $r_{min}$ | $p$  |             | 0.5               | 0.6               | 0.7               | 0.8               |
| 3         | 0.10 | accuracy    | 76.14 $\pm$ 12.20 | 78.74 $\pm$ 13.09 | 77.87 $\pm$ 16.57 | 63.57 $\pm$ 20.08 |
|           |      | dice index  | 76.88 $\pm$ 16.31 | 78.49 $\pm$ 16.68 | 76.44 $\pm$ 19.68 | 49.60 $\pm$ 31.90 |
|           |      | sensitivity | 77.11 $\pm$ 20.02 | 74.93 $\pm$ 20.04 | 69.68 $\pm$ 23.48 | 40.49 $\pm$ 31.19 |
|           |      | specificity | 77.01 $\pm$ 10.59 | 85.22 $\pm$ 6.87  | 90.35 $\pm$ 6.85  | 95.81 $\pm$ 3.95  |
|           | 0.15 | accuracy    | 79.24 $\pm$ 8.31  | 81.03 $\pm$ 9.82  | 79.11 $\pm$ 14.65 | 63.31 $\pm$ 20.23 |
|           |      | dice index  | 80.89 $\pm$ 8.95  | 81.83 $\pm$ 9.78  | 78.23 $\pm$ 15.93 | 48.44 $\pm$ 32.71 |
|           |      | sensitivity | 80.04 $\pm$ 12.09 | 77.34 $\pm$ 13.92 | 70.37 $\pm$ 20.44 | 39.33 $\pm$ 31.27 |
|           |      | specificity | 79.22 $\pm$ 9.32  | 86.64 $\pm$ 7.08  | 92.27 $\pm$ 5.58  | 96.80 $\pm$ 2.89  |
|           | 0.20 | accuracy    | 79.66 $\pm$ 8.44  | 81.22 $\pm$ 9.94  | 79.07 $\pm$ 14.47 | 62.79 $\pm$ 20.62 |
|           |      | dice index  | 80.58 $\pm$ 9.23  | 81.62 $\pm$ 10.07 | 78.11 $\pm$ 15.83 | 47.17 $\pm$ 33.71 |
|           |      | sensitivity | 77.37 $\pm$ 13.04 | 75.41 $\pm$ 14.08 | 69.69 $\pm$ 20.15 | 38.30 $\pm$ 31.57 |
|           |      | specificity | 84.44 $\pm$ 8.59  | 89.61 $\pm$ 6.94  | 93.40 $\pm$ 4.83  | 97.24 $\pm$ 2.76  |
| 4         | 0.10 | accuracy    | 83.19 $\pm$ 7.82  | 85.37 $\pm$ 7.10  | 85.48 $\pm$ 9.86  | 78.34 $\pm$ 15.49 |
|           |      | dice index  | 85.23 $\pm$ 9.04  | 86.85 $\pm$ 8.28  | 86.75 $\pm$ 9.74  | 77.68 $\pm$ 19.48 |
|           |      | sensitivity | 89.05 $\pm$ 11.09 | 88.49 $\pm$ 11.00 | 85.25 $\pm$ 14.26 | 71.32 $\pm$ 22.98 |
|           |      | specificity | 73.91 $\pm$ 11.76 | 79.88 $\pm$ 7.83  | 85.13 $\pm$ 7.37  | 90.45 $\pm$ 5.88  |
|           | 0.15 | accuracy    | 83.43 $\pm$ 7.87  | 85.39 $\pm$ 7.50  | 85.38 $\pm$ 9.92  | 77.22 $\pm$ 16.10 |
|           |      | dice index  | 84.89 $\pm$ 9.97  | 86.51 $\pm$ 9.26  | 86.28 $\pm$ 10.71 | 75.86 $\pm$ 21.53 |
|           |      | sensitivity | 86.89 $\pm$ 12.38 | 86.22 $\pm$ 12.37 | 83.29 $\pm$ 14.92 | 68.78 $\pm$ 23.81 |
|           |      | specificity | 77.90 $\pm$ 9.89  | 82.58 $\pm$ 10.02 | 87.22 $\pm$ 8.58  | 91.54 $\pm$ 5.26  |
|           | 0.20 | accuracy    | 82.68 $\pm$ 9.94  | 83.99 $\pm$ 9.83  | 84.16 $\pm$ 11.45 | 76.68 $\pm$ 16.24 |
|           |      | dice index  | 83.84 $\pm$ 11.32 | 84.95 $\pm$ 10.89 | 84.89 $\pm$ 12.02 | 75.18 $\pm$ 21.74 |
|           |      | sensitivity | 83.60 $\pm$ 14.93 | 82.98 $\pm$ 14.80 | 80.76 $\pm$ 16.57 | 67.80 $\pm$ 23.88 |
|           |      | specificity | 81.97 $\pm$ 8.78  | 85.17 $\pm$ 8.34  | 88.86 $\pm$ 6.41  | 91.80 $\pm$ 5.10  |
| 5         | 0.10 | accuracy    | 72.83 $\pm$ 15.59 | 73.61 $\pm$ 16.20 | 73.96 $\pm$ 16.49 | 70.39 $\pm$ 17.57 |
|           |      | dice index  | 73.27 $\pm$ 18.53 | 73.77 $\pm$ 18.88 | 73.89 $\pm$ 19.30 | 68.58 $\pm$ 22.29 |
|           |      | sensitivity | 72.22 $\pm$ 25.97 | 71.28 $\pm$ 25.44 | 69.95 $\pm$ 24.91 | 62.05 $\pm$ 25.75 |
|           |      | specificity | 75.78 $\pm$ 11.77 | 78.88 $\pm$ 10.51 | 81.18 $\pm$ 10.38 | 85.30 $\pm$ 8.17  |
|           | 0.15 | accuracy    | 75.94 $\pm$ 13.31 | 76.51 $\pm$ 13.92 | 76.48 $\pm$ 14.07 | 70.61 $\pm$ 17.27 |
|           |      | dice index  | 77.03 $\pm$ 16.71 | 77.32 $\pm$ 16.94 | 77.06 $\pm$ 17.11 | 68.33 $\pm$ 22.94 |
|           |      | sensitivity | 77.43 $\pm$ 22.27 | 75.94 $\pm$ 21.70 | 73.84 $\pm$ 21.34 | 61.55 $\pm$ 25.75 |
|           |      | specificity | 73.51 $\pm$ 11.56 | 76.92 $\pm$ 11.55 | 80.12 $\pm$ 9.71  | 86.42 $\pm$ 7.84  |
|           | 0.20 | accuracy    | 74.96 $\pm$ 13.51 | 75.77 $\pm$ 13.92 | 75.52 $\pm$ 14.23 | 69.43 $\pm$ 17.24 |
|           |      | dice index  | 75.32 $\pm$ 18.14 | 75.74 $\pm$ 18.31 | 75.31 $\pm$ 18.34 | 66.62 $\pm$ 22.67 |
|           |      | sensitivity | 74.35 $\pm$ 23.02 | 72.91 $\pm$ 22.30 | 70.58 $\pm$ 21.70 | 58.92 $\pm$ 25.39 |
|           |      | specificity | 75.61 $\pm$ 11.62 | 79.31 $\pm$ 10.63 | 82.20 $\pm$ 9.33  | 87.52 $\pm$ 6.72  |

Table 4.2: Quantative results obtained on the training images when n=60. Average results and their standard deviations are reported.

| n = 60    |      |             | $G_{thr}$         |                   |                                     |                   |
|-----------|------|-------------|-------------------|-------------------|-------------------------------------|-------------------|
| $r_{min}$ | $p$  |             | 0.5               | 0.6               | 0.7                                 | 0.8               |
| 3         | 0.10 | accuracy    | 76.20 $\pm$ 12.22 | 79.09 $\pm$ 13.18 | 77.99 $\pm$ 16.64                   | 63.52 $\pm$ 20.09 |
|           |      | dice index  | 77.18 $\pm$ 16.31 | 78.69 $\pm$ 16.74 | 76.44 $\pm$ 19.73                   | 49.38 $\pm$ 31.86 |
|           |      | sensitivity | 76.40 $\pm$ 19.77 | 74.26 $\pm$ 19.85 | 69.06 $\pm$ 23.31                   | 39.98 $\pm$ 30.81 |
|           |      | specificity | 78.54 $\pm$ 9.77  | 86.38 $\pm$ 6.66  | 91.07 $\pm$ 7.03                    | 96.13 $\pm$ 3.99  |
|           | 0.15 | accuracy    | 79.68 $\pm$ 8.20  | 81.26 $\pm$ 9.83  | 79.22 $\pm$ 8.20                    | 63.25 $\pm$ 20.17 |
|           |      | dice index  | 81.15 $\pm$ 8.78  | 81.98 $\pm$ 9.74  | 78.29 $\pm$ 15.91                   | 48.27 $\pm$ 32.60 |
|           |      | sensitivity | 79.59 $\pm$ 11.78 | 76.91 $\pm$ 13.67 | 69.95 $\pm$ 20.19                   | 38.91 $\pm$ 30.87 |
|           |      | specificity | 80.31 $\pm$ 8.57  | 87.36 $\pm$ 7.09  | 92.77 $\pm$ 5.69                    | 97.03 $\pm$ 2.91  |
|           | 0.20 | accuracy    | 79.87 $\pm$ 8.35  | 81.32 $\pm$ 9.91  | 79.09 $\pm$ 14.42                   | 62.72 $\pm$ 20.55 |
|           |      | dice index  | 80.68 $\pm$ 9.18  | 81.67 $\pm$ 9.97  | 78.09 $\pm$ 15.76                   | 47.02 $\pm$ 33.59 |
|           |      | sensitivity | 77.07 $\pm$ 12.75 | 75.11 $\pm$ 13.84 | 69.39 $\pm$ 19.91                   | 37.98 $\pm$ 31.25 |
|           |      | specificity | 84.98 $\pm$ 7.84  | 89.99 $\pm$ 6.91  | 93.66 $\pm$ 4.90                    | 97.36 $\pm$ 2.84  |
| 4         | 0.10 | accuracy    | 83.35 $\pm$ 7.75  | 85.50 $\pm$ 7.04  | <b>85.55 <math>\pm</math> 9.80</b>  | 78.34 $\pm$ 15.46 |
|           |      | dice index  | 85.30 $\pm$ 8.99  | 86.92 $\pm$ 8.23  | <b>86.77 <math>\pm</math> 9.71</b>  | 77.63 $\pm$ 19.48 |
|           |      | sensitivity | 88.80 $\pm$ 11.02 | 88.26 $\pm$ 10.93 | <b>84.97 <math>\pm</math> 14.12</b> | 71.00 $\pm$ 22.79 |
|           |      | specificity | 74.47 $\pm$ 11.74 | 80.36 $\pm$ 7.86  | <b>85.52 <math>\pm</math> 7.45</b>  | 90.78 $\pm$ 5.84  |
|           | 0.15 | accuracy    | 85.53 $\pm$ 7.80  | 85.47 $\pm$ 7.43  | 85.43 $\pm$ 9.86                    | 77.23 $\pm$ 16.07 |
|           |      | dice index  | 84.94 $\pm$ 9.90  | 86.55 $\pm$ 9.19  | 86.29 $\pm$ 10.64                   | 75.82 $\pm$ 21.49 |
|           |      | sensitivity | 86.72 $\pm$ 12.27 | 86.04 $\pm$ 12.26 | 83.09 $\pm$ 14.78                   | 68.51 $\pm$ 23.62 |
|           |      | specificity | 78.25 $\pm$ 9.91  | 82.91 $\pm$ 10.01 | 87.50 $\pm$ 8.62                    | 91.82 $\pm$ 5.20  |
|           | 0.20 | accuracy    | 82.74 $\pm$ 9.85  | 84.05 $\pm$ 9.73  | 84.19 $\pm$ 11.39                   | 76.68 $\pm$ 16.20 |
|           |      | dice index  | 83.86 $\pm$ 11.22 | 84.97 $\pm$ 10.79 | 84.89 $\pm$ 11.94                   | 75.14 $\pm$ 21.69 |
|           |      | sensitivity | 83.41 $\pm$ 14.77 | 82.78 $\pm$ 14.64 | 80.55 $\pm$ 16.40                   | 67.55 $\pm$ 23.68 |
|           |      | specificity | 91.82 $\pm$ 5.20  | 85.46 $\pm$ 8.30  | 89.12 $\pm$ 6.45                    | 92.06 $\pm$ 5.05  |
| 5         | 0.10 | accuracy    | 72.87 $\pm$ 15.61 | 73.67 $\pm$ 16.27 | 74.02 $\pm$ 16.57                   | 70.46 $\pm$ 17.64 |
|           |      | dice index  | 73.28 $\pm$ 18.61 | 73.78 $\pm$ 18.97 | 73.92 $\pm$ 19.38                   | 68.63 $\pm$ 22.35 |
|           |      | sensitivity | 72.08 $\pm$ 25.89 | 71.15 $\pm$ 25.37 | 69.84 $\pm$ 24.84                   | 61.93 $\pm$ 25.64 |
|           |      | specificity | 75.91 $\pm$ 11.71 | 79.03 $\pm$ 10.42 | 81.35 $\pm$ 10.32                   | 85.49 $\pm$ 8.13  |
|           | 0.15 | accuracy    | 75.94 $\pm$ 13.34 | 76.52 $\pm$ 13.95 | 76.51 $\pm$ 14.13                   | 70.67 $\pm$ 17.34 |
|           |      | dice index  | 77.04 $\pm$ 16.72 | 77.33 $\pm$ 16.95 | 77.09 $\pm$ 17.16                   | 68.39 $\pm$ 22.98 |
|           |      | sensitivity | 77.33 $\pm$ 22.19 | 75.84 $\pm$ 21.62 | 73.76 $\pm$ 21.28                   | 61.47 $\pm$ 25.66 |
|           |      | specificity | 73.46 $\pm$ 11.77 | 76.89 $\pm$ 11.76 | 80.15 $\pm$ 9.85                    | 86.56 $\pm$ 7.81  |
|           | 0.20 | accuracy    | 74.97 $\pm$ 13.52 | 75.78 $\pm$ 13.52 | 75.55 $\pm$ 14.29                   | 69.49 $\pm$ 17.32 |
|           |      | dice index  | 75.33 $\pm$ 18.14 | 75.76 $\pm$ 18.31 | 75.35 $\pm$ 18.36                   | 66.67 $\pm$ 22.71 |
|           |      | sensitivity | 74.28 $\pm$ 22.95 | 72.85 $\pm$ 22.24 | 70.52 $\pm$ 21.65                   | 58.86 $\pm$ 25.34 |
|           |      | specificity | 75.54 $\pm$ 11.66 | 79.25 $\pm$ 10.77 | 82.17 $\pm$ 9.46                    | 87.60 $\pm$ 6.72  |

Table 4.3: Quantative results obtained on the training images when n=80. Average results and their standard deviations are reported.

| n = 80    |      |             | $G_{thr}$         |                   |                   |                   |
|-----------|------|-------------|-------------------|-------------------|-------------------|-------------------|
| $r_{min}$ | $p$  |             | 0.5               | 0.6               | 0.7               | 0.8               |
| 3         | 0.10 | accuracy    | $76.93 \pm 12.27$ | $78.99 \pm 13.22$ | $77.62 \pm 16.59$ | $63.15 \pm 19.85$ |
|           |      | dice index  | $76.97 \pm 16.37$ | $78.23 \pm 16.83$ | $75.72 \pm 19.76$ | $48.52 \pm 31.55$ |
|           |      | sensitivity | $74.66 \pm 19.82$ | $72.63 \pm 19.95$ | $67.58 \pm 23.25$ | $38.81 \pm 30.09$ |
|           |      | specificity | $80.63 \pm 9.70$  | $87.76 \pm 6.98$  | $91.84 \pm 7.12$  | $96.56 \pm 3.87$  |
|           | 0.15 | accuracy    | $79.83 \pm 8.14$  | $81.12 \pm 9.74$  | $78.86 \pm 14.48$ | $62.91 \pm 19.85$ |
|           |      | dice index  | $81.02 \pm 8.70$  | $81.61 \pm 9.71$  | $77.69 \pm 15.76$ | $47.61 \pm 32.11$ |
|           |      | sensitivity | $78.24 \pm 11.81$ | $75.68 \pm 13.67$ | $68.75 \pm 19.89$ | $37.93 \pm 30.05$ |
|           |      | specificity | $81.65 \pm 8.95$  | $88.09 \pm 7.41$  | $93.13 \pm 5.81$  | $97.24 \pm 2.91$  |
|           | 0.20 | accuracy    | $80.00 \pm 8.26$  | $81.14 \pm 9.81$  | $78.76 \pm 14.22$ | $62.39 \pm 20.22$ |
|           |      | dice index  | $80.54 \pm 9.13$  | $81.26 \pm 9.95$  | $77.52 \pm 15.58$ | $46.39 \pm 33.09$ |
|           |      | sensitivity | $75.83 \pm 12.55$ | $73.92 \pm 13.71$ | $68.23 \pm 19.55$ | $37.05 \pm 30.42$ |
|           |      | specificity | $86.17 \pm 7.15$  | $90.59 \pm 7.05$  | $93.99 \pm 4.95$  | $97.53 \pm 2.86$  |
| 4         | 0.10 | accuracy    | $83.53 \pm 7.78$  | $85.51 \pm 7.07$  | $85.38 \pm 9.79$  | $78.14 \pm 15.36$ |
|           |      | dice index  | $85.27 \pm 9.09$  | $86.77 \pm 8.36$  | $86.46 \pm 9.80$  | $77.26 \pm 19.39$ |
|           |      | sensitivity | $87.84 \pm 11.16$ | $87.33 \pm 11.05$ | $84.02 \pm 14.09$ | $70.10 \pm 22.42$ |
|           |      | specificity | $75.90 \pm 11.88$ | $81.35 \pm 8.09$  | $86.15 \pm 7.58$  | $91.27 \pm 5.82$  |
|           | 0.15 | accuracy    | $83.55 \pm 7.82$  | $85.43 \pm 7.46$  | $85.30 \pm 9.84$  | $77.11 \pm 15.99$ |
|           |      | dice index  | $84.84 \pm 9.97$  | $86.40 \pm 9.29$  | $86.07 \pm 10.69$ | $75.58 \pm 21.40$ |
|           |      | sensitivity | $86.07 \pm 12.30$ | $85.41 \pm 12.30$ | $82.42 \pm 14.69$ | $67.86 \pm 23.28$ |
|           |      | specificity | $78.92 \pm 10.05$ | $83.44 \pm 10.14$ | $87.90 \pm 8.66$  | $92.24 \pm 5.10$  |
|           | 0.20 | accuracy    | $82.70 \pm 9.84$  | $83.98 \pm 9.72$  | $84.06 \pm 11.36$ | $76.57 \pm 16.12$ |
|           |      | dice index  | $83.73 \pm 11.25$ | $84.81 \pm 10.83$ | $84.68 \pm 11.94$ | $74.93 \pm 21.59$ |
|           |      | sensitivity | $82.81 \pm 14.66$ | $82.19 \pm 14.55$ | $79.92 \pm 16.25$ | $66.94 \pm 23.32$ |
|           |      | specificity | $82.77 \pm 8.82$  | $85.89 \pm 8.33$  | $89.47 \pm 6.49$  | $92.46 \pm 4.95$  |
| 5         | 0.10 | accuracy    | $72.90 \pm 15.68$ | $73.71 \pm 16.30$ | $74.04 \pm 16.57$ | $70.45 \pm 17.62$ |
|           |      | dice index  | $73.12 \pm 18.81$ | $73.64 \pm 19.17$ | $73.77 \pm 19.53$ | $68.49 \pm 22.37$ |
|           |      | sensitivity | $71.47 \pm 25.72$ | $70.56 \pm 25.20$ | $69.28 \pm 24.66$ | $61.42 \pm 25.34$ |
|           |      | specificity | $76.63 \pm 11.44$ | $79.76 \pm 10.05$ | $81.98 \pm 10.14$ | $85.99 \pm 7.82$  |
|           | 0.15 | accuracy    | $75.96 \pm 13.36$ | $76.55 \pm 13.96$ | $76.55 \pm 14.14$ | $70.71 \pm 17.35$ |
|           |      | dice index  | $76.97 \pm 16.76$ | $77.27 \pm 16.99$ | $77.04 \pm 17.21$ | $68.33 \pm 22.98$ |
|           |      | sensitivity | $76.92 \pm 22.05$ | $75.43 \pm 21.47$ | $73.36 \pm 21.12$ | $61.11 \pm 25.44$ |
|           |      | specificity | $73.91 \pm 11.66$ | $77.37 \pm 11.66$ | $80.63 \pm 9.84$  | $87.00 \pm 7.48$  |
|           | 0.20 | accuracy    | $75.00 \pm 13.50$ | $75.82 \pm 13.93$ | $75.55 \pm 14.29$ | $69.53 \pm 17.33$ |
|           |      | dice index  | $75.31 \pm 18.13$ | $75.74 \pm 18.32$ | $75.33 \pm 18.37$ | $66.66 \pm 22.72$ |
|           |      | sensitivity | $74.04 \pm 22.84$ | $72.60 \pm 22.13$ | $70.26 \pm 21.53$ | $58.62 \pm 25.17$ |
|           |      | specificity | $75.84 \pm 11.43$ | $79.57 \pm 10.59$ | $82.53 \pm 9.36$  | $87.96 \pm 6.45$  |

Table 4.4: The average results and their standard deviations obtained by our method and the pixel-based methods, for the training set.

|                            | Accuracy          | Dice Index        | Sensitivity       | Specificity       |
|----------------------------|-------------------|-------------------|-------------------|-------------------|
| Our Method                 | $85.55 \pm 9.80$  | $86.77 \pm 9.71$  | $84.97 \pm 14.12$ | $85.52 \pm 7.45$  |
| Region-Growing [4]         | $68.58 \pm 12.75$ | $56.12 \pm 30.54$ | $47.24 \pm 29.81$ | $97.70 \pm 8.42$  |
| Directional Filtering [24] | $56.34 \pm 18.31$ | $57.27 \pm 19.71$ | $55.88 \pm 28.48$ | $55.16 \pm 32.47$ |

Table 4.5: The average results and their standard deviations obtained by our method and the pixel-based methods, for the test set.

|                            | Accuracy          | Dice Index        | Sensitivity       | Specificity       |
|----------------------------|-------------------|-------------------|-------------------|-------------------|
| Our Method                 | $84.57 \pm 8.02$  | $86.05 \pm 8.80$  | $84.50 \pm 13.28$ | $83.65 \pm 7.30$  |
| Region-Growing [4]         | $67.62 \pm 17.17$ | $59.04 \pm 30.00$ | $52.59 \pm 32.88$ | $87.48 \pm 15.12$ |
| Directional Filtering [24] | $53.24 \pm 13.62$ | $54.33 \pm 19.69$ | $53.77 \pm 25.67$ | $51.67 \pm 33.64$ |

Secondly, we compare our results with those of the *object-graphs* method, which was previously developed in our research group [9]. This *object-graphs* method quantized the given image into three clusters, which correspond to nucleus, stroma, and lumen, by using the k-means algorithm. Then, the circle-fit algorithm is applied to fit circles on the nucleus and lumen clusters, separately. An object graph is constructed on lumen objects, by considering nucleus and lumen objects as nodes by drawing edges between each lumen object and its closest nuclear and lumen objects. Then, structural features are extracted for each lumen object, by taking advantage of the edges which are drawn in the graph construction process. According to these features, lumen objects are grouped into two classes as gland and non-gland, by using the k-means algorithm. Furthermore, another object graph is constructed on nucleus objects and edges are drawn between each nucleus object and its closest objects. Then, a region growing algorithm is applied to the gland objects. The algorithm stops when a gland structure reaches to the edges of the nucleus-graph. Finally, small regions are eliminated, and a decision tree classifier is used to eliminate false glands.

Tables 4.6 and 4.7 shows the results of the object-graph method and our proposed method. According to the comparison results, our proposed algorithm improves the segmentation results of the object-graph approach before applying a decision tree classifier. This shows that the object-graph approach necessitates using a supervised classifier whereas our algorithm leads to good results without using such a classifier. When such a classifier is incorporated to our method, the results might be further improved. This is considered as a future work of this thesis.

Table 4.6: Results of the object-graph method and our proposed method, for the training set.

|                                                           | Accuracy          | Dice Index        | Sensitivity       | Specificity       |
|-----------------------------------------------------------|-------------------|-------------------|-------------------|-------------------|
| Our Method                                                | $85.55 \pm 9.80$  | $86.77 \pm 9.71$  | $84.97 \pm 14.12$ | $85.52 \pm 7.45$  |
| Object-graph (before applying a decision tree classifier) | $77.39 \pm 14.99$ | $81.73 \pm 12.71$ | $89.61 \pm 4.28$  | $65.09 \pm 25.44$ |
| Object-graph (after applying a decision tree classifier)  | $88.00 \pm 4.16$  | $88.46 \pm 4.62$  | $83.43 \pm 7.73$  | $92.30 \pm 5.78$  |

Table 4.7: Results of the object-graph method and our proposed method, for the test set.

|                                                           | Accuracy         | Dice Index       | Sensitivity       | Specificity       |
|-----------------------------------------------------------|------------------|------------------|-------------------|-------------------|
| Our Method                                                | $84.57 \pm 8.02$ | $86.05 \pm 8.80$ | $84.50 \pm 13.28$ | $83.65 \pm 7.30$  |
| Object-graph (before applying a decision tree classifier) | $82.57 \pm 8.36$ | $85.59 \pm 7.73$ | $90.62 \pm 5.44$  | $72.80 \pm 15.38$ |
| Object-graph (after applying a decision tree classifier)  | $87.59 \pm 5.01$ | $88.91 \pm 4.63$ | $85.80 \pm 6.71$  | $89.14 \pm 10.40$ |

## Chapter 5

### Conclusion

In this thesis, we propose a new structural gland segmentation algorithm for histopathological images. The main contribution of this thesis is as follows: our proposed algorithm enables to encode the appearance of glands by using the defined gland generation rules and its fitness score, which have not been used by the previous methods. The results of our algorithm show that our proposed method improves the segmentation results of its pixel-based and structural counterparts without applying any further processing.

In the first part of this algorithm, a new representation is obtained for the given images, by using the primitives which are defined during the Voronoi diagram generation process. This representation facilitates the encoding of our observations and the quantification of *GlandScore*. In the second part of the algorithm, a set of rules are defined according to the spatial distribution of the primitives for candidate generation. Finally, an iterative algorithm is defined for the selection of the best candidates among the defined candidate sets.

As a future work, in order to outperform the segmentation results of the object-graph method after applying a decision tree classifier, two different approaches can be followed. In the first one, some features might be defined similar to the object-graph method. By decreasing  $G_{thr}$  to a reasonable value, more true positive and

false positive glands will be obtained. Then, by utilizing the defined features, such a classifier could be trained on these glands. In the second approach, the detected glands from our iterative algorithm can be considered as initial glands and a new method can be defined for the ones which are not detected. In this second method, focusing on the glands that are located on images' borders might lead to improved results.





# Bibliography

- [1] M. Malvezzi, G. Carioli, P. Bertuccio, P. Boffetta, F. Levi, C. La Vecchia, and E. Negri, “European cancer mortality predictions for the year 2018 with focus on colorectal cancer,” *Annals of Oncology*, vol. 29, no. 4, pp. 1016–1022, 2018.
- [2] R. L. Siegel, K. D. Miller, S. A. Fedewa, D. J. Ahnen, R. G. Meester, A. Barzi, and A. Jemal, “Colorectal cancer statistics, 2017,” *CA: A Cancer Journal for Clinicians*, vol. 67, no. 3, pp. 177–193, 2017.
- [3] M. Arnold, M. S. Sierra, M. Laversanne, I. Soerjomataram, A. Jemal, and F. Bray, “Global patterns and trends in colorectal cancer incidence and mortality,” *Gut*, pp. gutjnl–2015, 2016.
- [4] H.-S. Wu, R. Xu, N. Harpaz, D. Burstein, and J. Gil, “Segmentation of intestinal gland images with iterative region growing,” *Journal of Microscopy*, vol. 220, no. 3, pp. 190–204, 2005.
- [5] 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*, pp. 1–8, Citeseer, 2007.
- [6] H. Chen, X. Qi, L. Yu, and P.-A. Heng, “Dcan: Deep contour-aware networks for accurate gland segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2487–2496, 2016.
- [7] S. Manivannan, W. Li, J. Zhang, E. Trucco, and S. J. McKenna, “Structure prediction for gland segmentation with hand-crafted and deep convolutional

- features,” *IEEE Transactions on Medical Imaging*, vol. 37, no. 1, pp. 210–221, 2018.
- [8] Y. Xu, Y. Li, Y. Wang, M. Liu, Y. Fan, M. Lai, I. Eric, and C. Chang, “Gland instance segmentation using deep multichannel neural networks,” *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 12, pp. 2901–2912, 2017.
- [9] C. Gunduz-Demir, M. Kandemir, A. B. Tosun, and C. Sokmensuer, “Automatic segmentation of colon glands using object-graphs,” *Medical Image Analysis*, vol. 14, no. 1, pp. 1–12, 2010.
- [10] K. Sirinukunwattana, D. R. Snead, and N. M. Rajpoot, “A stochastic polygons model for glandular structures in colon histology images,” *IEEE Transactions on Medical Imaging*, vol. 34, no. 11, pp. 2366–2378, 2015.
- [11] P. Peckham, Knibbs, *Epithelia*, 2003 (accessed August 29, 2018).
- [12] E. Jorundsson, J. H. Lumsden, and R. M. Jacobs, “Rapid staining techniques in cytopathology: a review and comparison of modified protocols for hematoxylin and eosin, papanicolaou and romanowsky stains,” *Veterinary Clinical Pathology*, vol. 28, no. 3, pp. 100–108, 1999.
- [13] A. Banwari, N. Sengar, M. K. Dutta, and C. M. Travieso, “Automated segmentation of colon gland using histology images,” in *Contemporary Computing (IC3), 2016 Ninth International Conference on*, pp. 1–5, IEEE, 2016.
- [14] S. Xie and Z. Tu, “Holistically-nested edge detection,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1395–1403, 2015.
- [15] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in *Advances in Neural Information Processing Systems*, pp. 91–99, 2015.
- [16] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Ssstrunk, “SLIC superpixels,” tech. rep., 2010.

- [17] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [18] P. J. Green, "Reversible jump markov chain monte carlo computation and bayesian model determination," *Biometrika*, vol. 82, no. 4, pp. 711–732, 1995.
- [19] H. Fu, G. Qiu, J. Shu, and M. Ilyas, "A novel polar space random field model for the detection of glandular structures," *IEEE Transactions on Medical Imaging*, vol. 33, no. 3, pp. 764–776, 2014.
- [20] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," 2001.
- [21] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.
- [22] A. C. Ruifrok and D. A. Johnston, "Quantification of histochemical staining by color deconvolution," *Analytical and Quantitative Cytology and Histology*, vol. 23, no. 4, pp. 291–299, 2001.
- [23] A. B. Tosun and C. Gunduz-Demir, "Graph run-length matrices for histopathological image segmentation," *IEEE Transactions on Medical Imaging*, vol. 30, no. 3, pp. 721–732, 2011.
- [24] H.-S. Wu, R. Xu, N. Harpaz, D. Burstein, and J. Gil, "Segmentation of microscopic images of small intestinal glands with directional 2-d filters," *Analytical and Quantitative Cytology and Histology*, vol. 27, no. 5, pp. 291–300, 2005.