Automatic Segmentation of Cells of Different Types in Fluorescence Microscopy Images

Hamid Fehri
Department of Infection, Immunity and Cardiovascular Disease

The University Of Sheffield.

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (PhD)

Supervisors: Simon A. Johnston and Alejandro F. Frangi

December 2018

Abstract

Recognition of different cell compartments, types of cells, and their interactions is a critical aspect of quantitative cell biology. This provides a valuable insight for understanding cellular and subcellular interactions and mechanisms of biological processes, such as cancer cell dissemination, organ development and wound healing. Quantitative analysis of cell images is also the mainstay of numerous clinical diagnostic and grading procedures, for example in cancer, immunological, infectious, heart and lung disease. Computer automation of cellular biological samples quantification requires segmenting different cellular and sub-cellular structures in microscopy images. However, automating this problem has proven to be non-trivial, and requires solving multi-class image segmentation tasks that are challenging owing to the high similarity of objects from different classes and irregularly shaped structures.

This thesis focuses on the development and application of probabilistic graphical models to multi-class cell segmentation. Graphical models can improve the segmentation accuracy by their ability to exploit prior knowledge and model inter-class dependencies. Directed acyclic graphs, such as trees have been widely used to model top-down statistical dependencies as a prior for improved image segmentation. However, using trees, a few inter-class constraints can be captured. To overcome this limitation, polytree graphical models are proposed in this thesis that capture label proximity relations more naturally compared to tree-based approaches. Polytrees can effectively impose the prior knowledge on the inclusion of different classes by capturing both same-level and across-level dependencies. A novel recursive mechanism based on two-pass message passing is developed to efficiently calculate closed form posteriors of graph nodes on polytrees. Furthermore, since an accurate and sufficiently large ground truth is not always available for training segmentation algorithms, a weakly supervised framework is developed to employ polytrees for multi-class segmentation that reduces the need for training with the aid of modeling the prior knowledge during segmentation. Generating a hierarchical graph for the superpixels in the image, labels of nodes are inferred through a novel efficient message-passing algorithm and the model parameters are optimized with Expectation Maximization (EM).

Results of evaluation on the segmentation of simulated data and multiple publicly available fluorescence microscopy datasets indicate the outperformance of the proposed method compared to state-of-the-art. The proposed method has also been assessed in predicting the possible segmentation error and has been shown to outperform trees. This can pave the way to calculate uncertainty measures on the resulting segmentation and guide subsequent segmentation refinement, which can be useful in the development of an interactive segmentation framework.

Acknowledgment

First and foremost, I would like to express my sincere gratitude to my supervisors, Dr. Simon Johnston and Prof. Alejandro Frangi, for their invaluable guidance, support and absolute commitment to the project. I would like to thank Dr. Ali Gooya for his tremendous and wholehearted technical support, patience and devotion to the project.

I would really like to thank Mr. Miguel Carmona for his providing technical advice in programming. I thank Dr. Matthias Lange for the motivating discussions and the useful tips in my work. I also thank Mr. Yuanjun Lu, for his time for discussing the use of Sheffield high performance computing cluster. Special thanks to Miss. Amy Lewis, Mrs. Aleksandra Bojarczuk, and Dr. Philip Elks for their being welcoming and making practical comments, while listening to my presentations being full of maths equations, as my colleagues from the biology background. I cannot value your efforts for involvement in discussions and giving useful feedback. I would also express my gratitude to Dr. Vladimir Ulman and Dr. David Svoboda for providing me with the synthetic set of cell images for evaluating my methods.

I would like to thank Dr. Erik Meijering for hosting me during my visit to the Biomedical Imaging Group of the Erasmus University Medical Center in Rotterdam, the Netherlands. I am also thankful of Prof. Wiro Nissan for facilitation of the visit and his hospitality. My very special thanks to my colleagues and ultimate friends there, Dr. Ihor Smal, Miss. Yao Yao, Mrs. Zahra Sedghi Gamechi, Mr. Gennady Roshchupkin and Dr. Leila Karimi. I also want to sincerely thank my friends for being there to listen and help whenever I needed them, especially to Dr. Ali Sarrami and Mr. Mohammad Ali Khademi, the two I can name from the long list.

Last but not least, I should thank my dearest parents, sister and brother for providing me with unfailing support, continuous encouragement and motivation during and prior to this course, who have given up many things for me to be at Sheffield University. I would also like to thank Mr. Ahmad Nezami and Mr. Saeed Nakhaee who have always been of great help. Finally, the University of Sheffield scholarship is gratefully acknowledged for their financial support of the project.

Preface

This thesis is presented in the alternative thesis format containing chapters which are suitable for publication in peer-reviewed journals, alongside more traditional thesis chapters. I start with the Introduction chapter that presents the motivation of the work and gives an overview of the thesis. The second chapter, Biological Background, discusses the significance of cell imaging and analysis in the clinical diagnosis and biomedical research. An introduction to the current methods for biological staining and microscopy image acquisition are also presented in this chapter. Chapter three reviews the existing methods for modeling the prior knowledge in image segmentation and introduces the fundamental background of the graph theory as the basis of the proposed methods in the following chapters. A review of current graphical models for segmentation as well as other main state-of-the-art segmentation methods are also provided in this chapter.

Chapter four explains the proposed polytree graphical models for multi-class cell segmentation, which is an extension of the paper presented in the Information Processing in Medical Imaging (IPMI) 2017 conference. Following a submission of this extension to a journal, the first revision of it has been recently submitted. Chapter five presents weakly supervised hierarchical Bayesian networks for segmentation of cell images in microscopy. This chapter has also been submitted to a journal.

A summary of the contributions and applications of the proposed methods, along with the limitations and directions of the future work have been included in chapter six, Discussion and Conclusions.

DECLARATION

Hereby, I (Hamid Fehri) affirm that this dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except as specified in the text.

This dissertation is not substantially the same as any that I have submitted, or, is being concurrently submitted for a degree or diploma or other qualification at the University of Sheffield or any other University or similar institution.

I further state that no substantial part of my dissertation has already been submitted, or, is being concurrently submitted for any such degree, diploma or other qualification at the University of Sheffield or any other University or a similar institution.

Publications

Journal articles during the thesis

[1] Nemat, H.; **Fehri H.**; Ahmadinejad N.; Frangi A. F., and Gooya, A., ”Classification of Breast Lesions in Ultrasonography Using Sparse Logistic Regression and Morphology Based Texture Features.” *Medical Physics 45*, no. 9 (2018): 4112-4124.

[2] **Fehri, H.**; Gooya, A.; Lu, Y.; Meijering, E.; and Frangi A. F., “Bayesian Polytrees with Learned Deep Features for MultiClass Cell Segmentation.” *IEEE Transactions on Image Processing*, (2019).

[3] **Fehri, H.**; Gooya, A.; Lu, Y. and Frangi A. F., “Hierarchical Bayesian Networks for Weakly Supervised Cell Segmentation in Light Microscopy.”, *Submitted.*

Published conference proceeding articles during the thesis

[4] **Fehri, H.**; Gooya, A.; Johnston S. A, and Frangi A. F. “Multi-class Image Segmentation in Fluorescence Microscopy Using Polytrees.” *International Conference on Information Processing in Medical Imaging, pp. 517-528. Springer, Cham*, 2017.

Presented conference articles during the thesis

[5] **Fehri, H.**; Gooya, A.; Johnston S. A, and Frangi A. F. “Hierarchical Bayesian Networks for Modeling Inter-Class Dependencies: Application to Semi-Supervised Cell Segmentation.” *2nd Workshop on GRaphs in biomedicAl Image anaLysis at the International Conference on Medical Image Computing and Computer Assisted Intervention*, 2018.

Contents

List of Figures

List of Tables

Abbreviations

BN Bayesian Network

CAD Computer Aided Diagnosis

CCD Charge-Coupled Device

CMOS Complementary Metal Oxide Semiconductor

CNN Convolutional Neural Network

DAG Directed Acyclic Graph

DIC Differential Interference Contrast

DNA DeoxyriboNucleic Acid

DoG Difference of Gaussian

DSC Dice Similarity Coefficients

EM Expectation Maximization

FCN Fully Convolutional Networks

FP Fluorescent Protein

GFP Green Fluorescent Protein

GMM Gaussian Mixture Model

GUI Graphical User Interface

HINTS Hierarchically-structured Ineracting Segments

H& E Haemotoxylin and Eosin

LED Light Emitting Diode

LoG Laplacian of Gaussian

MRF Markov Random Field

MRI Magnetic Resonance Imaging

PEDT Position-Encoding Dynamic Tree

RNA RiboNucleic Acid

SBM Shape Boltzman Machine

SVA Structured Variational Approximation

SVM Support Vector Machine

List of Symbols

$\boldsymbol{\beta}$	The similarity index for graph generation
$\boldsymbol{\mu}_c$	The mean value of a class c with a Gaussian distribution
$\boldsymbol{\Sigma}_c$	The covariance matrix of a class c with a Gaussian distribution
$\boldsymbol{\Theta}$	The set of all parameters subject to optimization during the Expectation-Maximization
$\boldsymbol{w}_c$	Weights of the observation vector for a class c with a Gaussian distribution
$\boldsymbol{y}$	An observation node on a graphical model
Λ	The set of all possible labels for the random variables on a graphical model
$\mathbb{F}$	The set of all possible label configurations on a graphical model
$\mathbb{O}$	The set of orphan nodes on a graphical model
$\mathbb{X}$	The set of random variables x_s on a graphical model
$\mathbb{Y}$	The set of all observation nodes on a graphical model
$\mathcal{G}$	A graphical model
$\mathcal{S}$	The set of all nodes in a graphical model
$\mathcal{Y}_{a(s)}$	The set of observation nodes for the ascendant nodes of a node s on a graphical model
$\mathcal{Y}_{d(s)}$	The set of observation nodes for the descendant nodes of a node s on a graphical model
$\perp\!\!\!\perp$	Conditional independence

ψ_C	The potential function for a clique C on a graphical model
D	The number of features used for graph generation
DSC	Dice Similarity Coefficient
E	Set of edges in a graphical model
F	The number of features used for inference
K	The total number of classes
m_d	Mean value of the d^{th} feature
N	The total number of nodes in a graphical model
n	Number of iteration in the Expectation-Maximization
$p(x)$	Probability distribution function of the random variable x
pa_s	The set of parents of a node s
Q	The log-likelihood function subject to maximization during the Expectation-Maximization
s_i	Node i from the set of nodes V in a graphical model
V	Set of nodes in a graphical model
W_d	Fisher score for the d^{th} feature
x	A random variable on the graphical model
$x_{s'_i}$	Same-level node to a node s on a graphical model that is a mutual sibling with a node s
$x_{s_i^+}$	Descendant node from a node s on a graphical model
$x_{s_i^-}$	Ascendant node from a node s on a graphical model
Z	Partition function

Chapter 1

Introduction

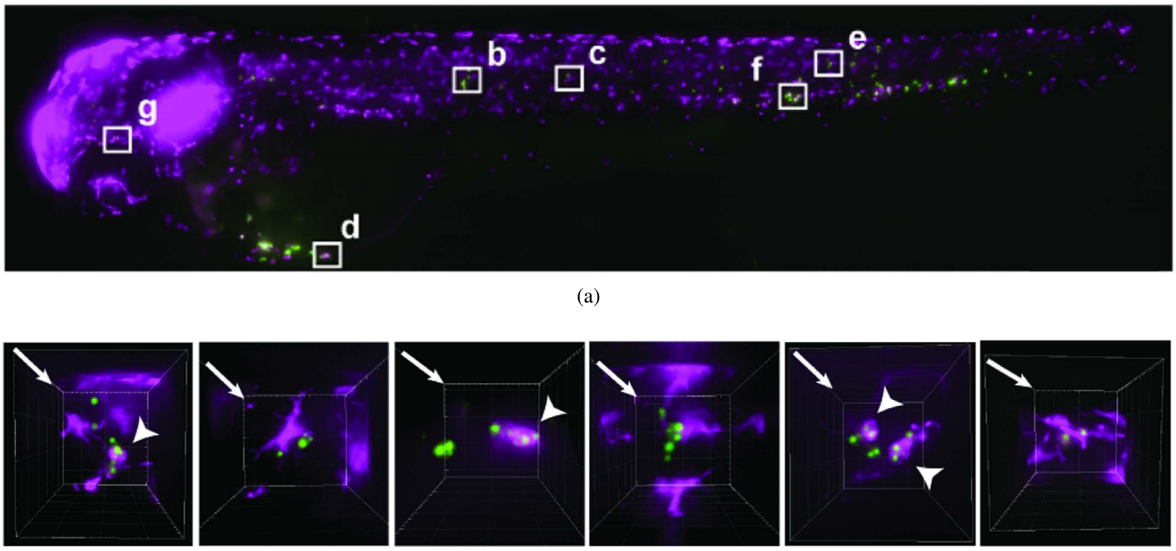
1.1 Introduction

This chapter describes the problem addressed in this thesis and will introduce the need, the required setup, and tools for experimenting on cells and generating microscopy images. The data that is used in this thesis is introduced, and its acquisition and the biological systems used are explained. Medical and research applications of histology image acquisition and analysis are reviewed in the first section, followed by experimental setups that are required for investigating cells. Next, different methods of biological staining and common microscopy image acquisition techniques are reviewed and the state-of-the-art technology is introduced.

1.2 Motivation

This thesis is motivated by the problems defined and dealt with by the Johnston lab at The University of Sheffield. The Johnston lab uses biological cell imaging to study infectious disease to improve patient treatment and to further the fundamental understanding of how the immune system functions to fight infection. Figure 1.1 shows a sample microscopy image from a zebrafish model of the immune system and its response to infection. Quantifying images of cells manually involves laborious work that can be inconsistent, prone to error and is subjective. Therefore, developing automatic algorithms that can analyze different types of cells and their relations in the image will increase analysis efficiency and reliability of the results. It is also desirable that such algorithms could be applied to images of different modalities and specimens once developed.

On the other hand, the Center for Computational Imaging and Simulation Technologies in Biomedicine (CISTIB) is interested in addressing the gaps in the state of the art by developing novel mathematical tools and applying them to the biomedical research and digital pathology. The proposed segmentation methods in this thesis, provide an effective way of modeling relations between objects in an image that also incorporates the prior knowledge. This can improve the performance of image analysis when training is available, and can reduce reliance on the training when there is limited ground truth available for the problem. Furthermore, the overwhelming growth of the spatial and temporal resolution of images has



(a)

(b) (c) (d) (e) (f) (g)

Figure 1.1: Quantification of host cell (macrophage) behavior in response to infection (caused by *Cryptococcus)*. Panel (a) shows maximum intensity z-projection from high content imaging. Panels (b-g) depict the boxed areas in (a) that were enlarged and reconstructed in three dimensions, where arrowheads point to intracellular pathogens. Image modified and adapted from [1].

urged the need for different tools for analyzing big data. The proposed techniques of the thesis address this aspect by the efficient algorithms that generate exact solutions for the segmentation problem.

1.3 Problem definition

This thesis aims to address the problem of automatic quantification of images of cells by the development of a new image quantification tool to address the need for image analysis techniques with improved performance compared to the state of the art. This is a common problem in cell image analysis where cells and their compartments, or host cells and pathogens exist in an image. The significance of the need for image analysis techniques for clinical applications and biomedical research is described in the following two sections.

Microscopy image analysis plays a key role in decision making in disease diagnosis by providing quantitative analysis of pathology images. Addressing the labor-intensity, inconsistency and bias problems of manual analysis renders automatic methods suitable for analyzing large image datasets such as those of high throughput imaging [2]. Computerized analyses are reproducible with a reduced bias and can provide personalized treatments for patients [3]. With the advancement of techniques for analyzing histology images, these images have been increasingly used for disease diagnosis and grading [4]. Examples include investigating the size and shape of nuclei as clues for cancer diagnosis and progression [4–6], disease identification by analyzing the organization of nucleus genome and proteins [7–9], differential cell counts in Pap smear for cancer or precancer of the uterine cervix [10], measuring the proportion of viremic cells during infection [11, 12], and investigating the role of cell compartments and their characteristics in disease outcome [1].

From a research point of view, quantitative analysis of microscopy images can provide a valuable insight into understanding cellular and subcellular interactions. Observing the changes in cells shapes helps to understand the mechanisms of different biological processes. For example, in synapse formation, which is required for cell-cell recognition, spatial receptors and the shape of the cells in the inter-cellular junction undergo evolutions. Quantitative analysis of physicochemical processes and measuring different binding characteristics, protein mobilities and membrane constraints are required for studying synapse formation (synaptogenesis), which can be achieved by employing image analysis techniques. Further, cell division and organ development are other active research areas that involve massive changes in cell shapes that can benefit from an automatic and more accurate quantification of cell images [13].

Tracking cells over time is highly informative in understanding their role in health and disease. For example, collective cell movement is a process whereby groups of cells move jointly without completely losing contact. This is of particular importance in the development of morphological characteristics as well as pathological processes such as dissemination of cancer cells to secondary sites (metastasis) and wound healing. Characterization and analysis of cell trajectories in movement, both individually and in pairs, is done by measuring the velocity, orientation, and entropies [14].

Other than the ability to help in studying cell migration [15, 16], image analysis methods can provide segmentations of the microscopy image, in which surfaces and boundaries of the visual objects in the image are delineated. Image segmentation can offer a more concise localization of subcellular proteins and in turn, provides the physiological context for their function [17–20].

Examples of cell shape change and movement, organelle localization, behavior and movement presented above show a few samples from the numerous cellular and subcellular events in cellular biology, studying which can lead to biological discoveries, such as identification of cells in the tissue [21], identifying organelles in cells [22], localizing proteins within cells [23], and cell interactions during organ development [24].

In the following, I provide an introduction to the tools and protocols that are required to generate microscopy images for biomedical research and clinical pathology.

1.4 Data used in this thesis

Different microscopy techniques can be used to study cells, each having their own advantages and disadvantages. These include but are not limited to brightfield, phase contrast, differential interference contrast (DIC), darkfield, confocal, lightsheet and electron microscopy techniques. This thesis focuses on the analysis of stained transmitted light and widefield fluorescence images, being the most common techniques used for cell analysis.

To evaluate the performance of the proposed methods and compare their performance to the state of the art, five image datasets have been used in this work. Manual annotations are available for all of the mentioned datasets.

Two real image datasets were chosen from the publicly available datasets on Broad Bioimage Benchmark Collection that contain two-channel fluorescence microscopy images with cells and nuclei, namely BBBC020 and BBBC007 datasets [25]. BBBC020 contains 20 two-channel *in vitro* microscopy images of murine bone marrow macrophages to investigate the role of toll-like receptor 4 for macrophage spreading. Toll-like receptors (TLRs) are a class of proteins that play a key role in the innate immune system. They are single, membrane-spanning receptors usually expressed on sentinel cells such as macrophages and dendritic cells, that recognize structurally conserved molecules derived from microbes. These receptors enable the immune system to recognize the threats by recruiting immune cells like macrophages. BBBC007 has 16 two-channel *in vitro* microscopy images of drosophila Kc167 cells. Drosophila is the genus of fruit flies that is used in developmental and evolutional studies and genetics, having a small genome which has been fully sequenced. Tissue culture cells in drosophila are a powerful tool for identifying the genes underlying cell biological functions and for annotating the fly genome. These two datasets have the same type of images and define similar multi-class segmentation problems of cells and nuclei. The BBBC007 dataset has a larger number of overlapping cells and noisier images, which makes the segmentation more challenging.

A synthetic dataset with 25 simulated images of HL60 cell nuclei and sub-nuclear particles, namely nucleoli, generated by the Mitogen software [26] have also been employed. The HL-60 cell line is a group of human blood cancer cells (leukemia cell line) introduced for research on blood cell formation and physiology. HL-60 proliferates continuously in suspension culture in nutrient and antibiotic chemicals. Mitogen is a framework for generating synthetic time-lapse sequences of cell populations in fluorescence microscopy. This framework can create image datasets for verification of image quantification algorithms with a computer-generated ground truth that provides a higher quality compared to manual annotations and can be generated in large quantities.

The zebrafish dataset having 10 random two-channel microscopy images of host and pathogen cells, viz. macrophages and the fungal pathogen *Cryptococcus neoformans* published in [1] was used as another dataset for benchmarking the different methods presented in this thesis. *Cryptococcus* is a significant fungal pathogen that infects immunocompromised patients causing fatal lung infection and meningitis. This dataset has been generated to study the function of macrophages in normal clearance of cryptococcal infection and the defects present in uncontrolled cryptococcosis. In this dataset, macrophages were labeled with mCherry and the cryptococci were GFP-labelled and the fish were imaged using a brightfield microscope. Zebrafish is a well-resembling model for the study of human disease having a similar immune system to that of humans, an easy and fast breeding and transparent embryos that allow the researchers to easily examine the development of its internal structures.

Furthermore, the ISBI2015 Challenge dataset with 900 images of cytoplasms and nuclei of cervical cells obtained from Pap smear [27] was used. Pap smear is a procedure to test for cervical cancer in women by analyzing cells collected from the cervix. Examination of the microscopy images taken from the extracted mucus is used to detect pre-cancerous and cancerous processes.

The datasets used in this thesis are used to evaluate the performance of the techniques developed as part of this project. However, as it will be explained in detail in the following chapters, the core mathematical methodology presented can be applied to images generated by different techniques. The first steps of the presented image quantification algorithm, namely image oversegmentation and graph generation steps, generate a probabilistic graphical model from the input image. The generated graphical model will then be analyzed using the inference techniques to find the most appropriate labelling of the graph, which will then be used for generating the output. For the presented evaluations of the proposed method, the first steps of the algorithm have been tuned for their application to the type of images used in this work. For application to other types of (microscopy) images, the preliminary steps would need to be matched with the prospective data types.

The following three sections describe the experimental setup and biological staining techniques employed in the datasets used in this thesis.

1.5 Experimental systems for analyzing cells

Understanding many basic biochemical and physical processes relies on studying cells involved in biological functions. This is made possible through different systems that have been designed for doing experiments on them.

Intravital microscopy (IVM) which allows high-resolution imaging of cells in living organisms such as mouse [28], zebrafish [29], Drosophila [30] and *C. elegans* [31], is an essential method for visualizing the dynamic behavior of cells in the tissue. Most of the IVM techniques are based on fluorescent light microscopy and can produce high-resolution images (up to $0.325\mu m^2$ per pixel [32]), which can be used for cellular and subcellular analysis.

Alternatively, dissection has been used for centuries for understanding anatomy. In this method, culture specialized cells are isolated from specific regions, e.g. cortical and hippocampal neurons of the brain, and are dissected to study the body of the deceased animal or plant, using *in vitro* imaging. The specimen can be fresh or prepared, being dissected within days and therefore retaining the characteristics of the living specimen or preserved in solutions and predissected by an anatomist.

Cells can also be isolated directly from living tissue or following immortalization of cell lines. Cell to cell differences can be the key to answer questions in cancer, neurobiology, stem cell biology, immunology, and developmental biology. Despite cell-based assays that analyze the average responses from a population of cells, single cell analyses, where individual single cells are isolated from a complex sample, can provide more detailed information for biomedical and therapeutic research [33].

1.6 Biological staining

Most cells are colorless and translucent, therefore they require staining to make them visible. Staining or use of optical contrast enhancement (see section 1.6.1) can be non-specific or specific, where either most of the cells are stained in the same way or particular chemical groupings or molecules are selectively stained, respectively. A dye is usually used in staining that stains some of the cell components with a bright color and the rest of them with a different color (counterstain). In the following, I briefly review the conventional stains, used in cell microscopy.

1.6.1 Dye or stain

H& E is the most common staining system that contains the two dyes Haemotoxylin and Eosin. Using this staining, the DNA (heterochromatin and the nucleolus) in the nucleus, RNA in ribosomes, and parts of the cytoplasm that contain RNA turn purple, and the rest of the cytoplasm (most of the proteins such as cytoplasmic filaments in muscle cells, intracellular membranes, and extracellular fibers) turn pink. Giemsa stain is usually used for blood and bone marrow smears, where nuclei are stained dark blue, cytoplasm pale blue and erythrocytes pale pink. Lipid structures in the cell, such as myelin, can be colored with Sudan Black and Osmium dyes.

Additionally, the bacterial cell wall is one of the most commonly used stains in clinical applications, in which bacteria can appear purple or pinkish orange, called Gram positive and Gram negative, respectively. DAPI is a fluorescent stain that binds strongly to adenine-themine rich regions in DNA and is extensively used in fluorescence microscopy. This stain-

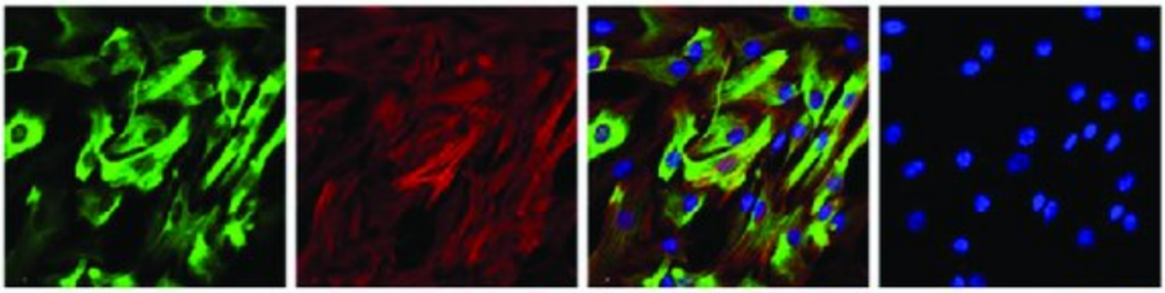


Figure 1.2: Immunofluorescence staining of human tumor cells with mouse antibody (green fluorescence), phalloidin (red fluorescence), DAPI (blue fluorescence). Individual and merged images are shown [34].

ing is excited by the violet laser line and is commonly used as a nuclear counterstain in fluorescence microscopy, flow cytometry, and chromosome staining. Being cell impermeant, DAPI is usually used to stain fixed cells, even though it will enter live cells when used at higher concentrations. In fixed cells, actin structures can be visualized with fluorescently labeled phalloidins.

1.6.2 Antibody

Immunofluorescence techniques use a primary antibody to label a specific protein and a fluorescently labeled secondary antibody to bind to the primary one. Using fluorescent light microscopy, the fluorescently labeled antibodies are excited and emit light (at different wavelengths) that can be visualized through appropriate filtering. For example, actin filaments that are the smallest filaments in the cell (about 7nm in diameter) can be immunofluorescently stained and visualized in the images. Figure 1.2 shows a sample set of cultured cells stained with a mouse antibody, phalloidin, and DAPI.

1.6.3 Genetically fluorescently tagged protein

Genetically engineered fluorescent proteins (FP) allow specific proteins to be directly visualized in real time to facilitate studying the binding patterns and dynamics of living proteins in cells. Additionally, these tags are suitable for *in vitro* applications of imaging single proteins bound to DNA to understand its localization and behavior. Green Fluorescent Protein

(GFP) is the most famous FP that emits bright green fluorescence when exposed to blue light. Many other derivatives of GFP have been engineered, for example, blue, cyan and yellow fluorescent proteins that bind to different materials.

1.7 Microscopy image acquisition

Primary techniques in optimal microscopy involve brightfield illumination that relies on changes in light absorption, refractive index, and color for generating contrast. When light passes through the specimen and the rays are focused by the objective, areas that alter the direction, speed and spectrum of the light waves generate optical disparities, that is called contrast. In a brightfield system, the image resolution depends on the objective and condenser numerical apertures. An immersion medium is required on both sides of the specimen. The limitations of the brightfield imaging in visualizing the cell outline and nucleus, and the low contrast of its images reduce its use for studying living cell structures and have led to the development of more modern techniques.

1.7.1 Transmitted light microscopy

Transmitted light microscopy involves passing light from the illumination source on the opposite side of the specimen to the objective so that the light is transmitted through the specimen. Since the quality of the observation and image recording is not always as expected in simple brightfield imaging, specialized contrast-enhancing methods are required to increase the imaging quality. Phase contrast, Differential Interference Contrast (DIC) and Darkfield are examples of these techniques.

1.7.2 Fluorescence microscopy

Fluorescence light microscopy is an indispensable approach for analyzing the biological mechanisms in cells. This method has become more powerful with the introduction of new tools such as genetically encoded fluorescent proteins and superresolution methods, which have been used to uncover many different mechanisms within cells such as cell division [35]

and cell motility [36].
Most fluorescence microscopes have the following essential components:

- A **light source**, that is usually a xenon or mercury lamp or an LED.
- An **excitation filter**, that narrows the wavelengths of the incoming light down the wavelengths that can result in light emission from the sample.
- A **mirror**, that reflects the excitation light to the sample and transmits the emitted light from the sample to the detector,
- An **emission filter**, that transmits only the wavelengths of the emitted light from the sample and blocks the light passed through the excitation filter
- A **camera**, to detect the emitted light, which is usually connected to a computer to generate the image.

A schematic of a typical set-up for a fluorescence microscope is shown in Figure 1.3.

In fluorescent light microscopy, light of specific wavelengths can excite fluorescent objects, such as GFP-labeled tumor cells, resulting in the emission of photons and detecting the objects.

Epifluorescence microscopes are arranged such that the excitation light from the light source and the emitted light from the specimen pass through the same objective. The majority of the fluorescence microscopes are of the epifluorescence design which are the basis for more advanced microscopes such as confocal and lightsheet microscopes.

1.7.3 Image detection

Advancements in fluorescence microscopy over the past years have led to the evolution of image production from the emulsion-based film to the computer images of the desired type. Electronic imaging sensors can be described by different characteristics including spatial resolution, uniformity, signal to noise, response speed and dynamic range.

Charge-Coupled Device (CCD) is a solid-state detector consisting of a dense matrix of photodiodes that incorporate electric charge storage regions. Each storage is coupled to a silicon diode photosensor (a pixel) and to an amplifier that measures the quantity of the

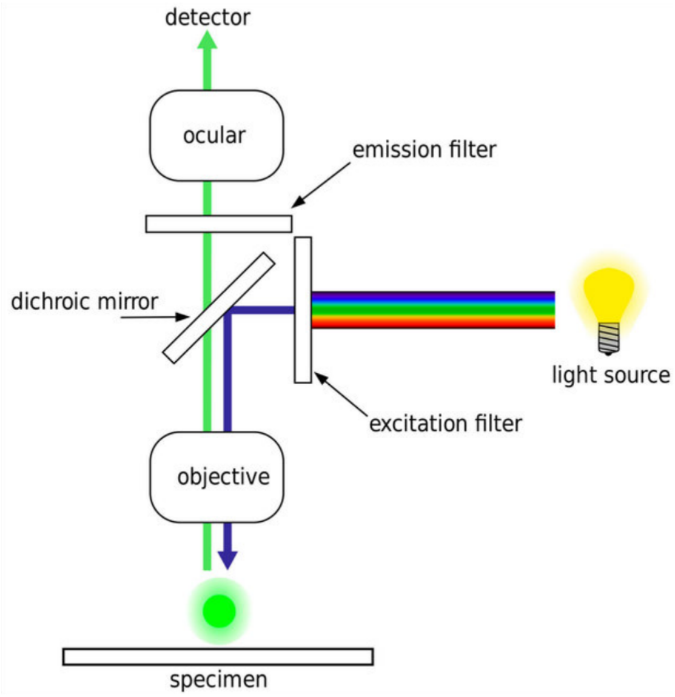


Figure 1.3: Optical set-up of a standard fluorescence microscope with the parts explained in the text. The picture has been adapted from [37].

accumulated charge. Complementary Metal Oxide Semiconductors (CMOS) are another type of solid-state detectors, in which each individual photosensor has an amplifier associated with it and the combined signal from a row of amplifiers is output.

1.8 Contributions

To address the problems mentioned in clinical and biomedical applications, I have developed and implemented an algorithm that segments images containing objects of different types, when there are underlying relations between the objects. Generating a segmentation of each of these objects in the image and finding their relationships are required for e.g. calculating metrics that determine the health or malignancy of cells [10] or to evaluate the immune system [11]. Moreover, such an algorithm can embed the prior knowledge, employed in the manual analysis, to improve the performance and reduce the training dependency and time.

The method developed in this thesis addresses the problem of quantifying cell images with objects of different types. The following points present the contributions of this method in the analysis of cell images and also summarize how the introduced problem is addressed.

- **Polytrees for modeling inter-object relationships in segmentation**: this is the first use of polytrees for the segmentation of multiple objects in images, when there are relationships between different types of objects. Trees have been widely used in segmenting and modeling simple relations between classes of objects (for example [38]). However, despite their ability to model more complex label configurations leading to a more accurate image segmentation, polytrees have been largely neglected in this area. Here, hierarchical polytree graphical models are employed that can model inter-object relationships during segmentation. These models pave the way for understanding and imposing relations between objects that can not be captured by other common graphical models such as trees.

  This model is useful in analyzing the immune response during infectious disease, as different types of cells exist in the images, i.e. host and pathogen cells. Understanding host-pathogen interactions can be used as useful information for disease diagnosis and treatment. Furthermore, different subcellular proteins can be delineated in the images and the physiological context for their function can be studied. Also, polytrees can model cell-cell relations which is useful in studying the biological processes where different types of cells are involved. On the other hand, looking at polytrees corresponding to successive times points, the evolution of the graph over time can be used for investigating cell movements which are the key characteristics in processes such as organ development and metastasis.

- **Closed-form solution for the polytree inference problem**: One possible reason for the limited use of polytrees for different applications has been the non-existence of a practical efficient algorithm for inferring labels on these structures. In this work, a two-pass inference algorithm has been derived that efficiently calculates exact posterior probability distributions for polytrees. The algorithm has been validated by classifying synthetic data and evaluated on the supervised segmentation of multiple image datasets, where it has performed favorably compared to the competing methods.

- **Weakly supervised segmentation using EM-polytrees**: Apart from the supervised application, using the Expectation-Maximization (EM) algorithm, polytree segmentation has been extended to be used in a weakly supervised setting. Using this method, the segmentation is performed with a significantly lower dependency on the training data. Furthermore, the model parameters are optimized while inferring labels of the nodes.

  This setting can be useful for clinical and biomedical research applications where there is limited or no access to the training data. A few number of training areas in the image are sufficient for the algorithm to calculate the initial parameter values and find the first set of labels based on them, at the same time respecting the imposed prior knowledge. As the algorithm iterates, estimations of the parameters and the calculated labels for the image are gradually improved.

- **Segmentation error prediction**: Generation of ground truth is time-consuming, error-prone and not always possible. With the vision of extending the proposed algorithm to be used interactively, a mechanism has been designed whereby the model nominates locations of a possible error in the segmented images. This has been done by assessing label configurations in the graphs corresponding to the segmented images and employing the calculated uncertainties in categorizing the labeled graph nodes.

  Performing an interactive segmentation allows starting from no training data and using small interactions from the user to improve the output segmentation. This can be useful when the algorithm is applied to new domains for which no previous method has been evaluated on.

The proposed methods in this thesis have been evaluated on the stained transmitted light microscopy and widefield fluorescence images. As it will be explained in chapters 3 and 4, the method consists of different steps including pre-processing (image oversegmentation feature extraction), graph generation, inference and segmented image reconstruction. Even though the experiments presented here have been on the two mentioned types of microscopy images, the proposed method can be applied to images acquired by other techniques, through altering the pre-processing and reconstruction steps.

1.9 Summary

This chapter expressed the data used and the problem addressed in this thesis and representative applications of microscopy imaging and analysis in disease diagnosis and treatment and biomedical research. A review of experimental systems for analyzing cells and methods for staining them was also provided. This background provides a practical vision of the research carried out in this project in pathological and biomedical applications. Furthermore, it gives an introduction to the terms that will be used in chapters 3 and 4 in describing the experiments on microscopy images. The next chapter more specifically discusses the need for segmentation of microscopy images and the methods have been used in the literature so far. It will also present a basis for the proposed graph-based method in the following chapters.

Chapter 3 introduces the use of polytrees for supervised cell segmentation in microscopy images. The details of polytree generation and the mathematics of the inference algorithm are presented and results of the method evaluation are provided. In chapter 4, the use of polytree has been extended to the weakly-supervised segmentation using the EM algorithm. The results presented in this chapter include those of sensitivity and convergence analysis, evaluation on segmentation of images of multiple datasets, image quantification, and segmentation error prediction. Finally, the last chapter discusses the work done of this thesis and possible future directions.

Chapter 2

Literature review

2.1 Need for multi-class segmentation

Cellular image analysis usually relies on a quantification of the image which is based on an image segmentation. This involves segmenting images that contain multiple classes of objects. For example, studying organ development [24] and the immune response during infection [1] require the identification of different cell types. On the other hand, some researchers work at the sub-cellular resolution where identification of organelles [22] and proteins [23] within cells are required.

Diagnosis, grading, and treatment of many diseases rely on extracting metrics from images, such as the proportion of cells during infection [11], differential counts in blood smears [10], and the ratio of abnormal nuclei shapes [39]. Manual analysis of microscopy images is arduous, time-consuming and prone to intra- and inter-subject variability. Moreover, the advances in microscopy techniques in taking large cohorts of images such as high throughput imaging [40] with higher resolutions, has made manual image analysis almost impossible. Therefore, efficient automatic methods for analyzing microscopy images are desired to deal with large volumes of data, which is also subject to high complexity and variance.

In performing a multi-class cell segmentation, existing information about the inter-object relations can improve the results. Examples would be the nucleus being restricted to be found within cells or sub-nuclear compartments expected to reside inside cell nuclei. Incorporating this information requires performing a constrained segmentation that eliminates implausible results that do not comply with the prior knowledge.

There is a wide range of segmentation methods that have been proposed for segmentation, including intensity thresholding as one of the first and simplest methods of segmentation [41, 42] and the watershed transform [43]. In the following, a brief review of the main segmentation techniques is firstly provided. Next, I provide an overview of methods for performing constrained segmentation. Graphical models are next introduced as tools for incorporating prior knowledge and the fundamentals of graph theory and inference algorithms on graphical models are presented. Finally, related works using graphical models for segmentation are reviewed.

2.2 Intensity thresholding

Thresholding is the first and simplest method for image segmentation. The procedure comprises dividing the pixels in the image into two groups of object and the background based on their intensity values being higher and lower than a threshold, respectively. Global image thresholding is sensitive to the value of the threshold and minor mistakes in the selection of the threshold value could result in drastic misclassifications. The threshold being set to lower values than the optimum causes the background pixels to be regarded as objects [44], while high values result in the truncation of object information with low intensities [45]. This has resulted in the popularity of manual thresholding and it remains the gold standard in certain applications [46]. Automatic thresholding techniques divide to the two categories of global and local thresholding, which are explained as follows.

One very famous global thresholding is the Otsus method [41] which determines the threshold by maximizing the between-class variance between the background and foreground pixels. In [42], the authors proposed an iterative algorithm that updates the threshold using average pixel intensities of each class. The algorithm is iterated until the difference of thresholds in two successive iterations is less than a certain value.

On the other hand, adaptive thresholding runs the segmentation process with various threshold values on the same image and then chooses the best value based on the results. This method has been widely used in the literature (e.g. [47]). Alternatively, some studies employ an aperture sliding on the image to make use of local threshold values for segmentation [48]. Even though these methods could outperform the former ones of Otsu and Ridler-Calvard in terms of robustness to noise and handling inhomogeneity, they provide a low performance in treating overlapping cells. Intensity thresholding techniques have been reviewed in [49, 50].

2.3 Watershed methods

At the expense of more exhaustive steps for processing, the watershed method [51] has helped to take care of the overlapped cells in images. Starting from local minima of intensity in the image, the watershed algorithm gradually sinks areas of the image, until these so-called basins meet. The algorithm then separates individual objects by inserting borders

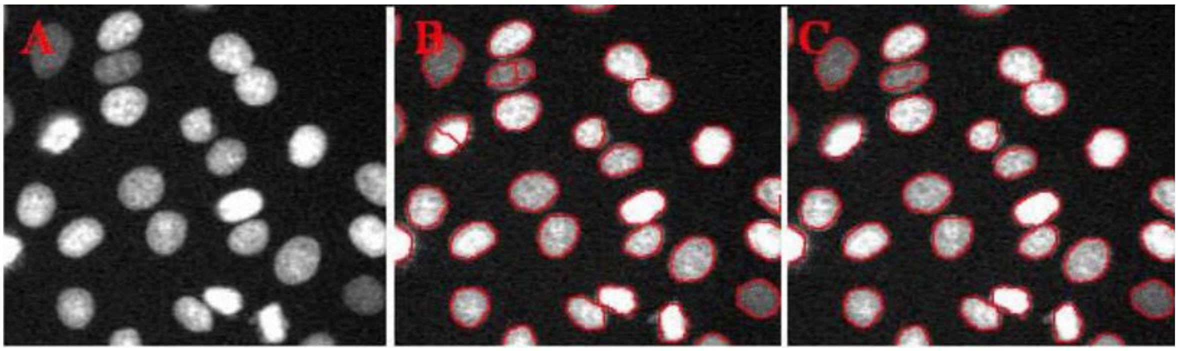


Figure 2.1: Segmentation results using Zhou *et al.* [52] method. A) Original Image. B) Segmentation results using the watershed algorithm [52]. C) Results after eliminating oversegmentation using fragment merging algorithm. Image has been adapted from [52].

at confluence locations. Zhou *et al.* [52] used adaptive thresholding for the segmentation of cell nuclei and employed the watershed algorithm to take care of the overlapping cells. They then corrected the oversegmentation in the output of the watershed algorithm by use of a fragment merging method. Using several features, the implemented fragment merging determines the quality of segmentation and merges the neighboring detected nuclei in cases where merging results in improvements of the desired features. Sample segmentation results of this method are shown in Figure 2.1.

Another approach to address the oversegmentation problem in the use of watershed methods is to merge the falsely segmented regions into components in the real objects in the image, based on a predefined constraint. This problem, as has been reported in the analysis of challenging cell images in [53], has been addressed by developing a statistical object model in [54] that merges the undesired regions. A multivariate Gaussian function is employed to model the nucleus distribution, that calculates a merging decision criterion. The algorithm merges the undesired regions, for which the resulting segment can provide a higher measurement score than the single region before merging with a threshold-based decision.

2.4 Blob detection methods

As a preliminary step to segmentation, blob detection methods can locate the cells, especially when a large population of cells exists in the image, by approximating them with small ellipses. The most popular method for this is the Laplacian of Gaussian (LoG) [55]. The algorithm consists of a convolution with a Gaussian distribution of the form:

$$G(x, y; \sigma) = e^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{2.1}$$

where σ (the standard deviation) is considered as a scale parameter to produce different representations of the image for detection of blobs of various sizes. This operation is done to smooth the image and remove noise from it and is followed by a Laplacian operator ($\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$) which is aimed at the detection of edges by finding instances of sudden variations. The idea used to simplify the implementation of this method is to first apply the Laplacian to the Gaussian and then convolve the Laplacian of the Gaussian with the image [55].

As a popular method for cell detection and especially for pre-processing of images, e.g. to estimate sizes and locations of cells, numerous improvements have been suggested to the LoG method. Kong *et al.* [56] have proposed a generalized version of the algorithm to address the limitation of detection of symmetric cells by using different LoG kernel operators to broaden the application of the method. For this reason, they have generated several kernels by modifying parameters regarding orientation and scale, of 2-D Gaussians, both isotropically and anisotropically. The scales of filters were then normalized and used for detecting nuclei from images. Finally, a marker-controlled watershed was used to split the nuclei. Results of applying the method are shown in Figure 2.2. In a more recent study, the authors of Zhang *et al.* [57] suggested a Hessian-based version of LoG to use the Hessian matrix for improving the detection results and reducing the computational cost. There are also other versions of blob detection such as Difference of Gaussians (DoG) [55]. In a similar approach to LoG, DoG provides an approximation of LoG, with the advantage of being more efficient. Despite the promising progress, blob detection methods are usually unable to capture deformable objects as they are not flexible enough. Even though, they might be well used as tools for nominating candidates of regions for possible detection of cells.

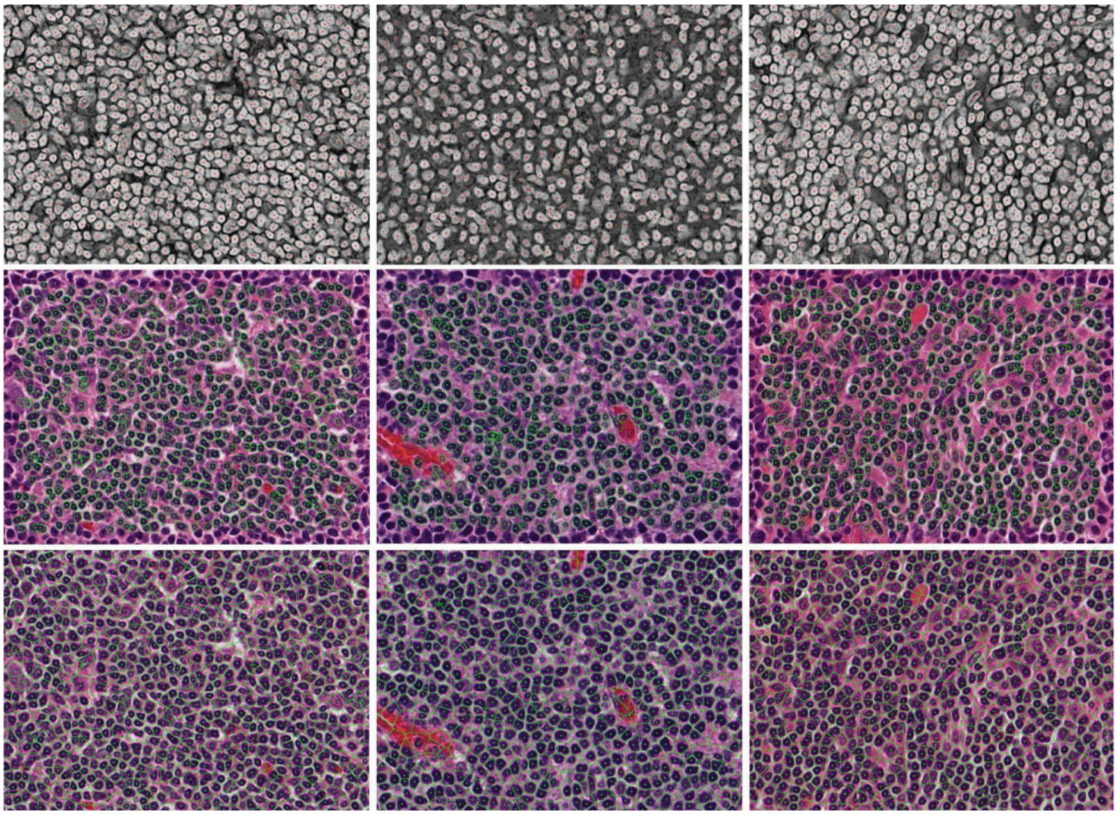


Figure 2.2: Detection of nuclei in pathological images (image adapted from Kong *et al.* [56]). The first row shows the detection of nuclei centers and the second row shows the detection of blobs using nuclei centers. The third row shows nuclei split using the watershed method.

2.5 Shape-based analysis

There are other segmentation methods taking into account the shapes of the objects, such as template matching, which have been widely used for the analysis of biomedical images [58]. Simply, template matching searches for instances of a template image within a larger image. In the early usages of this method, only simple transformations on the shapes including translation, rotation and scaling were considered [59], which were referred to as rigid shape matching. Even though considering shapes of objects in addition to merely investigating intensity values can improve the segmentation results, rigid template matching suffers from relying on a geometrically exact template which does not necessarily exist in cases where the objects shapes have high variations [60]. To address this limitation, deformable methods are proposed as flexible alternatives to cope with shape deformations.

Deformable models are used for segmentation as parametric models, which are optimized through iteration with the aim of minimizing an energy function. These models could be formulated explicitly, where object boundaries are expressed through a parametric contour in 2D or a parametric shape in 3D, or implicitly, as zero levels of a function which have one more dimension than that of the image. Level sets are also favorable for their use in the tracking phase, since they can keep track of morphological changes.

As members of deformable models, active contours, or Snakes, originally invented by Kass *et al.* [61] are based on Bayesian estimation method which consists of a prior and a likelihood. Assuming a set Γ to contain edge pixels, for each pixel x in the image domain Ω, we have

$$\gamma(x) = \begin{cases} 1 & x \text{ is an edge pixel} \\ 2 & \text{otherwise} \end{cases}, \Gamma = \gamma^{-1}(1) \tag{2.2}$$

The likelihood term (also called the generative model) $p(u|\Gamma)$, for image u defined on the same domain Ω, is to model image u when accurate information about the edge Γ is known. An exponential function is then defined on variables x, u and derivatives of image u, to best model the generative model. For a given image u, finding the edge set Γ, in a Bayesian framework, is to equivalently maximize the following posterior.

$$\max_{\Gamma} p(\Gamma|u) = \frac{p(u|\Gamma)p(\Gamma)}{p(u)}, \tag{2.3}$$

where $p(u)$ is considered as a normalization constant for the given image. Using Gibbs energy $E = \ln p$, the above formula results in the following energy minimization:

$$E[\Gamma[\Gamma|u] = E[u|\Gamma] + E[\Gamma] \tag{2.4}$$

$E[\Gamma]$ forces the extracted edges to be regular and visually acceptable. Brownian paths, Sobolev edges, length energy, and Euler's Elastica are some of the options in the definition of the energy for the edge [62]. The above method for finding edges is known as edge-based active contours. However, as determining the boundaries of the objects (x where $\gamma(x) = 1$) is equivalent to determination of regions of the objects surrounded by boundaries (x where $\gamma(x) = 0$), there are other groups of active contours, the likelihood models of which are based on regions of objects, called region-based active contours.

Active contours and level sets are frequently used in cellular image segmentation. These are capable of segmenting cells with low-intensity cell boundaries and also being used with cell shape models incorporated. Level sets are suitable for segmentation of large numbers of deformable objects, such as cells. However, one problem with these is the initial contour formation, which in many cases leads to the need for a detection phase.

Level sets are known as ideal methods of segmentation in cases where intensities of the objects which are subject to segmentation are the same. These methods could handle the segmentation for a large number of cells and at the presence of imaging artifacts [63]. However, they lack maintaining separated contours for distinct cells, when they overlap. Zhang *et al.* [64] proposed a method for cell segmentation, using one contour for each cell and defining constraints on them to prevent them from merging. To overcome the computational cost of individual level sets for each cell, Nath *et al.* [65] proposed a method to segment large numbers of cells defining only four level sets. In general, deformable models, with their ability to account for shape models and also the capability of extension to tracking phase seem to be promising for use in the analysis of cells.

2.6 Learning-based methods

Machine learning methods have been widely used in cell analysis literature. These methods learn the regularities present in the annotated data, called training data, and use these regularities to classify the data into different categories [66].

Arteta *et al.* [67] proposed an algorithm that first detects candidate regions by recognizing the corresponding elements in the images which was previously suggested by Matas *et al.* [68]. Arteta's algorithm works based on finding intensity maxima and minima in the image by sweeping the threshold, to be scored with a learning-based measure. This measure, chosen to assume a non-overlap constraint, uses structured Support Vector Machines (SVM) learning in the inference procedure from the training set. However, the method, like many other similar works in the literature, recognizes the nuclei rather than the whole cell.

Deep learning methods, like Convolutional Neural Networks (CNNs) [69] have performed well in cell segmentation [70, 71] and in image classification and scene labeling [72, 73]. These networks have the advantage that the features are automatically learned through kernel matrices in the convolution layers, which relieves the need for selection and training of the features [69, 74]. A CNN is usually composed of successive pairs of convolutional and pooling layers, followed by a number of fully connected layers. A convolutional layer consists of convolutional filters that (each corresponding to one output feature map) are used to calculate the output feature maps. The pooling (or max-pooling) layers function is to summarize the actions and to choose the maximum values over a neighborhood region in each feature maps. The fully connected layer learns higher level features representations. The last layer is often a softmax layer which is also a fully connected layer and calculates the posteriors of belonging to each of the object classes for each pixel in the image [75]. As an example of pixel-wise segmentation using a deep CNN-based method, Ciregan *et al.* [76] performed a segmentation of neuronal membranes in electron microscopy images. To improve the results, multiple networks are trained and the final prediction is achieved by averaging the outputs of the networks. Even though the performance is better than a single network, the segmentation is done using a sliding window, which might be time-inefficient in analyzing large images. In an application of this method for the detection of mitotic nuclei [77], the authors trained a neural network with the image patches centered at mitosis instances in the image. The trained algorithm was then used to find and detect mitotic nuclei in the images.

Badrinarayanan *et al.* [78] have proposed an encoder-decoder architecture for semantic segmentation, that reduces the computationality compared to the U-Net [79]. SegNet [78] has introduced new decoders for upsampling the lower resolution feature maps that use pooling indices computed in the max-pooling step of the corresponding encoder. Using this technique removes the need for learning to upsample and was shown to have a more efficient use of the memory.

Other pattern recognition methods, such as K-means and Expectation-Maximization (EM) algorithms, have been used for cell analysis as well. K-means [80] is a clustering algorithm that assigns each data sample (which are pixels in image analysis) to a cluster with its mean closest to that data sample. Iterative updates are then made on means of clusters, and consequently, their assigned members, to minimize sums of square distances between data samples and cluster means [66]. EM algorithm could also be considered as a similar, yet more powerful method that can maximize maximum likelihood estimators. EM is an iterative algorithm to optimize an energy function. The parameters of the energy function start from an initial first guess. In the expectation step, the value of the energy function is calculated using the current estimate of the variables. Then in the maximization step, the parameters are updated in a way to maximize the gradient of the energy function. The algorithm is iterated until a predefined convergence criterion is satisfied. Sinha and Ramakrishnan [81] make use of K-means for locating nuclei in the images and cropping the cell by a rectangular region. EM algorithm is then employed for refinement of model parameters of final segmentation of white blood cells from the background. A spatially-constrained version of EM algorithm is used by James Monaco [82] to incorporate Markov priors, with their novelty in reducing the computational cost.

The broad application of segmentation methods in medicine and biology has urged the literature to include reviews based on the application of segmentation, for example in the segmentation of MRI images of heart [83], multiple sclerosis [84] and brain [85]. In cell segmentation, reviews have been prepared for CAD systems in histology image analysis [5, 86, 87], breast cancer histopathology [88], computational pathology [89, 90] and on nucleus segmentation [4, 91].

Different methods addressing the segmentation problem have been improved over time to provide a higher accuracy and efficiency. This improvement has been predominantly made at the cost of suggesting more complex methods that require stronger computational resources. As an instance, convolutional neural networks can extract the patterns existing in the training data to be used for the analysis of new images in the test sets. Apart from their computational complexity, these models rely on training sets with sufficient quantity and variation. Other techniques can provide acceptably accurate models for the objects in the image, but are limited in the number of objects that exist in each single image. Inspired by the manual annotation that is done by human experts, employing rules that are based on the prior knowledge available for the problem can improve the results and reduce the reliance on the training data. To this end, ways for incorporating prior knowledge in the segmentation are reviewed in the following.

2.7 Use of prior knowledge in segmentation

The overall characteristics of the region of interest are usually known in medical imaging. This prior knowledge includes, but is not limited to, shape information, such as the brain being spherical, tree-like shapes of dendritic cells and the heart being cone-shaped, or spatial patterns of anatomical structures relative to one another, for example nuclei being inside cytoplasms in cells and amygdala being anterior and superior to the hippocampus. Many biological and medical applications require prior information to be respected in segmentation as they critically rely on the accuracy of segmentation. On the other hand, many disorders are associated with subtle changes that can modulate the imaging or anatomical properties of organs. Clinical and research approaches that deal with these disorders rely on the precision of the segmentations [92, 93]. Even though many methods have been proposed for performing constrained segmentation, it remains a challenging problem.

Segmentation methods that employ texture or intensity are sensitive to imaging artifacts including non-uniformity, geometric distortions, noise and subject motion [94]. Integration of constraints increases the complexity of the segmentation and cannot always fully model the problem. However, since unconstrained segmentations often contain deviations from the true characteristics of the objects of interest, constrained segmentation techniques are desirable.

2.7.1 Examples of prior-knowledge-based segmentation

Incorporating *a priori* information into segmentation improves the accuracy by eliminating semantically unfeasible outputs. Various types of prior information have been employed in segmentation by enforcing shape, appearance, adjacency and inclusion constraints. Shape constraints have been enforced through the classic approach of deforming a model to a target structure by regularizing deformable contours [95–97], or by fitting the current segmentation to a template to enforce the underlying structures [98,99]. Graph-cuts were also employed to improve segmentation by imposing shape [100, 101], ordering [102, 103] and nesting [104] priors.

In multi-class image analysis, relations between different classes can work as prior knowledge to improve the segmentation. Inter-object dependencies have been used in the segmentation of interacting objects in the knee joint [105] and enhancing the segmentation of cell organelles [106]. Other examples in cell segmentation include using priors to consider the shape [107] and relative topology of cells and nuclei [108, 109], and to impose area and size constraints on segmented regions [110], to achieve a better segmentation. In brain tissue analysis, appearance and spatial priors have been used to improve tumor localization [111], generalization to unseen images [112], and lesion recognition as atypical brain tissues [113].

In the context of neural networks, shape priors have been imposed through Shape Boltzmann Machines (SBM) [114, 115]. Successive works enforced topology and geometric constraints [116, 117] in fully convolutional networks (FCNs). Cascaded Convolutional neural networks (CNNs) [78] were also shown to learn priors using more complex models and with increasing computational expense [118, 119]. However, these require a training of a large set of parameters, which may result in intractable training or in inaccurate segmentation.

On the other hand, interactive segmentation systems were developed that consider weaker priors during segmentation, but detect inconsistencies in the segmented images and aim to correct them [120, 121]. These methods utilize user input to encode the implications of the prior knowledge through energy terms in their cost functions [122, 123].

Inter-class relations have been employed in segmentation through discriminative and generative probabilistic models. CNNs are an example of discriminative models that can learn dependencies between different classes through their cascade network architecture [118, 124]. These models perform well, and rely on a set of training samples, with a sufficient quantity and variance, to learn the underlying constraints. On the other hand, generative models, such as probabilistic graphical models, impose the interrelations by utilizing a prior term designed based on the problem. Incorporation of prior knowledge for deriving model posteriors reduces the dependency of graphical models on the training data, compared to CNNs. The focus of this thesis is on the generative graphical models for the incorporation of prior knowledge while performing image segmentation. The following section reviews the fundamentals of graph theory that will be used in the following chapters.

2.8 Graphical models for modeling associative relationships

Probabilistic graphical models provide a visual representation of probability distributions made up of **nodes** (vertices) connected by **edges** (links). Nodes represent the random variables of the model with edges showing their probabilistic relations. Graphs facilitate the expansion of the joint probability distribution into factors each depending on a subset of random variables. Given the visual representation of graphs, properties of the probabilistic model, such as conditional dependencies, can be inspected.

Graphical models enable modeling associative relations between objects to improve segmentation [125–127]. They provide a simple way to visualize probabilistic models by expressing random variables as graph nodes and defining the relationships between them through graph edges [66]. The key aspect of these models is that the label of each node is determined based on both its own attributes as well as attributes of other nodes connected to it through graph edges. This way, not only all the relevant information is incorporated in inferring the labels, but also label configuration constraints can be effectively projected on the model and be used during the inference. For instance, Chen *et al.* [128] employed graphical models to incorporate nuclear positions with boundary information for yeast cell segmentation. In a broader context of application, Uzunbas *et al.* [129] used an interactive graphical-model-based approach that seeks user's input for nodes with uncertain labels and learns from mislabelled nodes. Another instance is the use of graphical models for combining appearance models with shape priors for retinal segmentation [130]. Besides their application in image segmentation, graphical models have been used for modeling the web structure [131, 132], social systems [133], and biological networks [134], indicating the applicability of these models to a wider range of subjects.

2.8.1 Graph theory

In the following, a short overview of the basic definitions in graph theory [66, 135] is provided, that will be used in the rest of the thesis.

A graph $\mathcal{G}(V, E)$ comprises a set of nodes $V = \{s_1, s_2, ..., s_N\}$ connected through a set of links $E \subset V \times V$. Each edge (s_1, s_2) between nodes $s_1, s_2 \in \mathcal{G}$ may be **undirected**, where we call s_1 and s_2 **neighbors** in the graph. Alternatively, if the edge between the nodes is **directed** $(s_1 \to s_2)$, s_1 is called the **parent** of the **child** s_2. If a node on a directed graph does not have a parent, it is called an **orphan** node.

A sequence of nodes $\{s_1, s_2, ..., s_m\}$ define a **path** on the graph, where $(s_i, s_{i+1}) \in E$ or $(s_i \to s_{i+1}) \in E$, for each $i \in \{1, ..., m-1\}$. If all the edges are directed in a path, it is called a **directed path**. If the directions of edges are ignored for a path, we call it a **trail**. A directed path with $s_1 = s_m$ is called a **cycle** and a trail with $s_1 = s_m$ is called a **loop**.

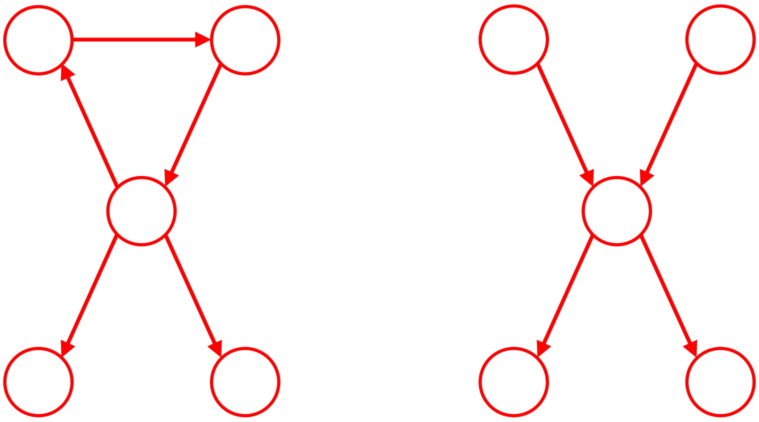
A directed graph without cycles is called a **directed acyclic graph (DAG)**. Figure 2.3 shows examples of cyclic and acyclic directed graphs. If there is only one trail between every pair of nodes in a DAG, it is called a **polytree**. If all nodes, except the source node, have exactly one parent node, the graph is called a **tree**. Alternatively, an undirected graph without loops is called a tree. Figure 2.4 shows samples from polytrees, directed and undirected trees. A graph $\mathcal{G}'(V', E')$ is called a subgraph of graph $\mathcal{G}(V, E)$, if $V' \subset V$, $E' \subset E$ and $E' = \{(s_i, s_j) : s_i \in V', s_j \in V'\}$.

A **probabilistic graphical model** corresponds to the tuple $(\mathcal{G}, P)$ of a graph $\mathcal{G} = (V, E)$ and a probability distribution P. Each node s_i corresponds to a random variable x_i (which can be discrete or continuous), that can be associated with an observation vector $\boldsymbol{y}_i$. Existence of graph edges indicate conditional dependencies on $\mathcal{G}$ that facilitate a factorization of P. Given the three variables a, b and c, the conditional distribution of a and b, given c is written as

$$p(a, b|c) = p(a|b, c)p(b|c). \tag{2.5}$$

If the distribution of a does not depend on that of b, given c,

$$p(a|b, c) = p(a|c), \tag{2.6}$$



(a)

(b)

Figure 2.3: Cyclic (a) and acyclic (b) directed graphical models.

or based on Equation 2.5,

$$p(a, b|c) = p(a|c)p(b|c). \tag{2.7}$$

In this case, a is **conditionally independent** of b, given c, denoted by $a \perp\!\!\!\perp b|c$.

Two types of graphical models have been used for image segmentation. Markov Random Fields have weighted edges indicating the degree of dependencies between variables. On the other hand, Bayesian Networks (BNs) have directed edges explicitly representing the dependence of random variables on each other. Directed graphs can have exact solutions if they are restricted to having no directed cycles. The two types of graphs are explained in the following sections.

2.8.2 Undirected graphical models

An undirected graphical model or a Markov random field [136] is a tuple $(\mathcal{G}, P)$ composed of undirected edges (s, t), where $\mathcal{G}(V, E)$ is an undirected graph with N nodes and P is a probability distribution. Each two nodes s_i and s_j are independent, if they are not connected

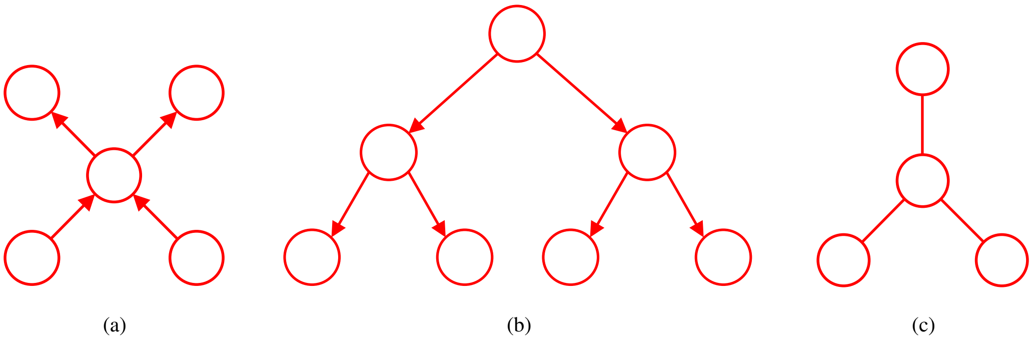


Figure 2.4: Polytree (a), directed (b) and undirected (c) trees.

by a link and all passes connecting the two nodes contain observed nodes. Therefore, the factorization of the joint probability should contain factors in which s_i and s_j do not appear in the same factor.

To write a factorization for the undirected graph, the joint probability is written as a product of functions over subsets of variables that are local to the graph. We call a subset of the undirected graph with links between every two nodes of it a **clique**. If a clique is such that no more links from the graph can be added to it, so that it still remains a clique, we call it a **maximal clique** C.

Denoting the variables in the clique C by $\boldsymbol{x}_C$, the joint probability distribution factorizes into a product of **potential functions** $\psi_C(\boldsymbol{x}_C)$ for the maximal cliques

$$p(\mathbb{X}) = \frac{1}{Z} \prod_C \psi_C(\boldsymbol{x}_C), \tag{2.8}$$

where Z is called the **partition function** and is a normalization constant

$$Z = \sum_{\boldsymbol{x}_C} \psi_C(\boldsymbol{x}_C), \tag{2.9}$$

and $\mathbb{X} = \{x_s\}$ is the set of all random variables in the graph ($s = 1, ..., N$).

Potential functions have no interpretations in terms of the marginal distributions in MRFs, however, as it will be explained in section 2.8.3, the joint probability in DAGs is factorized into the conditional probabilities on the graph.

2.8.3 Directed graphical models (BNs)

A directed graphical model or a Bayesian Network (BN) is a tuple $(\mathcal{G}, P)$, where $\mathcal{G}(V, E)$ is a directed graph containing N nodes with directed edges $s \rightarrow t$ between them, and P is a probability distribution. Directed links in BNs are useful for representing causal relationships between random variables, for example when an event is a result of another event. The conditional independence of nodes on the directed graph is explained in section 3.3.5.

The joint distribution on the directed graph is defined by the product of the conditional distributions of all nodes on the graph conditioned on their parents. Therefore, the factorization of the joint probability for the directed graph $\mathcal{G}(V, E)$, with the set of random variables denoted by $\mathbb{X} = \{x_s\}$ for $s = 1, ..., N$, takes the form

$$p(X) = \prod_{s \in \mathcal{G} \setminus \mathcal{O}} p(x_s | pa_s) \prod_{s \in \mathcal{O}} p(x_s), \tag{2.10}$$

where $\mathcal{O}$ is the set of orphan nodes, $\mathcal{G} \setminus \mathcal{O}$ is the rest of the nodes in the graph and pa_s is the set of parents of x_s. It can be easily shown from Equation 2.10 that $p(\mathbb{X})$ is normalized as each of the factors in the product is normalized.

2.9 Inference in graphical models

Inference in graphical models refers to finding the probability distributions for a set of random variables x_s, when a set of nodes are clamped to observation nodes $\boldsymbol{y}_s$ from the set of all observations. These probability distributions $p(x_s|\boldsymbol{y}_s)$ are called **posterior distributions**. Once the posterior distributions are found, many applications, require the value of the random variable $\hat{x}_s$ that maximizes the posterior probability

$$\hat{x}_s = \arg\max_{x_s \in \Lambda} p(x_s|\mathbb{Y}), \tag{2.11}$$

where Λ is the set of all K labels x_s can take. For example, in image segmentation, given the image, different features are extracted from the image that make the feature vectors stored in $\boldsymbol{y}_s$ nodes. Finding the optimal segmentation in this problem is equivalent to finding the most probable labels $\hat{x}_s$ in the posterior distributions, based on which pixels (or groups of pixels) are labeled as the objects in the image.

Graphical models facilitate devising efficient inference algorithms which can be exact or approximate both relying on a factorization of the joint probability. In the following, an overview of inference algorithms on graphical models is presented.

2.9.1 Graph-cuts

Graph-cuts are algorithms in which a weighted undirected graph $\mathcal{G}(V, E)$ is made based on the image, where V and E show the sets of nodes corresponding to the image pixels and weighted edges, respectively. Weights of the edges are calculated based on a similarity function $W : V^2 \rightarrow \mathbb{R}^+$, that measures the similarity of pixels according to their features, such as intensity value and spatial proximity. The graph-cut finds the optimal cut that divides the graph into disjoint subgraphs with a high similarity within the subgraphs and a low similarity between them. Denoting the subgraphs by $\mathcal{A}$ and $\mathcal{B}$, one can calculate the dissimilarity between the subgraphs by summing edges weights that are to be removed for dividing the graph to subgraphs $\mathcal{A}(V_{\mathcal{A}}, E_{\mathcal{A}})$ and $\mathcal{B}(V_{\mathcal{B}}, E_{\mathcal{B}})$.

$$cut(\mathcal{A}, \mathcal{B}) = \sum_{u \in V_{\mathcal{A}}, v \in V_{\mathcal{B}}} w(u, v). \tag{2.12}$$

Finding the optimal partitioning of the graph is equivalent to finding the minimum cut. However, the minimum cut favors cutting small isolated regions [137] in the graph. This is because the cut in Equation 2.12 increases with the number of edges connecting the two subgraphs. Figure 2.5 shows an example of such a case, where cuts that separate nodes $n1$ and $n2$ have minimum values, while cuts separating the right and left sides of the graph have higher values. To avoid this, another measure of dissimilarity is defined as normalized cut [138],

$$Ncut(\mathcal{A}, \mathcal{B}) = \frac{cut(\mathcal{A}, \mathcal{B})}{\sum_{u \in V_{\mathcal{A}}, t \in V} w(u, t)} + \frac{cut(\mathcal{A}, \mathcal{B})}{\sum_{v \in V_{\mathcal{B}}, t \in V} w(v, t)}. \tag{2.13}$$

The normalized cut computes the cut cost as a fraction of the total edge connections to all graph nodes. Using this measure, cuts partitioning small isolated points will not have small *Ncut* values, as the cut contains a large fraction of the total connections of the set with the

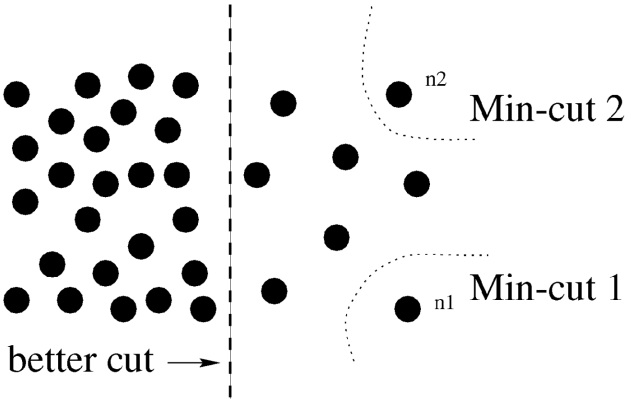


Figure 2.5: An example case for a non-optimal solution of graph-cut [137].

graph.

2.9.2 Message passing

Finding the posterior distribution is done through marginalizing the joint probability distribution which is known as variable elimination [139]. The inference algorithms which are based on distributivity and commutativity rules, such as max-product and max-sum [66], are examples of the variable elimination tasks. The efficiency of variable elimination depends on the order of elimination of variables which itself is dependent on the structure of the graph. Employing the principle of variable elimination has led to the emergence of a type of inference called message passing [140].

Message passing breaks down a summation over a set of random variables $\mathbb{A}$ into subproblems with summations over smaller sets $\mathbb{A}_i \subset \mathbb{A}$. Message passing inference algorithms pave the way for solving inference problems through dynamic programming. Since a subset of the messages are used in the calculation of posteriors of several nodes, the messages are stored in the memory once calculated and used in the other inference expansions to increase the algorithm's efficiency.

Popular examples of message passing are sum-product and max-product algorithms that are exact on tree-shaped graphs. A special case of the sum-product algorithm is called the belief propagation algorithm [140], which performs exact inference on directed graphs with-

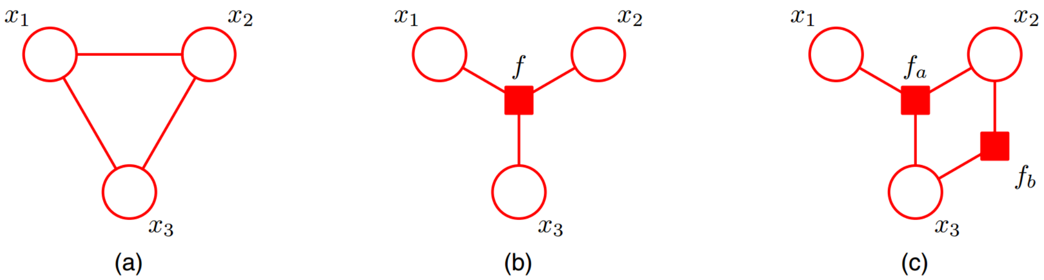


Figure 2.6: Factor graphs are generated by creating a graph with nodes that correspond to the random variables and creating additional factor nodes corresponding to the maximal cliques. Panel (a) shows an example undirected graph with the clique potential $\psi(x_1, x_2, x_3)$. Panels (b) and (c) show two factor graphs with $f(x_1, x_2, x_3) = \psi(x_1, x_2, x_3)$ and $f_a(x_1, x_2, x_3) f_b(x_1, x_2) = \psi(x_1, x_2, x_3)$, corresponding to (a). Picture adapted from [66].

out loops. The max-product algorithm finds the value of the random variable for which the probability distribution is maximum and finds the value of that probability. Even though, it does not guarantee to find the global maximum of the posteriors.

Using the sum-product algorithms on graphs is done through the generation of a new graph based on the given graphical model, called the factor graph [141]. These graphs make an explicit representation of the joint probability factorization by introducing new nodes in the graph that correspond to the factors of the joint probability distribution, in addition to the nodes for the random variables. Figure 2.6 shows two possible factor graphs for an undirected graphical model.

Factor graphs provide a straightforward representation for the factorization of the joint probability distribution, at the expense of generating an additional graph which increases the computations in the inference.

Although the sum-product algorithm calculates exact posteriors for tree-shaped graphs, in many applications the graphs are not tree-shaped so that exact methods cannot be employed. To address the inference on arbitrary graphs with cycles, an extension of the sum-product algorithm is proposed which is called loopy belief propagation [142]. Loopy Belief Propagation is a heuristic algorithm based on message passing on trees and is the most common approximation algorithm for general graphs. Since this algorithm does not guarantee convergence, a number of extensions have been made to the algorithm to provide better approximations and convergence properties [143, 144].

To calculate exact posteriors on arbitrary graphs, the junction tree algorithm [145] has been proposed that is based on the sum-product algorithm and works on a conversion of the input graph. This algorithm guarantees the cheapest computational cost among the exact inference algorithms for the graph. However, since it requires to work with the joint probability distribution, its practicality is determined based on the treewidth parameter of the graph, which is defined as the size of the largest clique on the graph. The junction tree algorithm becomes impractical when the treewidth takes large values.

As exact solutions either do not exist or are intractable for a set of inference problems, approximate methods have been proposed for the inference which include the two main categories of variational and sampling methods. Variational inference methods employ a distribution to approximate the original complex graphical model, such that an efficient inference algorithm exists for it [127]. A variational approximation algorithm chooses the best approximation of a probability distribution from the set of all tractable distributions for inference. On the other hand, sampling methods employ stochastic algorithms to generate unbiased samples from the distribution. Monte-Carlo Markov Chains [146] and Gibbs sampling [147] are methods that perform the inference through sampling techniques.

This thesis focuses on methods for exact inference on graphical models. In the next section, message passing is employed to provide exact solutions for two common inference problems in graphical models, i.e. chains and trees.

2.9.3 Exact inference in graphs without loops

In this section, exact inference on a chain of nodes is presented. Then, the sum-product algorithm is explained with its application to the tree-structured graphical models.

For the chain in Figure 2.7, the joint probability is written as follows

$$p(X) = \frac{1}{Z}\psi_{1,2}(x_1, x_2)\psi_{2,3}(x_2, x_3)...\psi_{N-1,N}(x_{N-1}, x_N), \tag{2.14}$$

consisting of the potential functions $\psi_{n-1,n}(x_{n-1}, x_n)$.

To solve the inference problem of finding the marginal distribution $p(x_n)$ for a node $x_n \in \{x_1, x_2, ..., x_N\}$ on the chain, the joint distribution should be summed over all of the variables

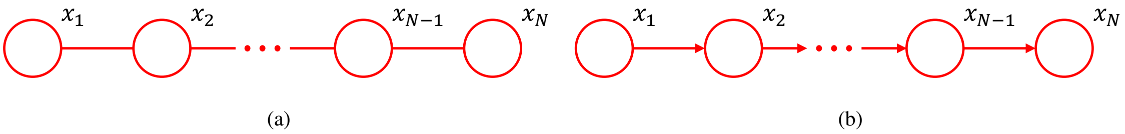


Figure 2.7: Undirected (a) and directed (b) chains of nodes.

except x_n

$$\begin{aligned} p(x_n) &= \sum_{\mathbb{X}\backslash x_n} p(X) \\ &= \sum_{x_1}\sum_{x_2}...\sum_{x_{n-1}}\sum_{x_{n+1}}...\sum_{x_N} \psi_{1,2}(x_1,x_2)\psi_{2,3}(x_2,x_3)...\psi_{N-1,N}(x_{N-1},x_N). \end{aligned} \tag{2.15}$$

The inference problem can be solved more efficiently by rearranging the summations and multiplications to the following form

$$\begin{aligned} p(x_n) =&\Big[\sum_{x_{n-1}}\psi_{n,n-1}(x_n,x_{n-1})...[\sum_{x_2}\psi_{2,3}(x_2,x_3)[\sum_{x_1}\psi_{1,2}(x_1,x_2)]]...\Big] \\ &\Big[\sum_{x_{n+1}}\psi_{n,n+1}(x_n,x_{n+1})...[\sum_{x_N}\psi_{N-1,N}(x_{N-1},x_N)]\Big]. \end{aligned} \tag{2.16}$$

Each of the factors $\Big[\sum_{x_{n-1}}\psi_{n,n-1}(x_n,x_{n-1})...[\sum_{x_2}\psi_{2,3}(x_2,x_3)[\sum_{x_1}\psi_{1,2}(x_1,x_2)]]...\Big]$ and $\Big[\sum_{x_{n+1}}\psi_{n,n+1}(x_n,x_{n+1})...[\sum_{x_N}\psi_{N-1,N}(x_{N-1},x_N)]...\Big]$ in Equation 2.16 can be interpreted as a **message** on the graph **passed** to the node x_n.

Note that these messages are results of marginalization of the joint probability over all the variables except x_n and are therefore merely functions of x_n. Denoting these factors by $\mu_a(x_n)$ and $\mu_b(x_n)$ respectively, the marginal probability distribution is given in the form

$$p(x_n) = \frac{1}{Z}\mu_a(x_n)\mu_b(x_n). \tag{2.17}$$

The inference problem presented above is solved for the undirected chain. For the case of the directed chain shown in Figure 2.7b, the potential functions will be replaced with the conditional probabilities

$$\begin{aligned} \psi_{1,2}(x_1, x_2) =& p(x_1)p(x_2|x_1) \\ \psi_{2,3}(x_2, x_3) =& p(x_3|x_2) \\ &\vdots \\ \psi_{N-1,N}(x_{N-1}, x_N) =& p(x_N|x_{N-1}), \end{aligned} \tag{2.18}$$

and the same inference in Equation 2.16 calculates the marginal probability of x_n.

This way of exact posterior calculation is useful for use in the Expectation step of the EM algorithm [148], where efficient inference algorithms can increase the overall efficiency of the iterative EM. This algorithm is explained in section 4.3.4.

The exact inference on chains shows the efficient use of message passing algorithm. This algorithm can be extended to the tree-shaped graphs through the sum-product algorithm. I now present the sum-product algorithm for tree graphical models.

For a tree graph $T = (V, E)$ the factorization of the joint probability can be written as

$$p(X) = \frac{1}{Z} \prod_{s \in V} \psi_s(x_s) \prod_{(s,t) \in E} \psi_{s,t}(x_s, x_t). \tag{2.19}$$

Similar to the inference on chains, the aim is to find the marginal distribution for an arbitrary node $s \in V$. The set of nodes in the neighborhood of s is denoted by $ne(s)$. For each node $t \in ne(s)$, a subgraph $T_t = (V_t, E_t)$ is defined such that all of its nodes can be reached without passing over the node s. Each of such subgraphs is also a tree, as depicted in the example shown in Figure 2.8.

The probability distribution for each subgraph T_t is written as

$$p(X_t) = \frac{1}{Z_t} \prod_{u \in V_t} \psi_u(x_u) \prod_{(u,w) \in E_t} \psi_{u,w}(x_u, x_w), \tag{2.20}$$

where Z_t is the normalization coefficient for the subgraph t.

Having calculated the distributions on each subgraph, one can find the marginal distribu-

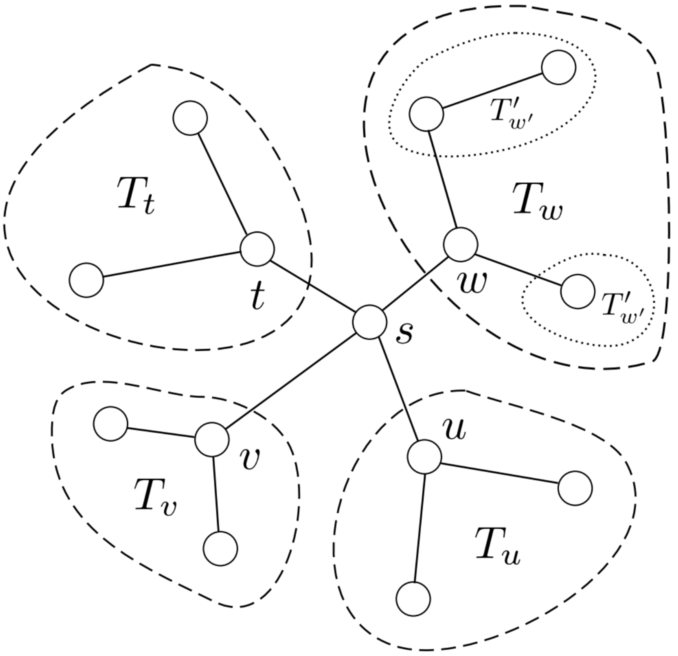


Figure 2.8: A tree decomposed into subtrees rooted at the node s. The illustration has been adapted from [127].

tion for node s

$$\begin{aligned} p(x_s) &= \sum_{x_t \neq x_s} p(X) \\ &= \sum_{x_t \neq x_s} \alpha.\psi_s(x_s) \prod_{t \in ne(s)} \psi_{s,t}(x_s, x_t) p(X_t) \\ &= \alpha.\psi_s(x_s) \prod_{t \in ne(s)} \sum_{x_u \in V_t} \psi_{s,u}(x_s, x_u) p(X_u), \end{aligned} \tag{2.21}$$

where α is the normalization coefficient.

Each subgraph can in turn, be decomposed into subgraphs, as shown in Figure 2.8 for T_w. Hence the marginal posterior calculation is recursively broken into subproblems until leaf nodes, i.e. nodes with only one neighbor, are reached.

The presented sum-product algorithm can be efficiently used for inference on undirected trees. To apply the sum-product algorithm to directed trees, they should first be converted to undirected graphs through moralization. Moralizing directed trees generates undirected trees so that the sum-product can be applied to them for exact inference. However, for polytrees as members of DAGs, the moralized undirected graph is not a tree and has loops in it. Therefore, the sum-product algorithm cannot be directly applied to them and there will be the need for the conversion of the undirected graph to a factor graph which has no loops.

2.10 Related work

To employ graphical models for image segmentation, it is required that a graph is first created based on the image. Such a graph consists of nodes corresponding to pixels or superpixels in the image with edges between them that present a similarity or a spatial proximity between them.

Graph-cut methods have been widely used in image segmentation. They operate based on a graph with edges, weights of which are determined based on a similarity measure. Graph-cut usually provides approximate solutions on graphs, for certain models however, it guarantees to find the global maximum, given the energy function based on the joint distribution [149–151].

Alkofahi *el al.* [152] proposed a graph-cut segmentation algorithm for cell nuclei segmentation that initially extracts the foreground through graph-cut binarization. Pixels are labeled by minimizing an energy function that comprises a Poisson distribution for the cost of assigning labels to pixels, and a pixel continuity term that penalizes different labels for neighboring pixels. Following binarization, LoG filtering is used for detecting nuclear seeds, which are then used for initial segmentation refined by a second graph-cut-based algorithm. The method was applied to the segmentation of 25 histopathological images with 7400 nuclei. The authors reported inaccuracies in the segmented images for highly textured chromatins, nuclei with elongated shapes, and highly clustered nuclei with weak borders.

Chang *et al.* [153] proposed a multi-reference graph-cut formulation for the segmentation of nuclei in 440 tissue sections originating from different laboratories with biological and technical variations. The proposed algorithm uses an energy function consisting of global and local data terms plus a smoothness term to convey geodesic information of the input image. The global and local energy terms are calculated based on the manually annotated reference patches and local color space information in the original image, respectively.

Song *et al.* [154] developed a multi-scale convolutional neural network by combining a

CNN with an undirected graphical model for considering size variations in the segmentation of nuclei. Working on a superpixel version of the image generated by the SLIC algorithm [155], they employed the CNN to extract scale-invariant features, based on which a coarse segmentation was generated with a fast min-cut/max-flow algorithm [156]. An undirected graph was made for refinement of the segmentation and the nuclei were extracted using the marker seeking strategy [157].

In [158], graph-cuts were used within an iterative segmentation algorithm that refines the contours based on a cut cost. The contour has to be the global minimum in a neighborhood where its size is set *a priori*. The image is represented as a pixel adjacency graph for which the cost function is calculated. Generating a graph for an image as a set of pixels makes the graph analysis difficult due to the large volume of the data [159]. Ta *et al.* [160] proposed a graph-based regularization algorithm that reduces weighted undirected graphs with arbitrary structures for faster image segmentation. They compared their method to *k*-means clustering and Bayesian classification methods on the segmentation of cellular images.

One way of graph generation that ensures the generated graph does not have any loops, is the construction of a merge tree. In this method, starting from nodes that correspond to pixels or superpixels in the image, each two or more nodes that are similar based on a similarity measure are connected to a new parent node. The procedure recursively continues by connecting the orphan nodes to new parent nodes, until there is only one orphan node in the graph, which is the root node. Lin *et al.* [161] used an undirected merge tree generated based on the oversegmented image for neuronal and glial nuclei segmentation. Their method searches for the optimal merging pattern of the regions created by watershed. Using region selection based on watershed regions combined with dynamic programming, Santamaria *et al.* [162] proposed an unsupervised cell segmentation in immunohistopathology images using merge trees that employs shape and scale information.

Watershed method [43] is a popular method for partitioning the image to homogeneous areas that can be associated with graph nodes. This method divides the image to segments by simulating a flooding of a given scalar function, for example, the gradient of the image intensity. The output of partitioning an image with a watershed method involves areas with boundaries that are likely to lie on the object boundaries. Applying watershed segmentation to the image gradient, Kale and Aksoy performed a joint segmentation of cervical cytoplasms and nuclei in Pap smear images [163]. A two-pass search algorithm [164] was used that combines spectral and shape information to find the most meaningful groupings of segments in a hierarchy modeled based on the multi-scale region adjacency graph.

Exact inference algorithms have been developed for graphical models with simple structures. Two-pass inference algorithms were initially proposed for chain-based models, which calculate exact probabilities for node labels [165]. Extension of this forward-backward algorithm, known as belief propagation [140, 166], resulted in exact solutions for two main categories of DAGs: *trees* and *polytrees*. These early methods factorize the posterior of the desired node into two factors, including a marginal posterior given a subset of the observations, and a subgraph likelihood function given the label of the node. Although this factorization provides an exact and non-iterative inference on graphs, it could not be implemented due to their dependency on the likelihood function that required working with infinitesimal values. This takes place in cases where the likelihoods include a large number of data components, leading the probability values to become very small and resulting in arithmetic underflow.

To address this problem [166], Laferte *et al.* proposed an exact calculation of posteriors on quadtrees [167] that finds the posteriors based on two passes on the graph. In their work, they factorized the posterior of each node into two marginal posteriors, one being based on the observation nodes in the top nodes, and the other based on the observation nodes in the bottom nodes. Therefore, the posterior was composed of two types of messages being calculated in two passes, one bottom-up and one top-down message. Extensions of this method have been proposed for different image segmentation applications [168–170]. Feng *et al.* combined tree-structured belief networks as prior models with neural networks for multi-label image segmentation [171]. They optimized the model parameters using the EM algorithm and evaluated their method on scene image segmentation.

Using fixed tree structures makes the inference efficient, however since it may cause neighboring pixels in the image in different uncorrelated nodes without a common parent, it can result in blocky artifact [172–174]. Blocky segmentations do not present accurate object boundaries by not acknowledging the natural curves in the image. To address this problem of trees, dynamic and irregularly structured graphs have been proposed [175, 176].

Dynamic tree approaches consider a distribution over the tree structure and optimize it as part of the inference [177]. Storkrey *et al.* [178] introduced Position-Encoding Dynamic Tree (PEDT) that allows manipulation of objects locations through a flexible hierarchical representation of the image structure. They defined the probabilistic model using belief networks and presented a variational inference for it. The model was evaluated on image segmentation and was compared to the original dynamic trees and fixed trees. Designing a new architecture of dynamic trees, Todorovic *et al.* used a dynamic tree that incorporates multi-layered data at all scales of the model [179]. To increase the inference efficiency, they used a Structured Variational Approximation (SVA) [180], that renders approximate independence of variables in the graph. Evaluating the method in unsupervised segmentation, they showed that the model can capture relationships between objects and their parts. Additionally, dynamic trees have been used to handle large deformation in registration and multi-atlas segmentation [181].

Another group of trees called Hierarchically-Structured Interacting Segments (HINTS) facilitate the incorporation of prior knowledge about geometric interactions between regions and boundaries, for example being interior and exterior of an object [104, 182]. Yin *et al.* [183] proposed a method for segmentation of multiple interacting surfaces from different objects while embedding multiple inter-relations in an n-dimensional graph. They evaluated their method on the simultaneous segmentation of six bone and cartilage surfaces in the knee joint. Extending this approach to trees with arbitrary structures, Stack *et al.* proposed an optimization while preserving inclusion, exclusion and margin constraints [184]. They developed an optimization algorithm for general HINTS that addresses the local minima and high-order term problems in the previous methods.

Using dynamic programming to perform bottom-up message passing on an undirected tree graphical model, Liu *et al.* [185] proposed an algorithm for muscle cell segmentation. The hierarchical graph was generated based on an edge map created for the image by the method presented in [186]. A score was then assigned to each region based on edge strength and sparse shape representation [187], and the best region was selected by maximizing a total score that guarantees the selection of one region among overlapping regions. The presented method provides a segmentation with the highest overall score which is equivalent to selecting the best non-overlapping regions. In a similar work for muscle cell segmentation [185], the authors used a region selection method based on edge maps created by a

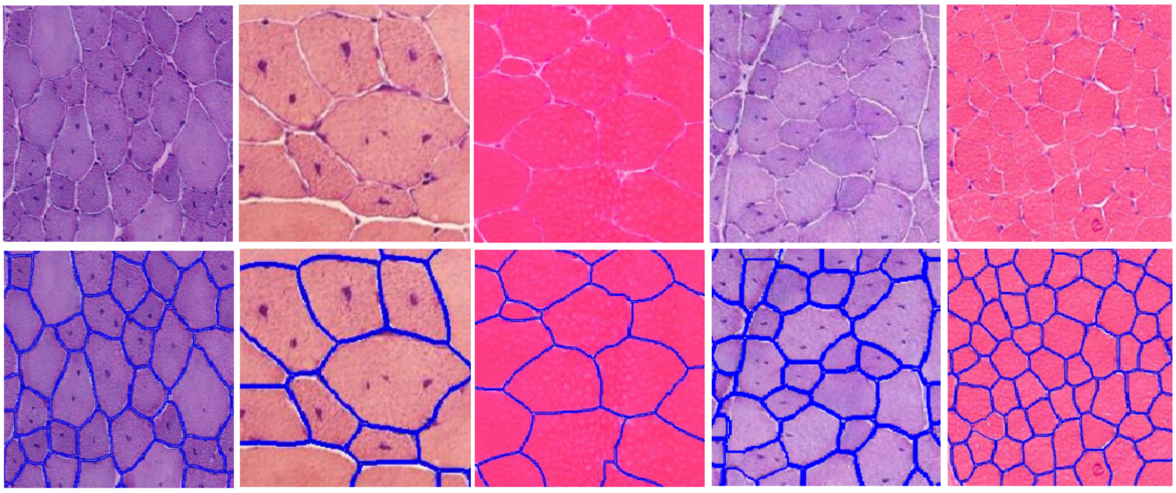


Figure 2.9: Segmentation results of using the method in [185] for four sample muscle cell images. The top row shows the original image patches (at 10x magnification) and the bottom row shows their corresponding segmentations, with blue contours. Image has been adapted from [185].

random-forest-based hierarchy. Figure 2.9 shows sample results using this method.

2.11 Discussion

This chapter presented the need for multi-class segmentation methods in microscopy image analysis and the role of employing prior knowledge in improving segmentation. More specifically, graphical models are introduced and elaborated for their use in capturing different types of prior knowledge as labeling constraints in inference. The fundamental definitions of the graph theory and principal inference algorithms are provided for undirected and directed graphs, which closely relate to the proposed method presented in the following chapters.

Despite the effectiveness of the existing methods, maximal incorporation of prior knowledge as part of an efficient inference algorithm is yet to be established. Problems with the existing graphical models for segmentation include two aspects. The captured interrelations between objects in the image are too simplistic and/or the inference algorithm is iterative or approximative. Any of these two aspects can reduce the segmentation performance. With a focus on these two issues, I will present a novel inference method using polytree graphical models in chapter 4. Chapter 5 extends the proposed model to weakly supervised segmentation with a more comprehensive evaluation. Further, applications of the segmentation algorithm in measuring quantitative indices from the segmented images are also explored.

Chapter 3

Bayesian Polytrees with Learned Deep Features for Multi-Class Cell Segmentation

3.1 Introduction

Accurate and efficient image segmentation of complex spatial object arrangements composed of multiple constituting structures (or classes) is challenging yet paramount for biological discoveries underpinned by quantitative imaging. For example, the identification of different cells within tissue [21] or organelles within cells [22], the sub-cellular localization of proteins [23], the interactions of different cell types in organ development [24], or the immune response during infection [1], as just a few examples of relevant problems in biology. To assess the morphological and behavioral characteristics of these cells (some having unknown causes [188]) quantitative metrics are devised, which require image segmentation as an unavoidable first step. Additionally, histology images are increasingly used for disease diagnosis and grading. Quantitative analysis of these images through the developed metrics (e.g. for abnormal nuclei as a potential indicator of cancer) helps pathologists by providing a supporting diagnosis and disease progress evaluation [4, 5]. Still, at a finer resolution, the biology of the cell nucleus, i.e. the organization of the genome and the proteins, has a functional relevance with the biological cell processes, and their mislocalization (hence segmentation) can be a valuable indicator for many pathologies [7–9]. Given that all the above-mentioned examples are multi-class segmentation problems, automatic methods are of high significance due to their labor-intensity, and inter- and intra-observer variability of manual analysis, especially for large datasets. However, common features of such images, such as defused or overlapping boundaries, irregular shapes and high deformability of objects, limited resolution, and quality in biological images, may contribute to the poor segmentation performance of automatic methods.

In this work, we are looking at two microscopy image datasets. The first dataset, BBBC020, containing images depicting bone-marrow derived macrophages from C57BL/6 mice stained with DAPI to label the nuclei and CD11b/APC to label the cell surface, to investigate the role of toll-like receptors for macrophage spreading [25]. Automatic segmentation of the image data is required for measurement of experimental parameters. The second dataset. BBBC007, has drosophila melanogaster Kc167 images stained for DNA to label nuclei and actin to show the cell body. A segmentation of the images is required to be used in the automatic cytometry [25]. Figure 3.1 shows samples from each of the datasets. The images from the two datasets define a challenging cell segmentation problem due to the irregular shapes

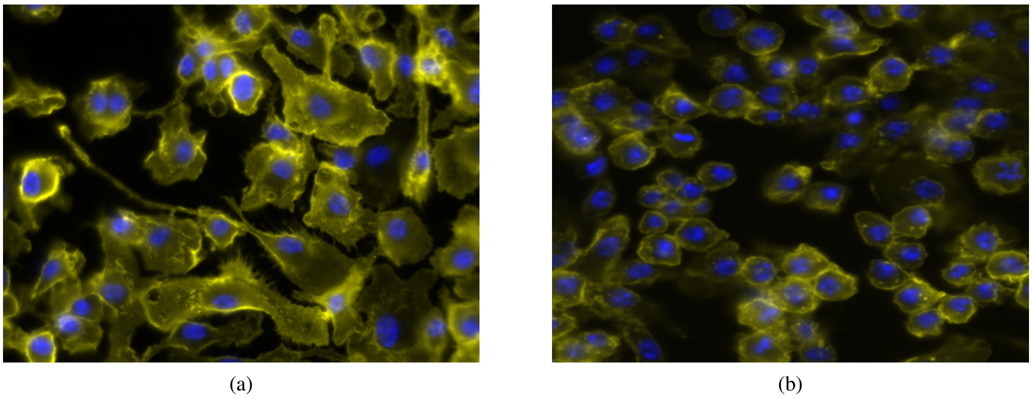


Figure 3.1: Samples from BBBC020 (a) and BBBC007 (b) datasets [25].

and touching and overlapping cells in them.

Incorporation of prior knowledge can play an important role in aiding and improving segmentation. Inter-object dependencies have been used in the segmentation of interacting objects [105] and intra-object spatial relationships were shown to enhance the segmentation of cell organelles [106]. Other examples in cell segmentation include using priors to consider the relative topology of cells and nuclei [108, 109], and to impose area and size constraints on segmented regions [110], to achieve a better segmentation. In brain tissue analysis, appearance and spatial priors have been used to improve tumor localization [111], generalization to unseen images [112], and lesion recognition as atypical brain tissues [113].

Graphical models enable modeling associative relations between random variables [127]. These probabilistic models can improve segmentation by imposing constraints emerged from prior knowledge [125, 126]. The key aspect of graphical models is that the label of each node is determined based on both its own attributes and attributes of other nodes connected through graph edges. This way, not only all the information is incorporated in inferring the labels, but also label configuration constraints can be effectively applied during inference. For instance, Chen *et al.* [128] employed graphical models to incorporate nuclear positions with boundary information for yeast cell segmentation. In a rather different application, segmentation of retinal images, graphical models have been used for combining appearance models with shape priors [130].

We propose polytree graphical models for implementation of local label configurations for multi-class segmentation problems. Polytrees are a type of Bayesian Networks (BNs) whose nodes can have more than one parent. Compared to other well-known BNs based on trees [179, 189–191], where each node only has one parent, polytrees can capture more complex configurations and constraints. This higher flexibility of polytrees also inhibits certain unfeasible label cliques on the graph that trees are unequipped to exclude in spite of their contravening prior knowledge. The performance of the proposed method was compared to that of the directed trees and three convolutional neural networks to assess the modeling and error prediction efficiencies.

3.2 Related work

Two types of graphical models have been used for image segmentation, namely Markov Random Fields (MRFs) and Bayesian Networks (BNs). MRFs have weighted edges indicating dependencies between variables and are used for capturing correlations between random variables. Directed edges in BNs indicate causal relationships between random variables [66]. In this paper, we focus on BNs and enforce the constraints using conditional probabilities that appear in the joint probability distribution. To find optimal labels of the graph, different inference algorithms have been proposed. Two-pass inference algorithms were initially proposed for chain-based models, which calculate exact probabilities for node labels [165]. Extension of this forward-backward algorithm, known as belief propagation [140, 166], resulted in exact solutions for two main types of Directed Acyclic Graphs (DAGs): *trees* and *polytrees*. Directed trees are BNs with only one route between each pair of nodes in the graph (i.e. singly connected [127]), with each node, except the root node, having exactly one parent node. Polytrees, however, are singly connected BNs where each node can have more than one parent node. Existing solutions for these two DAGs factorize the posterior of each node into two factors: a marginal posterior given a subset of observations, and a subgraph data likelihood given the label of the node [166]. Despite their being exact and non-iterative, the dependency on the likelihood function in these factorizations makes the numerical implementation impractical [167]. This is because probability values become very small at some nodes, where the likelihoods involve a large number of data components, hence causing arithmetic underflow.

To address the implementation problem of the proposed algorithms for inference, Laferte *et al.* [167] designed a recursive framework that calculates exact posteriors of nodes on a regular quadtree, based on posteriors of its neighboring nodes. Feng *et al.* [171] used Tree-Structured Belief Networks (TSBNs) as a prior model combined with a neural network for local prediction of class labels. TSBNs suffer from block artifacts [172] resulting from the descendants of a node s on a tree being conditionally independent, given the state of s. More complex graph structures, such as overlapping trees [173] whose nodes do not point to distinct areas of the image, and two-dimensional trees [176] have been proposed to reduce this effect at the expense of higher computational costs. Alternatively, a group of authors proposed trees with dynamic structures fitting the image contents (e.g. [178, 181]) where the labels and the graph structure are inferred. Priors have also been incorporated into trees using Hierarchically-Structured Interacting Segments (HINTS) [105], where the nodes represent interacting segments in the image. Iterative algorithms were proposed for approximating the optimal label configurations for binary [182] and multi-label [38] cases. However, the proposed optimization algorithms do not always converge and may require modifying the graph structure or relaxing the constraints for convergence.

To address the limitations of trees, which mainly stem from the independence of same-level nodes [171], we propose polytrees for multi-class image segmentation. Compared to trees, polytrees can eliminate a wider range of unfeasible label configurations, by modeling both inter- and intra-level dependencies. Similar to the work of Laferte *et al.* [167], we derive a two-pass inference algorithm on the polytree for exact calculation of posterior probabilities on the graph. The proposed polytree based method is evaluated by segmenting objects from multiple classes in real microscopy images. We show it outperforms state-of-the-art convolutional neural networks and directed trees.

The proposed model is also evaluated on its ability to predict segmentation error. Areas of the segmented image that do not comply with the imposed priors are nominated and their similarity to the actual segmentation error is measured. Polytrees are shown to outperform trees in finding the wrongly segmented areas.

Our polytree-based segmentation method is fundamentally different from the method proposed by Laferte *et al.* [167] and entails important extensions. The proposed hierarchical graph structure (Fig. 3.3) is made based on an initial superpixelation step [192] and subsequently merges the most similar superpixels (graph nodes) until the highest level. The graph structure is asymmetric and irregular. This property allows capturing more natural cell boundaries for a more complex implementation. Conversely, Laferte *et al.* use symmetric and regular quadtrees, where the nodes are represented by square regions. The shapes of the nodes do not match the actual morphologies of the cells, rendering the method unsuitable for comparison. Inference-wise, our method uses features extracted by convolutional neural networks (CNN) (details explained in section 3.4.6) and is applied to supervised multi-class image segmentation, while Laferte *et al.* use pre-defined intensity and texture features for an EM-based unsupervised image classification. See Table 3.1 for a summary of fine differences between the three mentioned methods.

This paper significantly extends our preliminary work presented in [193] through the following specific contributions:

- The role of features in the final segmentation performance is investigated by using scale-space differential invariants of intensity with an automatic feature selection scheme, employing the most relevant features for the analysis.

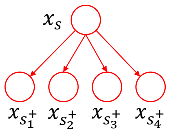
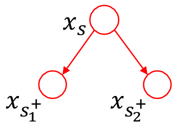
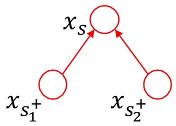
- We have shown how deep features from recent convolutional neural networks can be systematically exploited within the proposed polytree framework for improved segmentation quantities.

- Polytrees are compared to customized trees, and state-of-the-art CNNs, namely SegNet [78], DeepLab [194] and PSPNet [195], using synthetic and two real microscopy image datasets.

- A novel mechanism is employed to predict possible errors in the segmented images, by comparing the label configurations with the imposed label constraints.

- The error introduced through the superpixel generation step of the algorithm is analyzed for a more accurate evaluation of the inference.

Table 3.1: Summary of key differences between Laferte *et al.* method and the proposed tree and polytree

Method	Laferte *et al.*	Proposed tree	Proposed polytree
Descendants	4	2	2
Structure	Regular	Irregular	Irregular
Features	Intensity and texture	CNN	Intensity / CNN
Application	Unsupervised	Supervised	Supervised
Constructing element	 x_s $x_{s_1^+}$ $x_{s_2^+}$ $x_{s_3^+}$ $x_{s_4^+}$		

3.3 Method

Herewith, we present our proposed graphical model for image segmentation. First, a polytree is generated for the image, grouping similar pixels and regarding them as nodes in the graph. Next, the parameters of the likelihood functions are trained and labels of the nodes are inferred. Finally, the segmented image is constructed based on the optimal labels on the graph.

3.3.1 Graphical modeling for image segmentation

We perform the image segmentation by reformulating it as the problem of finding the optimal labeling for a graphical model, generated based on the image. The graph contains two types of nodes that represent the latent variables and the observations for their corresponding part of the image. Given the observations, finding the values of the latent variables is equivalent to labeling the corresponding area in the image i.e. segmenting the image. As shown in Fig. 3.4a, each element s (representing an area in the image) in the graph $\mathcal{G}$ with M elements comprises a latent variable node x_s attached to an observation node $\boldsymbol{y}_s$. This label-observation configuration is an element of the graph, in which $\boldsymbol{y}_s$ and x_s can be considered as input and output values, respectively. The latent variable $x_s \in \mathbb{X}$ ($\mathbb{X}$ being the set of all latent variable nodes) takes a discrete value from the label set Λ and $\boldsymbol{y}_s$ contains feature vectors extracted from its corresponding area in the image. The process of generating the graph and labeling it based on the imposed priors is explained in the rest of this section.

3.3.2 Graph generation

Initially, the graph is generated by grouping pixels into locally coherent areas (superpixels), each representing a single root node (Fig. 3.3). We use the SEEDS algorithm [192], which refines an initial grid of identically block shaped superpixels into more coherent ones.

SEEDS algorithm does not rely on an initial oversegmentation and starts from a grid on the image. The initial superpixel partitions are then iteratively updated to maximize an objective function that enforces colour homogeneity and shape boundary smoothness in each region. For a partitioning s the energy function would be as follows,

$$E(s) = H(s) + \gamma G(s), \tag{3.1}$$

where s is the partitioning and term $H(s)$ accounts for colour distribution corresponding to s. Denoting the k^{th} superpixel by A_k, we denote $c_{A_k}(j)$ as the colour histogram of pixels in A_k for j^{th} histogram bin.

$$c_{A_k}(j) = \frac{1}{Z}\sum_{i\in A_k} \delta(I(i) \in H_j). \tag{3.2}$$

In the above formula, H_j is the set of colours in the j^{th} histogram bin, $I(i)$ returns the colour of pixel i, $\delta(.)$ is a function returning 1 for pixels with colours in H_j and 0 otherwise, and Z is the normalization coefficient. Defining the colour quality measure as $\psi(c_{A_k}) = \sum_{H_j}(c_{A_k}(j))^2$, $H(s)$ would then be an evaluation of the quality measure by taking into account the quality of all superpixels in the form $H(s) = \sum_k \psi(c_{A_k})$. The term $\psi(c_{A_k})$ reaches its maximum when all the pixels are in the same histogram bin leading to $\psi(c_{A_k}) = 1$. All other distributions would result in lower amounts of the function. According to this, the defined energy terms enforce the superpixels to have one or few colours for $H(s)$ to be maximized. The term $G(s)$ is responsible for the shape of the superpixel. For each pixel i in the image, an $N \times N$ patch N_i is considered around the pixel and the number of superpixels present in the patch is considered as a measure of smoothness. Higher number of different superpixels

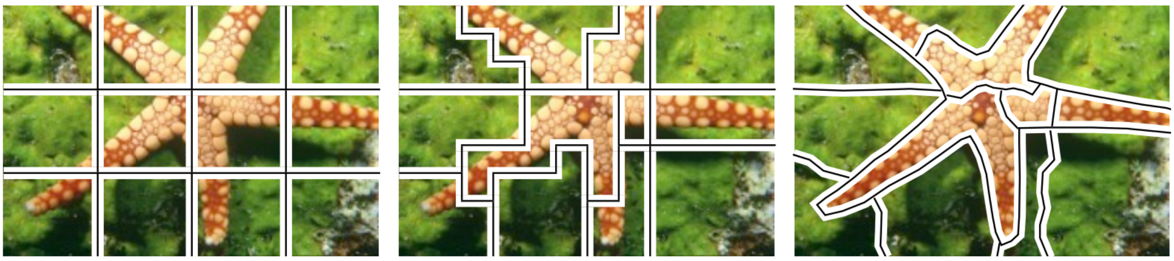


Figure 3.2: Sample image oversegmented by the SEEDS algorithm. Superpixels start from the trivial grid on the image and their boundaries are iteratively updated to more accurately be projected on the object boundaries. [196]

present in a patch corresponds to lower smoothness. In accordance with colour distribution, a superpixel histogram is defined for the area N_i as

$$b_{N_i}(k) = \frac{1}{Z} \sum_{j \in N_i} \delta(j \in A_k). \tag{3.3}$$

Similar to the colour term, the measure of quality for shape is defined as follows

$$G(s) = \sum_i \sum_k (b_{N_i}(k))^2. \tag{3.4}$$

The defined energy term favors that each patch consists of pixels of only one superpixel. Even though, this is not possible for all the pixels, such as pixels near boundaries, maximizing the energy term would result in more regular boundaries. This happens since boundaries which are not smooth cause patches to have a higher number of superpixels in them which are penalized to maximize the energy. Having written the energy function, it is updated using a hill climbing algorithm. Using this algorithm, small local changes are made to each superpixel in each iteration to increase the value of the energy function. Continuing iterations makes the quality of superpixels better. However, the iterations continue until a certain amount of time is passed, which is manually set. Having the images oversegmented, the generated regions are used for graph generation. Results of applying the oversegmentation method to a sample image is illustrated in Figure 3.2.

Once the oversegmentation is done, the two most similar superpixels are recursively merged to generate higher-level nodes in the graph hierarchy, in a similar manner to generating a merge-tree [197].

For each superpixel at the finest level, one (root) node in the lowest level of the graph is created (see Fig. 3.3). Every two nodes achieving highest scores according to a similarity metric are then merged to create a new supernode. The new supernode is the union of image regions attached to its two lower level descendant nodes. We define the similarity metric as a superposition of distances using spatial and intensity features of the superpixels. A vector $\boldsymbol{\beta} = [\boldsymbol{\beta}_s;\boldsymbol{\beta}_i]$ is introduced to adjust contributions of each feature in the similarity metric. An adaptive scheme is designed for setting $\boldsymbol{\beta}$, which helps in the generation of more meaningful nodes on the graph. Nodes in lower levels of the graph represent subregions of objects, rather than their full areas. For these nodes, we set $\boldsymbol{\beta}$ such that $\boldsymbol{\beta}_s$ consists of greater values compared to $\boldsymbol{\beta}_i$. This makes merging neighboring nodes that correspond to parts of the same object (i.e. a cell or a nucleus in our case) more probable. In higher levels, however, values of $\boldsymbol{\beta}_i$ are set to be greater than those of $\boldsymbol{\beta}_s$ to facilitate the merging of regions belonging to the same class, although they might not be neighbors. Assuming $\boldsymbol{\beta}_i = \beta_i\mathbf{1}$ and $\boldsymbol{\beta}_s = \beta_s\mathbf{1}$ and setting β_i=1 for simplicity, $\boldsymbol{\beta}$ is determined by a cross-validation merely on β_s.

After each merging step, the new node and all the other *orphan* nodes, are assessed with the similarity metric to recognize candidate nodes for merging next. Region merging is continued until only two orphan nodes remain in the graph, which are eventually merged to create the leaf node that corresponds to the whole image (Fig. 3.3). Since two nodes are merged at each step of the graph evolution, the resulting structure is a binary graph; i.e. each non-root node has two descendant nodes directly connected to it. We call this three-wise structure a *clique* and denote it by $parent1 - child - parent2$.

Figure 3.4b shows a symbolic process of merging for a cell (C) with a nucleus (N). Here, nodes 1 and 2 align with blue and yellow areas in the synthetic cell. If these two nodes are chosen to be merged based on their value in the similarity metric, node 3 is generated, which corresponds to the union of blue and yellow areas annotated by the dashed ellipse. This clique is represented by $1 - 3 - 2$.

3.3.3 Graph definition

The generated graph is a hierarchical structure modeling the interrelations between areas corresponding to different classes. Nodes in lower levels correspond to smaller superpix-

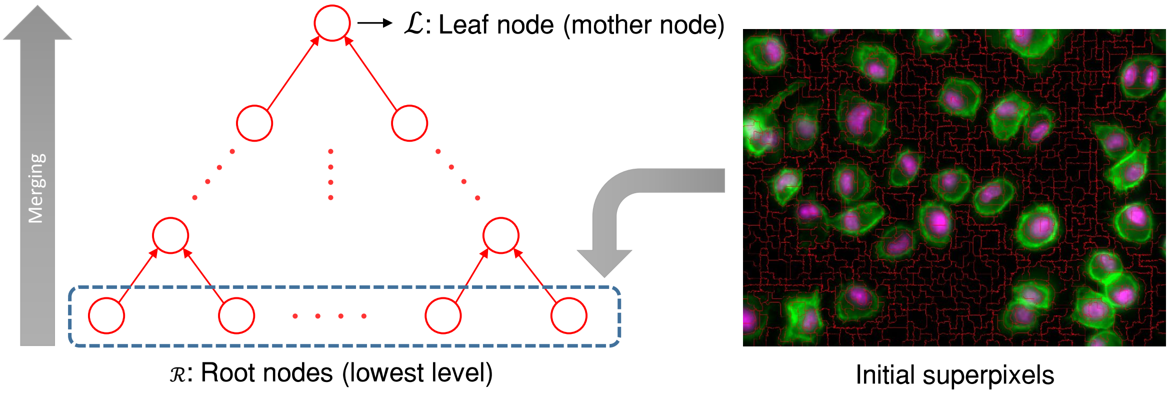

Merging
$\mathcal{L}$: Leaf node (mother node)
$\mathcal{R}$: Root nodes (lowest level)
Initial superpixels

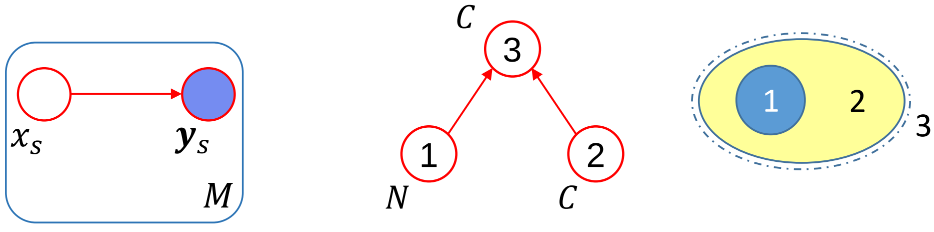
Figure 3.3: Generating a polytree from an oversegmented input image.

els, such as sub-areas of cells, and are therefore more homogeneous. Higher level nodes correspond to one or multiple objects that can be of different classes. The hierarchical structure allows merging smaller areas from the same class (in lower levels), and embedding of objects within larger regions with different classes (in higher levels) according to certain merging rules. These rules are introduced in the model by defining and applying priors on label configurations. In this setting, segmenting the image equals inferring labels x_s given the observations $\boldsymbol{y}_s \in \mathbb{Y}$ ($\mathbb{Y}$ being the set of all observation nodes), where the label configurations comply with the prior imposed on the model.

3.3.4 Imposing priors on the graph

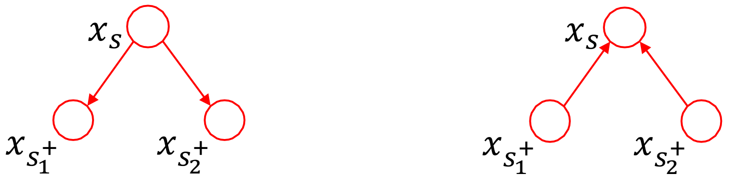
Applying inclusion-based prior knowledge is the main advantage of using hierarchical graphs and is a way to constrain the solution to plausible results. In a directed graphical model, prior knowledge can be modeled through setting specific forms of the conditional probabilities that implement causality according to the directions of the edges. These probabilities act as the prior factor in the Bayesian factorization of the posterior.

In directed trees, the joint probability consists of one-to-one priors that can only model across-level dependencies. For instance, in the constructing element of a dyadic tree depicted in Fig. 3.5a (excluding the observation nodes temporarily for simplicity) the joint probability is written as $p(\mathbb{X}) = p(x_{s_1^+}|x_s)p(x_{s_2^+}|x_s)p(x_s)$, where $p(x_{s_1^+}|x_s)$ and $p(x_{s_2^+}|x_s)$ are the one-to-one priors. In polytrees, however, the joint probability has multiple-to-one priors



(a) (b)

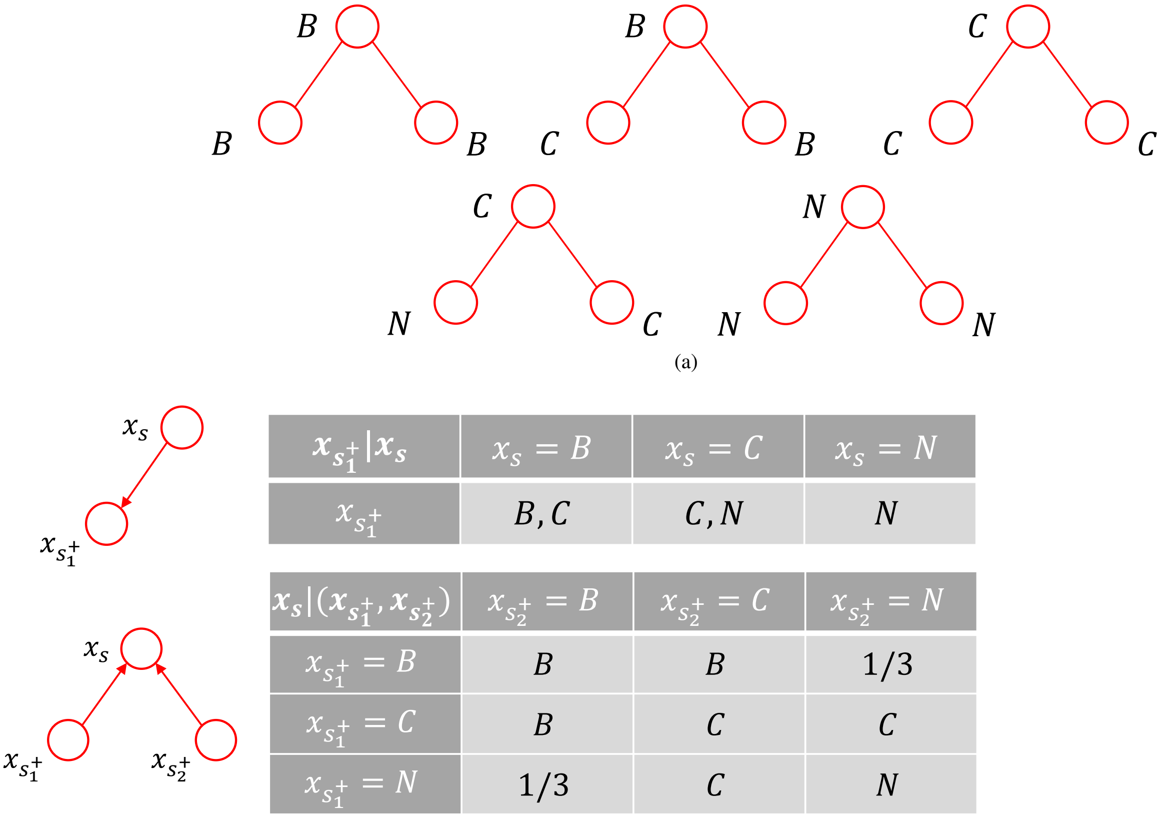
Figure 3.4: Explanation of the graphical model used for segmentation. A label-observation element s corresponding to an area in the image is shown in (a), in which the blue plate represents M elements of which only an example is shown.
Panel (b) shows a symbolic process of node merging for a synthetic cell (C) with a nucleus (N) resulting in a polytree constructing element.



(a) (b)

Figure 3.5: Edge directions on cliques in directed tree (a) and polytree graphical models (b).

modeling both across-level and same-level dependencies. The joint probability for the sample polytree structure of Fig. 3.5b is factorized as $p(\mathbb{X}) = p(x_s|x_{s_1^+}, x_{s_2^+})p(x_{s_1^+})p(x_{s_2^+})$, in which the factor $p(x_s|x_{s_1^+}, x_{s_2^+})$ is the prior. To show how this can influence the modeling ability of the hierarchy, imagine the label set consists of two classes: $\Lambda = \{A, B\}$. Also, assume $B-A-A$ is a feasible and $B-A-B$ is an unfeasible configuration. Using trees, $B-A-A$ is allowed by setting probabilities $p(x_{s_i^+} = B|x_s = A)$ and $p(x_{s_i^+} = A|x_s = A)$ to non-zero values. However, enforcing the former constraint also makes $B-A-B$ cliques feasible, even though they are to be prevented by the model. But thanks to the more complex priors in the polytree, setting $p(x_s = A|x_{s_1^+} = B, x_{s_2^+} = A)$ to non-zero values and setting $p(x_s = A|x_{s_1^+} = B, x_{s_2^+} = B)$ to zero satisfies both of the constraints with no conflicts. This simple example shows the advantage of polytrees over directed trees in modeling more complex problems, by using a larger number of parameters.



(a)

$x_{s_1^+} \mid x_s$	$x_s = B$	$x_s = C$	$x_s = N$
$x_{s_1^+}$	B, C	C, N	N

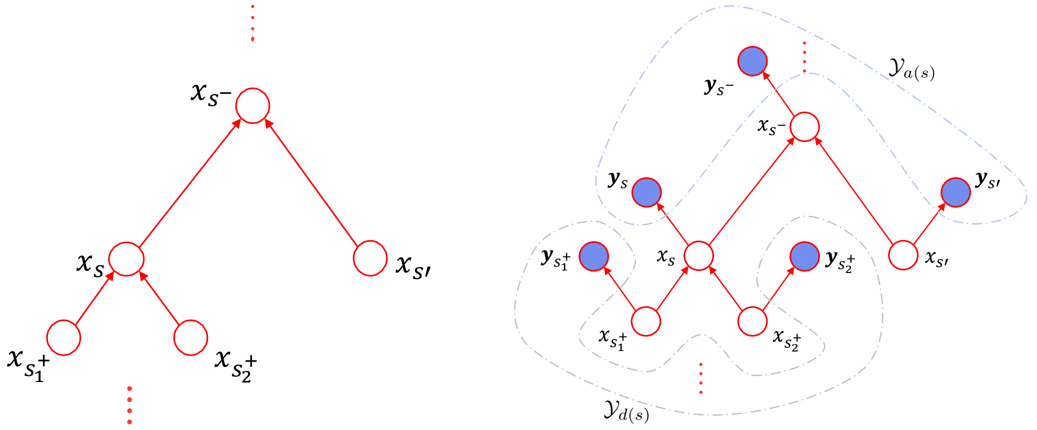
$x_s \mid (x_{s_1^+}, x_{s_2^+})$	$x_{s_2^+} = B$	$x_{s_2^+} = C$	$x_{s_2^+} = N$
$x_{s_1^+} = B$	B	B	$1/3$
$x_{s_1^+} = C$	B	C	C
$x_{s_1^+} = N$	$1/3$	C	N

(b)

Figure 3.6: The prior knowledge used in this paper for the three-class problem of cell and nucleus segmentation. Panel (a) shows the plausible label-configurations based on the inclusion of nuclei by cells and cells by the background. Panel (b) shows equivalent probabilistic conditionals when directed trees or polytrees are used for modeling the image. When no child label x_s is plausible for a pair of parent labels $x_{s_1^+}$ and $x_{s_2^+}$, a uniform prior $1/3$ was considered.

In this paper, we use the generated polytree (details explained in section 3.3.2) to segment the image by inferring the optimal labels for latent variable nodes. Each node at the lowest graph level (finest image resolution) is a root (in contrast to the single root node in directed trees) and there is only one leaf node (see Fig. 3.3).

Figure 3.6 shows the tables of priors $p(x_{s_1^+}|x_s)$ in trees and $p(x_s|x_{s_1^+}, x_{s_2^+})$ in polytrees, and possible label configurations, when three classes of background (B), cell (C) and nucleus (N)



(a) (b)

Figure 3.7: Distribution of latent and observation nodes on the graph. The notation for nodes connected to an internal node s of the graph is shown in (a). In (b), the graphical representation of ascendant, $\mathcal{Y}_{a(s)}$, and descendant, $\mathcal{Y}_{d(s)}$, observation nodes is depicted.

exist in the image. Conditional probabilities were set to zero for implausible configurations, e.g. $p(x_s = C|x_{s_1^+} = B, x_{s_2^+} = B) = 0$, and to nonzero for plausible configurations, e.g. $p(x_s = B|x_{s_1^+} = B, x_{s_2^+} = B) = 1$. For cases where no child label x_s is possible for a pair of parent labels $x_{s_1^+}$ and $x_{s_2^+}$, a uniform prior was considered, e.g. $p(x_s|x_{s_1^+} = B, x_{s_2^+} = N) = 1/3$.

3.3.5 Label inference

Let $\mathbb{X} = \{x_s\}$ and $\mathbb{Y} = \{\boldsymbol{y}_s\}$ denote sets of labels (latent variables) and the corresponding observed features at nodes, respectively, $\mathcal{G}$ denote the set of nodes and edges and $x_s \in \Lambda$, where Λ is the set of all possible labels. For an internal node (neither in the lowest level nor the leaf node) s in the graph, s^-, s^+ and s' denote nodes in higher, lower and same layers, respectively (Fig. 3.7a).

We now derive equations governing the posterior probabilities of graph nodes. Given the observed data $\mathbb{Y}$, finding the best segmentation equals the best configuration of labels $\mathbb{X}$ for the graph. Bayesian inference associates the most probable label from the set of possible labels Λ, given all observations:

$$\forall s \in \mathcal{G}, \hat{x}_s = \arg\max_{x_s \in \Lambda} p(x_s|\mathbb{Y}) \tag{3.5}$$

A new set of equations is derived to calculate the closed-form posterior probabilities at each node in the polytree. The inference algorithm calculates the posteriors of the nodes in two passes. These two consist of a pass from the leaf to the roots, (*top-down* pass), and another from the roots to the leaf (*bottom-up* pass).

The probability of a node's label x_s, given all data $\mathbb{Y}$, is computed by marginalizing the probability of the clique over two parent nodes s_1^+ and s_2^+ given $\mathbb{Y}$, and the *joint posterior* is given by

$$p(x_s|\mathbb{Y}) = \sum_{x_{s_1^+}, x_{s_2^+}} p(x_s, x_{s_1^+}, x_{s_2^+}|\mathbb{Y}) \tag{3.6}$$

Three-wise constraints on cliques appear in the posterior calculation. To factorize the joint probability, we need a mechanism to identify the dependency of the nodes in the graph.

D-separation

Consider three sets of nodes A, B, and C in a directed acyclic graph. We want to verify the conditional dependency of A and B, given C. D-separation (directional separation) rule [140] can determine this based on the paths that exist between A and B on the graph. Each path connecting A and B is blocked if it involves a node s for which either: a) arrows meet head-to-tail or tail-to-tail at node s and $s \in C$ (Fig. 3.8a), or b) arrows meet head-to-head at node s and neither the node nor any of its descendants are in the set C (section 3.8b). If all paths from A and B are blocked, they are conditionally independent, given C (A and B are d-separated by C and $A \perp\!\!\!\perp B|C$).

Using the d-separation rule, the joint posterior in Eq. 3.6 is expanded as

$$\begin{aligned} p(x_s, x_{s_1^+}, x_{s_2^+}|\mathbb{Y}) &= p(x_s|x_{s_1^+}, x_{s_2^+}, \mathbb{Y})p(x_{s_1^+}, x_{s_2^+}|\mathbb{Y}) \\ &= p(x_s|x_{s_1^+}, x_{s_2^+}, \mathcal{Y}_{a(s)}) \\ &\quad p(x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s_1^+,s_2^+)}, \mathcal{Y}_{d(s_1^+,s_2^+)}), \end{aligned} \tag{3.7}$$

where $\mathcal{Y}_{a(.)}$ and $\mathcal{Y}_{d(.)}$ refer to the sets of observation nodes of the ascendant and descendant nodes, respectively (Fig. 3.7b). For each node s (or a set of nodes $\mathcal{S}$), ascendant nodes refer to the set of all nodes that are connected to s ($\mathcal{S}$) through edges with inward directions. Similarly, descendant nodes include the nodes connected to node s ($\mathcal{S}$) through outward oriented graph edges. The union of ascendant and descendant observation nodes constructs the set of all observations. See Fig. 3.7b for a graphical explanation.

We first elaborate on the factor $p(x_s|x_{s_1^+}, x_{s_2^+}, \mathcal{Y}_{a(s)})$ on the right-hand side of Eq. 3.7. This factor enforces posteriors of unfeasible configurations to zero, as it is a product of the joint probability of a child node and its two parent nodes.

$$p(x_s|x_{s_1^+}, x_{s_2^+}, \mathcal{Y}_{a(s)}) = \frac{p(x_s, x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s)})}{\sum_{x'_s} p(x'_s, x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s)})} \tag{3.8}$$

Using the d-separation rule, the numerator becomes:

$$\begin{aligned} p(x_s, x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s)}) &= p(x_{s_1^+}, x_{s_2^+}|x_s)p(x_s|\mathcal{Y}_{a(s)}) \\ &= \frac{p(x_s, x_{s_1^+}, x_{s_2^+})}{p(x_s)} p(x_s|\mathcal{Y}_{a(s)}). \end{aligned} \tag{3.9}$$

The factor $p(x_s, x_{s_1^+}, x_{s_2^+})$ in Eq. 3.9 controls the occurrence of feasible and unfeasible configurations on the graph, by setting nonzero and zero values, respectively. The factor $p(x_s|\mathcal{Y}_{a(s)})$ in Eq. 3.9 is the posterior of node s given the observations of all its ascendant nodes and its

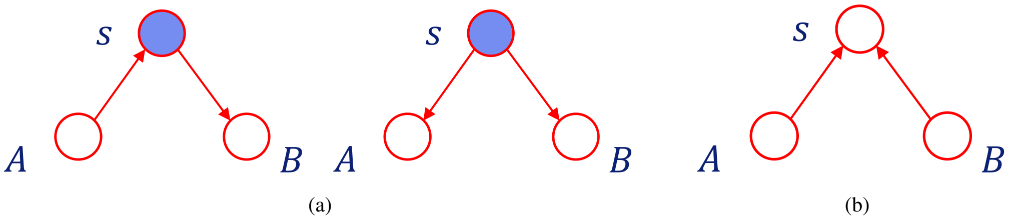


Figure 3.8: D-separation rule. Nodes A and B are conditionally independent given C, when graph edges meet head-to-tail or tail-to-tail and $s \in C$ (a), or when graph edges meet head-to-head and $s \notin C$ (b).

own observations. This *top-down posterior* is expanded as:

$$\begin{aligned} p(x_s|\mathcal{Y}_{a(s)}) \propto \sum_{x_{s^-},x_{s'}} & p(\boldsymbol{y}_s|x_s)p(\boldsymbol{y}_{s'}|x_{s'})p(x_{s'}|\mathcal{Y}_{d(s')}) \\ & \frac{p(x_s, x_{s'}, x_{s^-})}{p(x_{s^-})p(x_{s'})}p(x_{s^-}|\mathcal{Y}_{a(s^-)}). \end{aligned} \tag{3.10}$$

Equation 3.10 indicates that having calculated the likelihood probabilities $p(\boldsymbol{y}_s|x_s)$, $p(\boldsymbol{y}_{s'}|x_{s'})$, and the posterior $p(x_{s'}|\mathcal{Y}_{d(s')})$, the top-down posterior of node s is calculated based on top-down posterior of the node s^-. This implies that a top-down recursion calculates the top-down posteriors for all nodes.

The factor $p(x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s_1^+,s_2^+)}, \mathcal{Y}_{d(s_1^+,s_2^+)})$ on the right-hand side of Eq. 3.7 is factorized by several applications of d-separation rule. This factorization separates parts calculated from ascendant and descendant nodes as follows.

$$\begin{aligned} & p(x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s_1^+,s_2^+)}, \mathcal{Y}_{d(s_1^+,s_2^+)}) \\ & \propto p(\mathcal{Y}_{a(s_1^+,s_2^+)}, \mathcal{Y}_{d(s_1^+,s_2^+)}|x_{s_1^+}, x_{s_2^+})p(x_{s_1^+}, x_{s_2^+}) \\ & = p(\mathcal{Y}_{a(s_1^+,s_2^+)}|x_{s_1^+}, x_{s_2^+})p(\mathcal{Y}_{d(s_1^+,s_2^+)}|x_{s_1^+}, x_{s_2^+})p(x_{s_1^+}, x_{s_2^+}) \\ & = p(\mathcal{Y}_{a(s_1^+,s_2^+)}|x_{s_1^+}, x_{s_2^+})p(\mathcal{Y}_{d(s_1^+)}|x_{s_1^+})p(\mathcal{Y}_{d(s_2^+)}|x_{s_2^+})p(x_{s_1^+}, x_{s_2^+}) \\ & \propto p(x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s_1^+,s_2^+)})\frac{p(x_{s_1^+}|\mathcal{Y}_{d(s_1^+)})}{p(x_{s_1^+})}\frac{p(x_{s_2^+}|\mathcal{Y}_{d(s_2^+)})}{p(x_{s_2^+})} \end{aligned} \tag{3.11}$$

Similar to Eq. 3.10, $p(x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s_1^+,s_2^+)})$ on the right-hand side of Eq. 3.11 is calculated through a top-down recursion as below.

$$\begin{aligned} p(x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s_1^+,s_2^+)}) \propto \sum_{x_s} & p(\boldsymbol{y}_{s_1^+}|x_{s_1^+})p(\boldsymbol{y}_{s_2^+}|x_{s_2^+}) \\ & p(x_{s_1^+}, x_{s_2^+}|x_s)p(x_s|\mathcal{Y}_{a(s)}) \end{aligned} \tag{3.12}$$

The factors $p(x_{s_1^+}|\mathcal{Y}_{d(s_1^+)})$ and $p(x_{s_2^+}|\mathcal{Y}_{d(s_2^+)})$ in Eq. 3.11 are called *bottom-up posteriors* as they are calculated based on posteriors of their descendant nodes. For each node s in the graph, the bottom-up posterior is written as

$$\begin{aligned} p(x_s|\mathcal{Y}_{d(s)}) \propto \sum_{x_{s_1^+}, x_{s_2^+}} & p(\boldsymbol{y}_{s_1^+}|x_{s_1^+})p(\boldsymbol{y}_{s_2^+}|x_{s_2^+}) \\ & p(x_{s_1^+}|\mathcal{Y}_{d(s_1^+)})p(x_{s_2^+}|\mathcal{Y}_{d(s_2^+)})p(x_s|x_{s_1^+}, x_{s_2^+}). \end{aligned} \tag{3.13}$$

Derivations of Eq. 3.10, 3.12 and 3.13 are included in Appendix 3.6.

Making use of Eq. 3.7, 3.8, 3.9 and 3.11, the node's posterior in Eq. 3.6, given all the observations, is written as follows.

$$\begin{aligned} p(x_s|\mathbb{Y}) \propto \sum_{x_{s_1^+}, x_{s_2^+}} & \frac{p(x_s, x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s)})}{\sum_{x'_s} p(x'_s, x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s)})} \\ & p(x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s_1^+, s_2^+)}) \frac{p(x_{s_1^+}|\mathcal{Y}_{d(s_1^+)})}{p(x_{s_1^+})} \frac{p(x_{s_2^+}|\mathcal{Y}_{d(s_2^+)})}{p(x_{s_2^+})} \end{aligned} \tag{3.14}$$

Equation 3.14 calculates the posterior at each node s using three marginal posteriors $p(x_s, x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s)})$, $p(x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s_1^+, s_2^+)})$ and $p(x_s|\mathcal{Y}_{d(s)})$, in Eq. 3.9, 3.12 and 3.13. Each term is calculated through either a top-down or a bottom-up recursion. The inference is summarized in Algorithm 1. Note that $\mathcal{R}$ and $\mathcal{L}$ denote the set of root nodes and the leaf node in the graph, respectively.

3.4 Experiments and results

3.4.1 General experimental design

We evaluated the proposed inference algorithm and compared it to trees by classifying synthetic data generated using ancestral sampling. After quantifying the performance of the superpixel generation, two fluorescent microscopy image datasets were used for evaluating our multi-class segmentation method. The results were compared to SegNet [78], DeepLab [194], and PSPNet [195] as instances of deep convolutional neural networks introduced for multi-label image segmentation. We also compared segmentation using both trees and polytrees on the real image datasets to explore how changing the direction of edges and therefore the use of two-wise priors instead of three-wise priors affect the results. For infer-

Algorithm 1 Label inference on polytrees

```
□ Preliminary pass. This initial upward recursion computes prior marginals for each node. The parameters $p(x_s|x_{s_1^+}, x_{s_2^+})$ are set based on problem the model represents, as explained in Fig. 3.6 and section 3.3.4.
for all $s \in \mathcal{R}$ do
    $p(x_s) = \frac{1}{|\Lambda|}$
end for
for all $s \notin \mathcal{R}$ do
    $p(x_s) = \sum_{x_{s_1^+},x_{s_2^+}} p(x_s|x_{s_1^+}, x_{s_2^+})p(x_{s_1^+})p(x_{s_2^+})$
    $p(x_{s_1^+}, x_{s_2^+}|x_s) = \frac{p(x_s|x_{s_1^+},x_{s_2^+})p(x_{s_1^+})p(x_{s_2^+})}{p(x_s)}$
end for
△ Bottom-up pass. Upward recursion for calculating bottom-up posteriors of nodes.
for all $s \in \mathcal{R}$ do
    $p(x_s|\mathcal{Y}_{d(s)}) = p(x_s)$
end for
for all $s \notin \mathcal{R}$ do
    $p(x_s|\mathcal{Y}_{d(s)}) \propto \sum_{x_{s_1^+},x_{s_2^+}} p(\boldsymbol{y}_{s_1^+}|x_{s_1^+})p(\boldsymbol{y}_{s_2^+}|x_{s_2^+})$
    $p(x_{s_1^+}|\mathcal{Y}_{d(s_1^+)})p(x_{s_2^+}|\mathcal{Y}_{d(s_2^+)})p(x_s|x_{s_1^+}, x_{s_2^+})$
end for
∇ Top-down pass. Downward recursion for calculating top-down posteriors and calculation of complete posteriors from marginal posteriors.
if $s = \mathcal{L}$ then
    $p(x_s|\mathcal{Y}_{a(s)}) = p(x_s|\boldsymbol{y}_s) \propto p(\boldsymbol{y}_s|x_s)p(x_s)$
end if
for all $s \neq \mathcal{L}$ do
    $p(x_s|\mathcal{Y}_{a(s)}) \propto \sum_{x_{s^-},x_{s'}} p(\boldsymbol{y}_s|x_s)p(\boldsymbol{y}_{s'}|x_{s'})p(x_{s'}|\mathcal{Y}_{d(s')})$
    $\frac{p(x_s,x_{s'}|x_{s^-})}{p(x_{s'})}p(x_{s^-}|\mathcal{Y}_{a(s^-)})$
    $p(x_s, x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s)}) = p(x_{s_1^+}, x_{s_2^+}|x_s)p(x_s|\mathcal{Y}_{a(s)})$
    $p(x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s_1^+,s_2^+)})$
    $\propto \sum_{x_s} p(\boldsymbol{y}_{s_1^+}|x_{s_1^+})p(\boldsymbol{y}_{s_2^+}|x_{s_2^+})p(x_{s_1^+}, x_{s_2^+}|x_s)p(x_s|\mathcal{Y}_{a(s)})$
end for
```

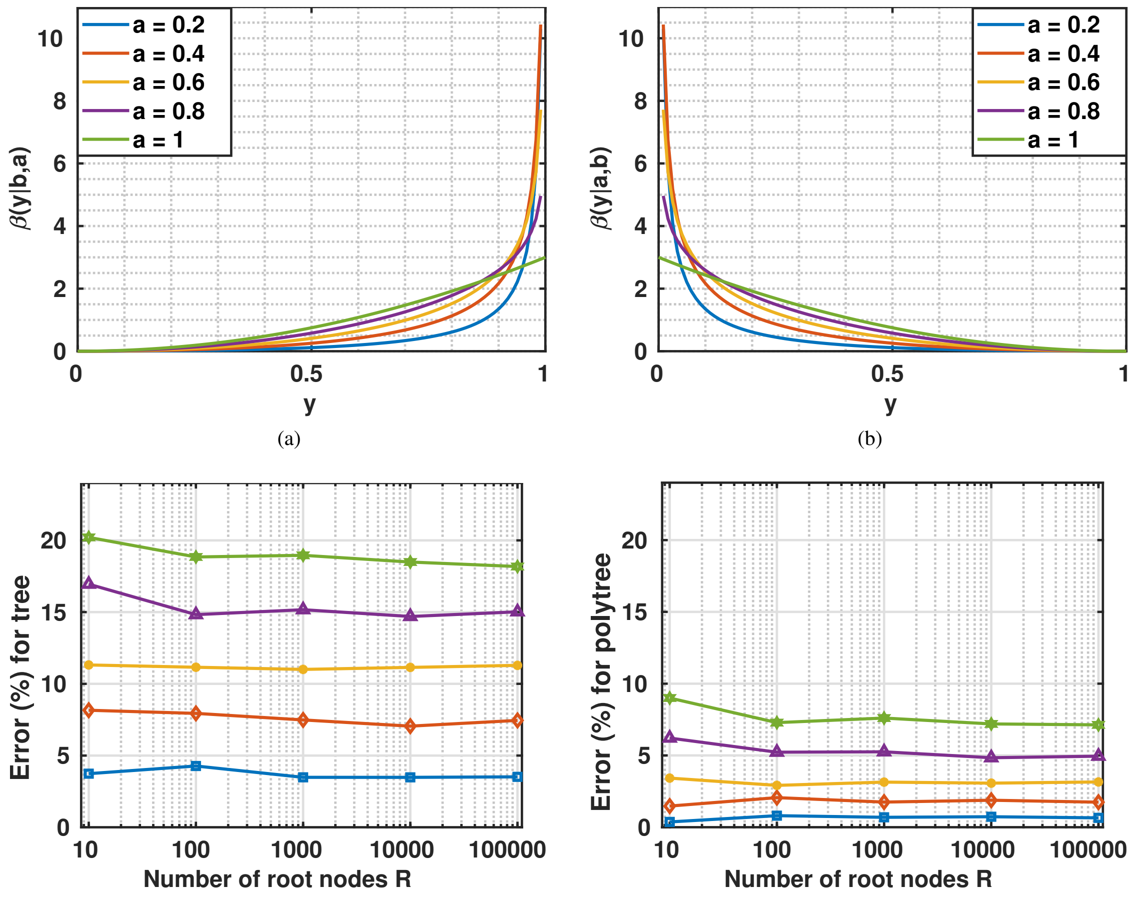
ring posteriors on trees, we adapted Laferte *et al.* [167] formulation into the graphs generated in this work.

3.4.2 Validation of the inference algorithm: ancestral sampling

To assess the performance of the inference algorithm, regardless of the image processing tools employed, we compared polytrees to trees on the classification of synthetic data generated by ancestral sampling technique. Samples are drawn for x_s variables to represent ground truth data. Based on this, the $\boldsymbol{y}_s$ variables are then drawn according to the presumed class-conditional distributions. Next, ignoring the reference x_s variables of the first step, new values are inferred for x_s from the observed $\boldsymbol{y}_s$ variables only. We then compare the inferred x_s variables to the ground truth and experimentally validate the viability of our inference algorithm.

To draw samples $\hat{x}_1, \hat{x}_2, ..., \hat{x}_N$ from the joint distribution $p(\mathbb{X}, \mathbb{Y})$, we first sample from the probability distribution $p(x_s)\big|_{s\in\mathcal{R}}$ for all root nodes. Visiting each internal node in an upward recursion, we sample from the conditional distribution $p(x_s|x_{s_1^+}, x_{s_2^+})$, where the parent labels $\hat{x}_{s_1^+}$ and $\hat{x}_{s_2^+}$ have been sampled in previous steps. Once we have sampled from the leaf node of the graph, $\hat{x}_N$, we will have obtained a sample from the joint distribution $p(\mathbb{X}, \mathbb{Y})$.

In this section only, we considered two classes for x_s for simplicity and selected y_s from the continuous range of $[0, 1]$. class-conditional likelihood functions, $p(y_s|x_s)$ were Beta distributions. For different numbers of root nodes ranging from 10 to 100000 (i.e. 19 to 199999 nodes in total as the graph is binary), graphs with random structures were generated and labels were inferred. Figures 3.9a and 3.9b show Beta distributions for different selectivities. Figures 3.9c and 3.9d depict the percentages of the wrongly inferred labels for different graph sizes and the corresponding Beta distributions in directed trees and polytrees, respectively. Results show that polytrees achieve higher accuracies in predicting labels of graph nodes, compared to directed trees. This experiment shows that even with significant overlaps between the likelihoods of two classes, where $a > 0.8$, the polytree inference error is stable and small (i.e., less than 10%). Therefore, this experiment verifies the correctness of the developed derivations and also indicates that inference accuracy increases with the selectivity of the likelihood functions.



(a)

(b)

(c)

(d)

Figure 3.9: Panels (a) and (b) show Beta distributions used as class-conditional likelihood functions in ancestral sampling. The value of b was fixed and curves correspond to the values of a ranging from 0.2 to 1, respectively, with an increasing overlap on the likelihoods (thus potential classification errors). In (c) and (d), the percentages of wrongly inferred labels using ancestral sampling are shown for tree and polytree models, respectively.

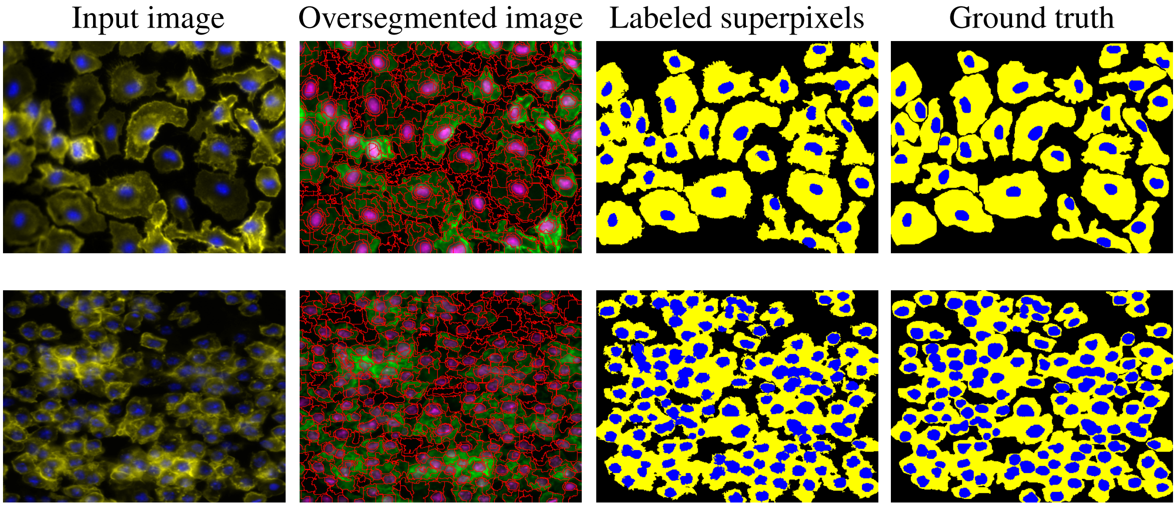


Figure 3.10: Evaluating the performance of the oversegmentation. First and second rows show the superpixels and the best possible labeling of the image using the generated superpixels, for two samples from BBBC020 and BBBC007 datasets, respectively. The finest superpixels were not shown in the oversegmented images for a better visualization.

3.4.3 Oversegmentation performance evaluation

The SEEDS oversegmentation algorithm [192], used for generating superpixels, finds areas in the image based on intensity homogeneity and boundary smoothness. Ideally, all of the object and within-object boundaries should lie on superpixel boundaries. However, due to the existence of noise and illumination artifacts in the images, not all the superpixels accurately resemble boundaries. To investigate the error introduced by oversegmentation, we labeled the superpixels in the image merely according to the ground truth, to calculate the maximum achievable segmentation accuracy for the segmentation algorithms employing SEEDS. To do this, the label of each superpixel was set based on the labels of the majority of its pixels in the Ground Truth. Figure 3.10 shows two samples from BBBC020 and BBBC007 datasets, for which the oversegmented image and the labeled superpixels can be compared to the ground truth. The Dice similarity coefficients (DSC) between the labeled superpixels and the ground truth in Fig. 3.11 quantitatively show the maximum segmentation accuracy that the graph-based algorithms employing superpixels in this work can achieve for the two datasets.

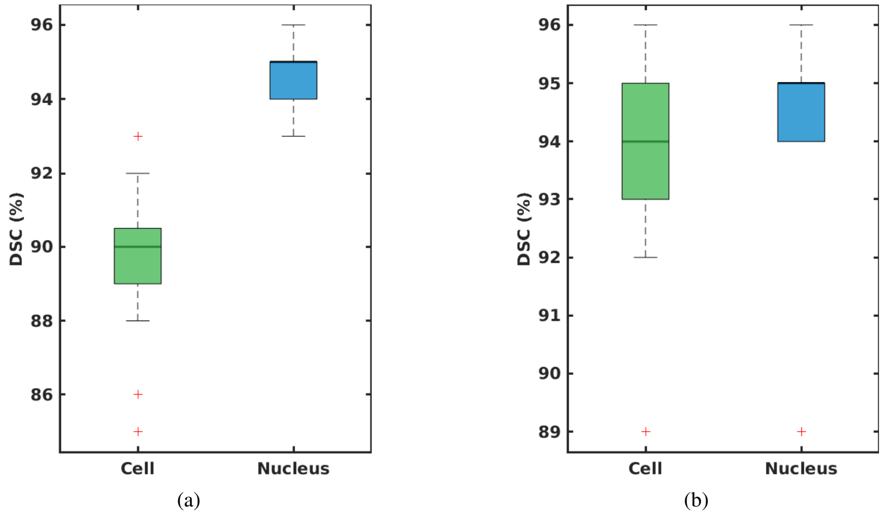


Figure 3.11: Dice similarity coefficients between the labeled superpixels and the ground truth on (a) BBBC020 and (b) BBBC007 datasets. These values show the accuracy of the SEEDS algorithm [192] in generating superpixels.

3.4.4 Validation on multi-class image segmentation

The proposed algorithm was applied to the problem of supervised multi-class image segmentation, and to evaluate the role of exploiting prior knowledge in segmentation. Two real image datasets were chosen from the publicly available datasets on Broad Bioimage Benchmark Collection that contain two-channel fluorescence microscopy images with cells and nuclei, namely BBBC020 and BBBC007 datasets [25]. In these cases, between-class relationships can help to improve the segmentation results, as only a certain set of label configurations are plausible. The results of this experiment were compared to those of SegNet, DeepLab, and PSPNet.

BBBC020 contains 20 two-channel *in vitro* microscopy images of murine bone marrow macrophages, and BBBC007 has 16 two-channel *in vitro* microscopy images of drosophila Kc167 cells. Manual annotations are available for both datasets. These two datasets have the same type of images and define similar multi-class segmentation problems of cells and nuclei. The BBBC007 dataset has a larger number of overlapping cells and noisier images, which makes the segmentation more challenging. See Fig. 3.12 for samples from the two datasets.

To explore the role of features used for inference, we used two types of features: 1) scale-space first and second order differential invariants [198], 2) deep representations extracted by SegNet. In the following, details of experiments with the two feature sets are explained and results are compared to the three convolutional neural networks. The accuracy of the segmentation was measured by calculating confusion matrices and the Dice similarity coefficients [199] computed by comparing the segmentation results with the ground truth.

3.4.5 Polytree with scale-space differential invariant features

In this experiment, features were chosen to be intensity value, the absolute value of the gradient, and determinants and traces of the Hessian matrix at 7 scales, for each microscopy channel. A total of 32 features were initially calculated for each image, out of which the most relevant features were selected using the Fisher discriminant score [66]. Fisher scores, W_d, are weights with higher values for features that have higher discrimination abilities and are calculated as follows.

$$W_d = \frac{\sum_{c=1}^{K}(m_d - m_{d,c})^2}{\sum_{c=1}^{K} s_{d,c}^2} \frac{K}{K-1} \tag{3.15}$$

Where d is the index of the feature, K is the total number of classes, m_d is the mean of d^{th} feature over the training images. $m_{d,c}$ and $s_{d,c}$ denote the mean and standard deviation of d^{th} feature within samples of c^{th} class, respectively.

For each dataset, four images were used for feature selection through ranking features based on their Fisher scores. The rest of the images were used for cross-validation, i.e. 4- and 6-fold cross-validations were applied on the 16 and 12 remaining images in BBBC020 and BBBC007 datasets, respectively. The four images used for Fisher score calculation were always included in the training sets during cross-validation.

Once the Fisher scores were calculated, features were ranked for each class separately, and the first F of them were selected for classification. Gaussian distributions were used for class-conditional likelihood functions with a layer dependent variance that allows higher within-class variances for nodes in the higher levels of the graph. Parameters of the method, including β_s (explained in section 3.3.2), number of intensity features used for graph genera-

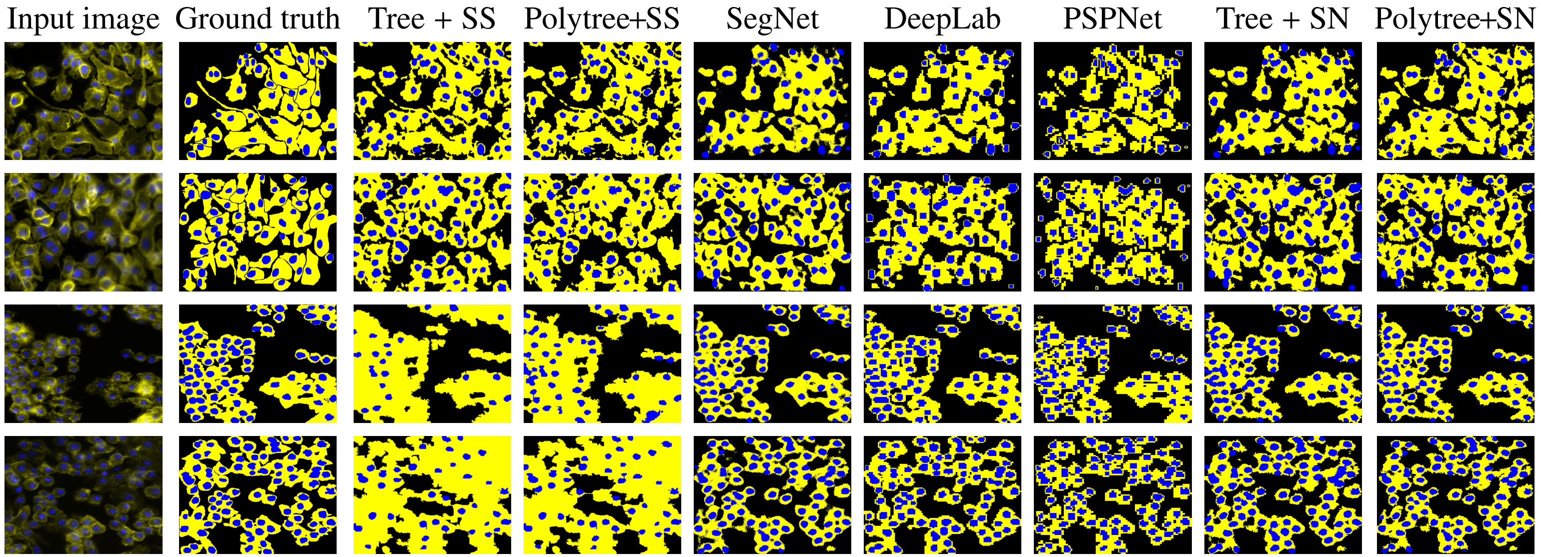


Figure 3.12: Sample images from BBBC020 (first and second rows) and BBBC007 (third and fourth rows), their corresponding ground truth and automatic segmentations. Third and fourth columns show segmentation results using trees and polytrees with scale-space (SS) features (section 3.4.5), respectively. Fifth, sixth and seventh columns show results of applying SegNet, DeepLab, and PSPNet to the images, respectively. The last two columns depict segmentation results using directed trees and polytrees with features generated by SegNet, labeled Tree + SN and Polytree + SN, respectively.

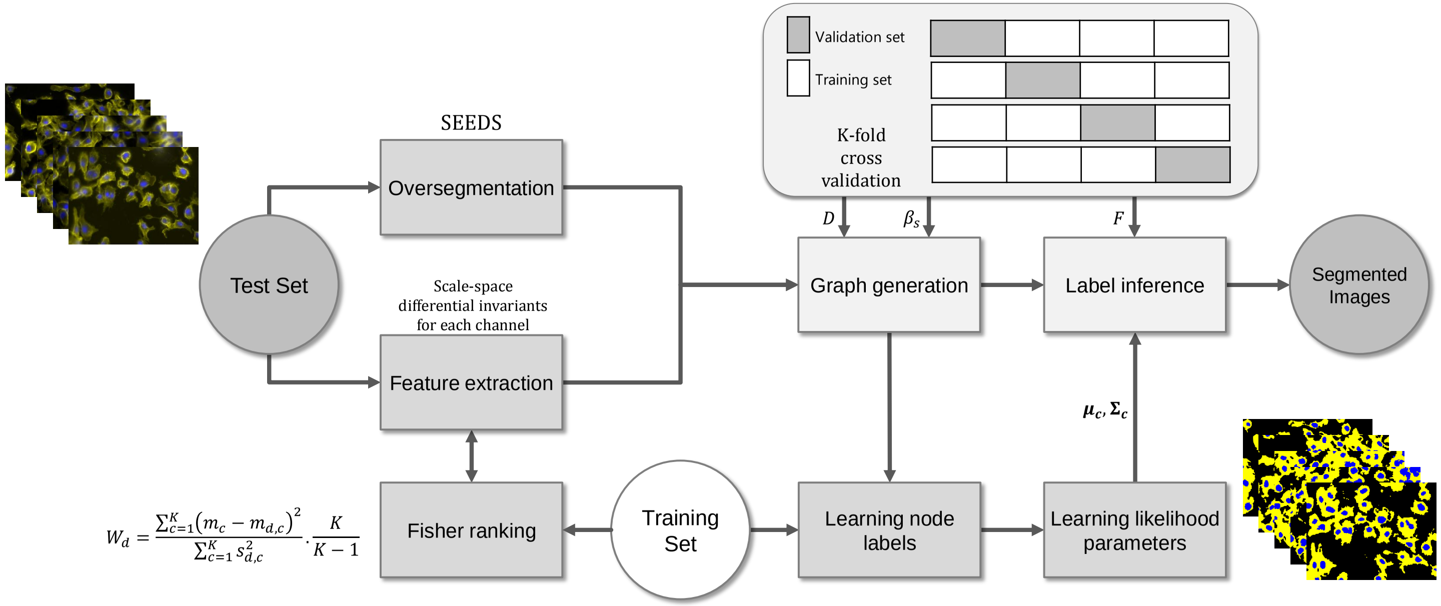
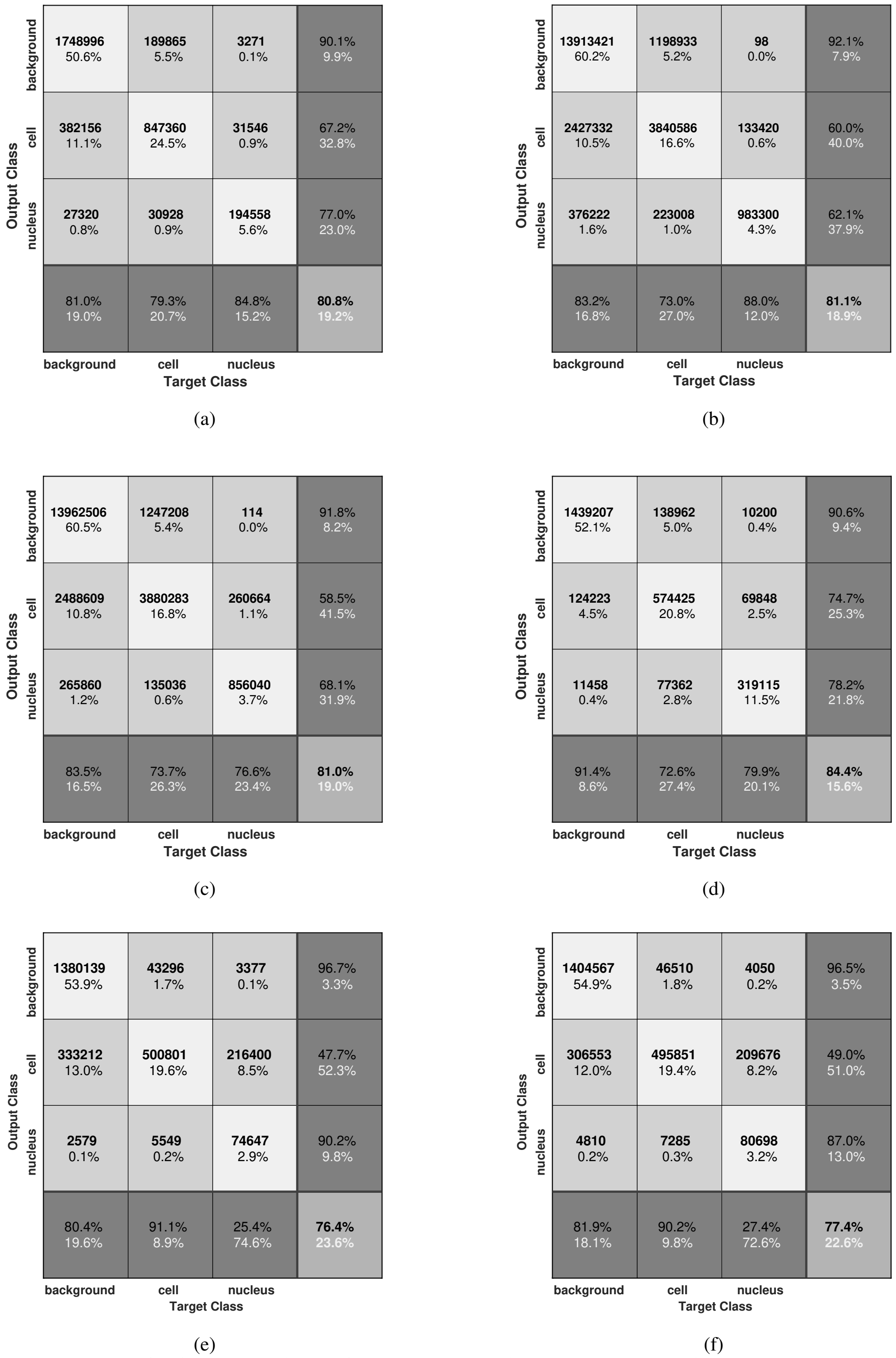

Validation set
Training set
K-fold cross validation
SEEDS
Oversegmentation
Test Set
Scale-space differential invariants for each channel
Feature extraction
D
β_s
F
Graph generation
Label inference
Segmented Images
$\boldsymbol{\mu_c}, \boldsymbol{\Sigma_c}$
$W_d = \frac{\sum_{c=1}^{K}(m_c - m_{d,c})^2}{\sum_{c=1}^{K} s_{d,c}^2} \cdot \frac{K}{K-1}$
Fisher ranking
Training Set
Learning node labels
Learning likelihood parameters

Figure 3.13: Block diagram for polytree and tree segmentation with scale-space differential invariant features.



	background	cell	nucleus	
background	**1748996** 50.6%	**189865** 5.5%	**3271** 0.1%	90.1% 9.9%
cell	**382156** 11.1%	**847360** 24.5%	**31546** 0.9%	67.2% 32.8%
nucleus	**27320** 0.8%	**30928** 0.9%	**194558** 5.6%	77.0% 23.0%
	81.0% 19.0%	79.3% 20.7%	84.8% 15.2%	**80.8%** **19.2%**

(a)

	background	cell	nucleus	
background	**13913421** 60.2%	**1198933** 5.2%	**98** 0.0%	92.1% 7.9%
cell	**2427332** 10.5%	**3840586** 16.6%	**133420** 0.6%	60.0% 40.0%
nucleus	**376222** 1.6%	**223008** 1.0%	**983300** 4.3%	62.1% 37.9%
	83.2% 16.8%	73.0% 27.0%	88.0% 12.0%	**81.1%** **18.9%**

(b)

	background	cell	nucleus	
background	**13962506** 60.5%	**1247208** 5.4%	**114** 0.0%	91.8% 8.2%
cell	**2488609** 10.8%	**3880283** 16.8%	**260664** 1.1%	58.5% 41.5%
nucleus	**265860** 1.2%	**135036** 0.6%	**856040** 3.7%	68.1% 31.9%
	83.5% 16.5%	73.7% 26.3%	76.6% 23.4%	**81.0%** **19.0%**

(c)

	background	cell	nucleus	
background	**1439207** 52.1%	**138962** 5.0%	**10200** 0.4%	90.6% 9.4%
cell	**124223** 4.5%	**574425** 20.8%	**69848** 2.5%	74.7% 25.3%
nucleus	**11458** 0.4%	**77362** 2.8%	**319115** 11.5%	78.2% 21.8%
	91.4% 8.6%	72.6% 27.4%	79.9% 20.1%	**84.4%** **15.6%**

(d)

	background	cell	nucleus	
background	**1380139** 53.9%	**43296** 1.7%	**3377** 0.1%	96.7% 3.3%
cell	**333212** 13.0%	**500801** 19.6%	**216400** 8.5%	47.7% 52.3%
nucleus	**2579** 0.1%	**5549** 0.2%	**74647** 2.9%	90.2% 9.8%
	80.4% 19.6%	91.1% 8.9%	25.4% 74.6%	**76.4%** **23.6%**

(e)

	background	cell	nucleus	
background	**1404567** 54.9%	**46510** 1.8%	**4050** 0.2%	96.5% 3.5%
cell	**306553** 12.0%	**495851** 19.4%	**209676** 8.2%	49.0% 51.0%
nucleus	**4810** 0.2%	**7285** 0.3%	**80698** 3.2%	87.0% 13.0%
	81.9% 18.1%	90.2% 9.8%	27.4% 72.6%	**77.4%** **22.6%**

(f)

Figure 3.14: Confusion matrices for SegNet with augmented images, tree and polytree segmentations with scale-space differential invariants on the two real datasets. The overall accuracies of tree (b) and polytree (c) were slightly higher than SegNet (a) on the BBBC020 dataset, while SegNet (d) outperforms tree (e) and polytree (f) on the BBBC007 dataset. The Number of pixels corresponding to each percentage is shown in bold. Black and white percentages in each box show the proportion of correctly and incorrectly classified pixels, respectively.

tion (D) and inference (F), and values of mean ($\boldsymbol{\mu}_c$) and covariance matrix ($\boldsymbol{\Sigma}_c$) for each class c are optimized through cross-validation for each of the two datasets. Figure 3.13 shows the block diagram of polytree and tree based segmentation using scale-space differential invariant features.

We applied SegNet to the two datasets and compared the results with polytree and tree segmentation using scale-space differential invariants. As the size of the datasets was not sufficiently large for training SegNet, random elastic deformations of the training images and their annotations were added to the training sets during each cross-validation experiment. This way, the size of the training sets for each experiment on the two datasets was increased to 400 images (chosen based on experiments with different numbers of augmented images) to improve shift and rotation invariance, and robustness to deformations and gray value variations [79, 200]. Furthermore, 5000 iterations were performed for the experiments on the two datasets with the cost function reaching its minimum after about 1000 iterations. The trained network was then evaluated on its segmentation of the test set. Figure 3.14 shows the confusion matrices for SegNet, tree and polytree segmentations of BBBC020 and BBB007 datasets. The overall segmentation accuracies are similar for the three methods on BBBC020 dataset, while SegNet outperforms the other two on BBBC007. Dice similarity coefficients (DSC) in Fig. 3.15 indicate SegNet is more accurate than tree and polytree in both classes on the BBBC007, while tree and polytree provide higher DSC values for the segmentation of cells in BBBC020. DSC values for the segmentation of nuclei in BBBC020 are similar for SegNet and tree, being more accurate than that for polytree.

This experiment indicates outperformance of SegNet in segmentation. However, the three methods were compared using different experimental setups. First, SegNet was trained using a larger set of training images (through augmentation). The numbers of features (F) selected after ranking them based on the Fisher scores were 20 and 6, for BBBC020 and BBBC007 datasets, respectively, which are very small compared to the number of features extracted by SegNet. To investigate the methods regardless of the type of features used, we propose the use of polytrees with features employed by SegNet in the next section.

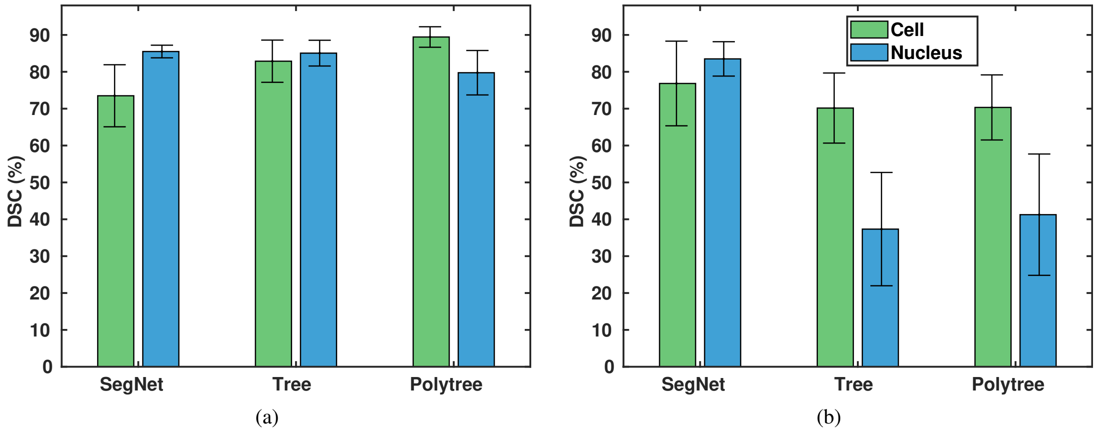


Figure 3.15: Dice similarity coefficients (DSC) of polytree and tree based segmentations using scale-space differential invariant features compared to SegNet on (a) BBBC020 and (b) BBBC007 datasets.

3.4.6 Polytree with SegNet-based deep features

To compare the developed polytree inference and SegNet using similar preprocessing, feature extraction and selection and training size, we developed a framework to employ features calculated by SegNet, shown in Fig. 3.16. In this section, we have also applied directed trees with SegNet features to the segmentation of images in the two datasets. The directed tree was generated by reversing the directions of edges on the irregular polytree and the inference proposed by Laferte *et al.* was adapted to it. Softmax [66] functions were chosen as posteriors.

$$p(x_s = c|\boldsymbol{y}_s) \propto \frac{\exp(\boldsymbol{w}_c^T \boldsymbol{y}_s)}{\sum_k^K \exp(\boldsymbol{w}_k^T \boldsymbol{y}_s)} \tag{3.16}$$

In Eq. 3.16, $\boldsymbol{w}_k$'s are the vectors of weights for each class k, calculated by the CNN to describe the distribution of each class, and K denotes the total number of classes ($K = 3$ in our case).

Note that Eq. 3.16 implies that a set of improper (unnormalized) class-conditional likelihoods, i.e. exponentials, have been used. However, looking at Algorithm 1, the proposed inference algorithm normalizes every term that contains the likelihood probability of nodes,

Table 3.2: Mean Dice score coefficients of the five methods on BBBC020 and BBBC007 datasets.

Dataset		BBBC020	
Number of images	20	200	400
Polytree	**78.60** ± **5.42**	80.43 ± 4.76	81.35 ± 5.18
Directed tree	78.45 ± 5.39	80.52 ± 4.82	**81.45** ± **5.21**
SegNet	77.00 ± 5.28	79.42 ± 4.71	80.40 ± 5.06
DeepLab	78.17 ± 3.73	**81.35** ± **2.60**	81.37 ± 2.98
PSPNet	76.72 ± 3.14	78.35 ± 3.30	78.27 ± 3.13
Dataset		BBBC007	
Number of images	16	200	400
Polytree	**80.28** ± **8.44**	**82.09** ± **7.46**	**83.06** ± **6.85**
Directed tree	79.65 ± 10.62	81.00 ± 10.40	81.75 ± 9.64
SegNet	77.40 ± 8.83	80.06 ± 7.79	81.03 ± 7.43
DeepLab	79.96 ± 5.97	80.56 ± 5.95	80.75 ± 4.95
PSPNet	78.37 ± 4.84	77.56 ± 4.50	77.37 ± 4.71

facilitating the utilization of unnormalized likelihood functions. For this reason, we chose exponentials as the likelihood functions, i.e. $p(\boldsymbol{y}_s|x_s = c) \propto exp(\boldsymbol{w}_c^T \boldsymbol{y}_s)$. Both of the class parameters ($\boldsymbol{w}_c$) and feature vectors ($\boldsymbol{y}_s$) are provided by the SegNet. Therefore, having trained the SegNet, we do not require any additional training steps.

In this section, we compared the results of the proposed polytree and tree methods with SegNet, Deeplab, and PSPNet. In applying the CNNs to BBBC020 and BBBC007 datasets, the same image augmentation procedure as explained in section 3.4.5 was employed. Segmentation performance of the methods was compared at three different sizes of datasets; original dataset size (20 images for BBBC020 and 16 images for BBBC007), 200, and 400 augmented images. In each of the experiments, a four-fold cross-validation was done. To perform a cross-validation, the augmented images were generated based only on the images in the training folds, so that the network was trained independently of the testing set.

For these experiments, the images were first oversegmented using the SEEDS algorithm [192]. The features provided by SegNet were then used for graph generation and, in the next step, for label inference ($F = D$).

Figure 3.17 shows the DSC of the five methods when SegNet features are used for poly-

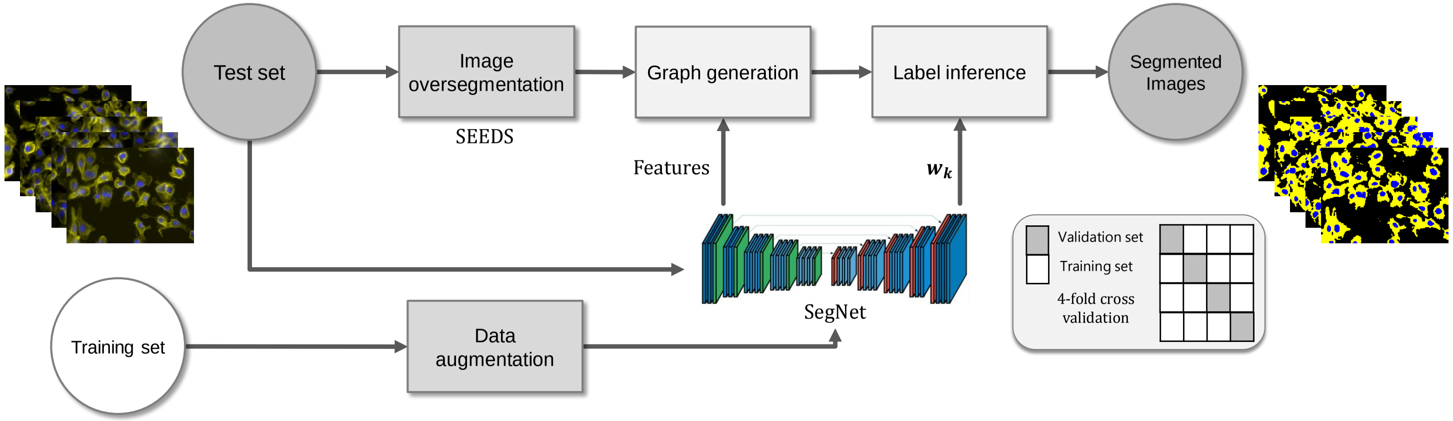
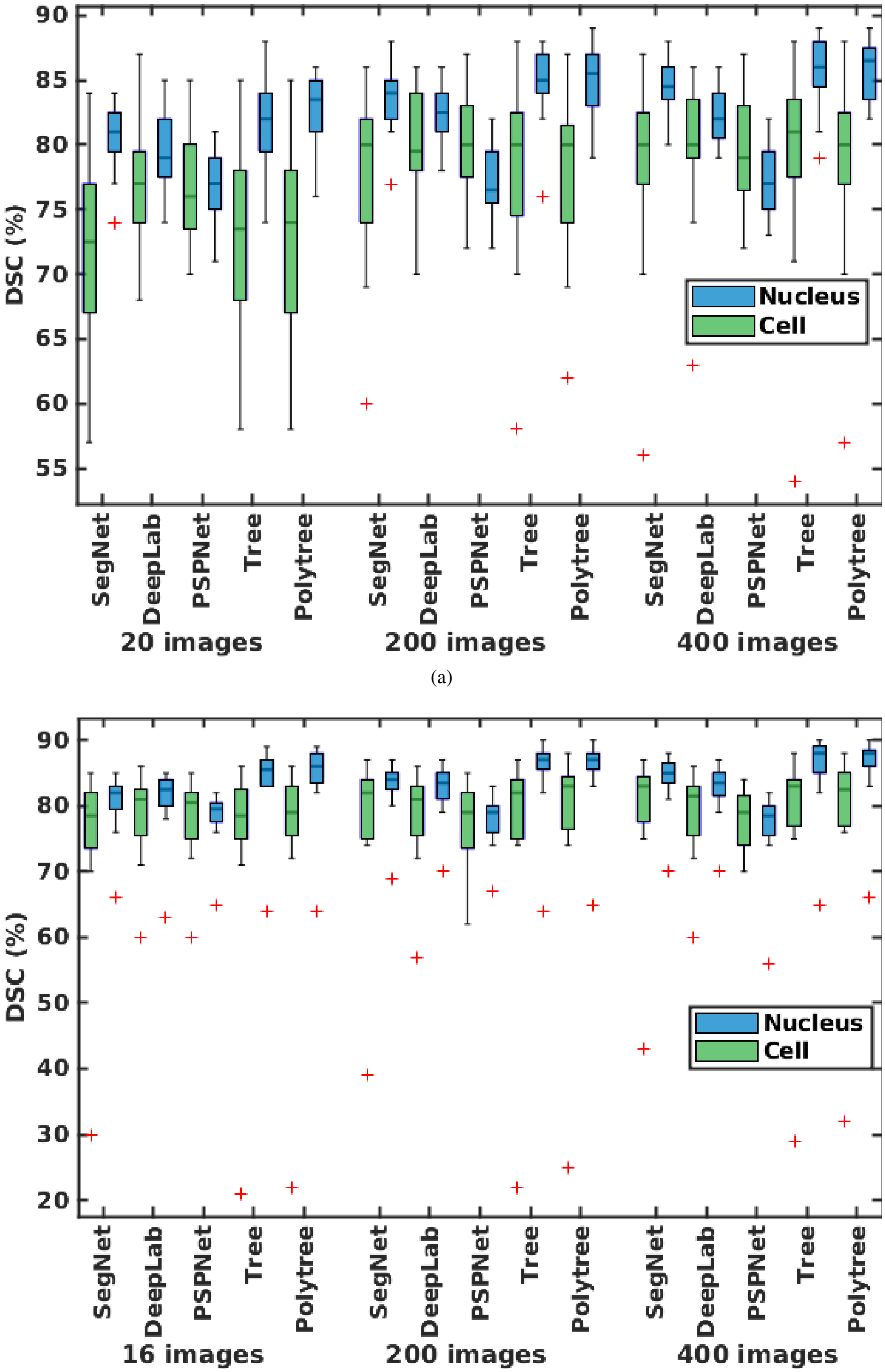


Figure 3.16: The proposed architecture for using SegNet-based deep features and learning class-conditional likelihood functions.



(a)

(b)

Figure 3.17: Dice similarity coefficients of the five methods for segmenting cells and nuclei in (a) BBBC020 and (b) BBBC007 datasets, respectively.

tree and tree with variable numbers of the training samples. Table 3.2 shows average accuracy values for each of the five methods and for each size of the training set for BBBC020 and BBBC007 datasets. The superior results of the directed tree and polytree indicate the effectiveness of the prior knowledge imposed by these directed graphical models, which cannot be explicitly modeled by CNNs. It can also be seen that the performance of directed trees tend to have larger variances compared to polytrees. This higher uncertainty is likely to stem from the inability of directed trees to eliminate unfeasible label configurations, eliminated by polytrees, that allows semantically wrong segmentations (see section 3.3.4).

To assess the complexity of the segmentation algorithm, graph generation and Bayesian inference stages were timed for graphs ranging from 20 to 200000 nodes. Results show that the time of run, t, on a machine equipped with Intel Xeon(R) CPU E5620 2.40GHz and 32GB of RAM, using Ubuntu 14.04, scales with the number of graph nodes, n, with $t = 5 \times 10^{-5} n^{1.3}$ and $t = 2.4 \times 10^{-6} n^{2}$ for graph generation and inference, respectively. This shows a sustainable scalability of the proposed algorithm with increasing the number of nodes.

3.4.7 Prediction of segmentation error

Unlike discriminative models, generative models incorporate priors in calculating the posterior distributions. Accordingly, the proposed polytree graphical model can evaluate to what extent its estimated clique labels comply with the imposed priors. A strong disagreement can indicate an erroneous segmentation that can be flagged up for manual inspection. To implement this, the labels of cliques are read from the graph representing the segmented image and their probabilities are calculated using the constraints in Fig. 3.6b. Areas in the image that correspond to cliques with unfeasible labels (zero probabilities) are then marked as potential segmentation errors. Figure 3.18 shows samples from BBBC020 and BBBC007 and the error predicted for them. To represent the confidence of the model in labeling the wrongly segmented areas, they are marked by red colors with different values, corresponding to the entropy of the joint posterior of the clique. Areas with lower and higher error likelihoods (entropies) are shown in lighter and darker colors, respectively.

The error prediction ability of the directed trees was also evaluated. Figure 3.19 shows Dice similarity coefficients between the potentially incorrectly segmented areas and the ac-

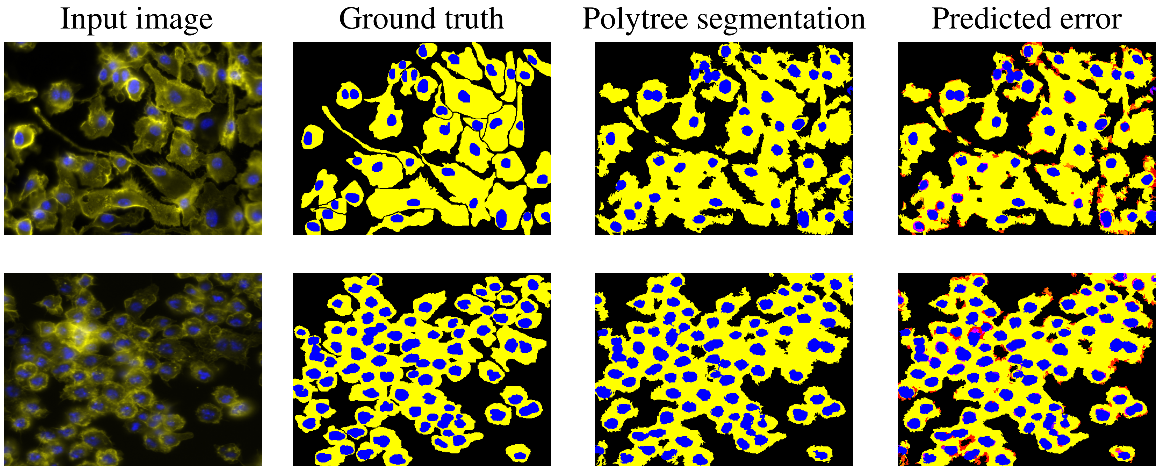


Figure 3.18: The ability of the proposed method in nominating the possibly wrongly segmented areas shown for samples from BBBC020 (first row) and BBBC007 (second row) datasets. Value of red color is proportional to the probability of being an error in the segmentation.

tual segmentation error for both methods. Figure 3.20 shows the average Dice similarity coefficients for different thresholds of entropies for the models on the two datasets. These two figures indicate that polytrees are superior in predicting the segmentation error. This superiority is due to the more effective imposition of prior knowledge in polytrees compared to trees (three-wise constraints versus two-wise constraints, respectively).

3.5 Discussion and conclusions

This work proposes a new inference algorithm for multi-class segmentation using irregular directed graphical models. The image is first oversegmented and a graph is generated by recursively merging the two most similar nodes in the graph until a hierarchical graphical model is generated that has no loops. Two types of features were used in this study: 1) scale-space differential invariants of intensity and 2) SegNet-based deep image representations. This was done to investigate the dependency of the method performance on the features used. Two publicly available real microscopy image datasets were used for evaluation. We showed that our polytree based method outperforms the customized tree and three state-of-

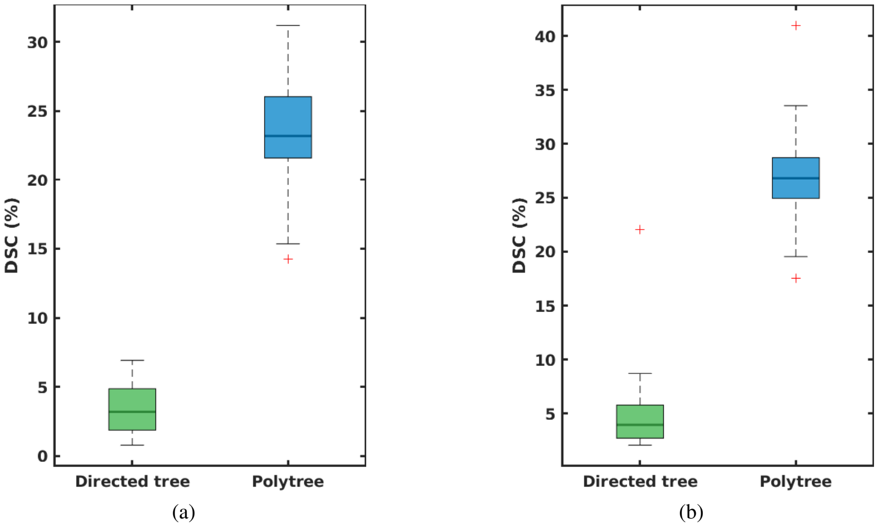


Figure 3.19: Dice similarity coefficients between the predicted and the actual segmentation error for directed trees and polytrees on (a) BBBC020 and (b) BBBC007 datasets.

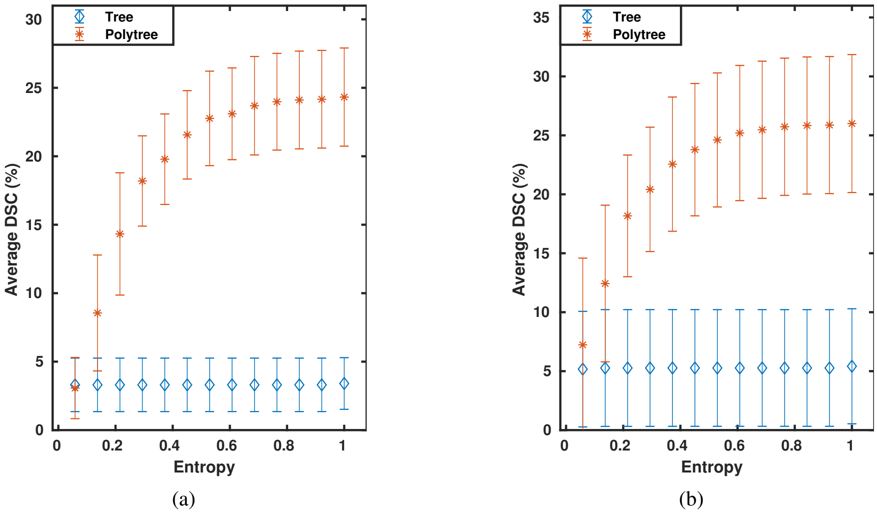


Figure 3.20: Average Dice similarity coefficients between the predicted and actual segmentation error for directed trees and polytrees at different thresholds of entropies of cliques on (a) BBBC020 and (b) BBBC007 datasets.

the-art convolutional neural networks, SegNet [78], DeepLab [194], and PSPNet [195]. The oversegmentation performance was evaluated by comparing the labeled superpixels to GT to determine the maximum achievable accuracy of the segmentation methods employing the generated superpixels using the SEEDS algorithm [192]. In terms of predicting segmentation errors, polytrees also outperformed directed trees.

In the literature, directed graphical models have been employed to incorporate prior knowledge to improve segmentation [128, 130]. However, a large majority of the works rely on directed trees, due to more simple inference and the existence of efficient closed-form solutions for posteriors. This work uses polytrees for multi-class segmentation and models more complex label dependencies between the child and parent nodes, deriving closed-form solutions for the posteriors on the polytree. The distinct orientation of edges on polytrees allows them to model label configurations for nodes in horizontal vicinity, in addition to the vertical nodes modeled by trees. This improves the compliance of the inferred labels with the imposed constraints and is a key feature of polytree, as modeling the same relations with Markov Random Fields requires graphs with loops, for which the inference is iterative and approximate. It should be noted that factor graphs [142] can also provide closed form solutions as long as the original graph structure can be converted to a factor graph without loops. However, the proposed inference method does not require the extra step for generating a second factor graph, simplifying the implementation.

Using polytrees with scale-space differential invariant features (Fig. 3.15) suggests that depending on the choice of model features and parameters, it can outperform SegNet, even though the latter is trained on a much larger training set (16 vs. 400 images). Additionally, the distinct performance of the polytree segmentation on BBBC007 dataset when different types of features were used reveals the key role of features in the segmentation performance. By using the same features of the SegNet, polytree provides a superior segmentation compared to directed trees and three CNNs (see Table 3.2). This superiority owes to the model's ability to explicitly enforce prior knowledge and to eliminate unfeasible label configurations. An example of these configurations for the problem of segmenting cells and nuclei is the existence of a cell area inside a nucleus. CNNs can also learn such dependencies through their cascade of convolutional layers. However, their efficiency relies on the quality of the training data and the existence of sufficient instances of the dependencies, which might not be possible for every dataset.

Evaluating the performance of oversegmentation shows that this stage significantly contributes to the overall segmentation error. The maximum achievable accuracies depicted in Fig. 3.11 show an upper bound for the Dice scores that could be achieved by segmentation methods employing SEEDS on the two datasets. Using other superpixel generation algorithms might address this problem by drawing superpixels with boundaries more accurately matching objects boundaries in the image.

In the current implementation of the proposed algorithm, the overall segmentation performance of the method can be confined by the graph generation quality. To address this, one line of future work can be the development of a Maximum Posterior (MAP) estimation [66] for graph generation that optimizes the graph structure jointly with label inference. On the other hand, it is worth mentioning that the small margin of improvement by the proposed graph-based segmentation over SegNet is because features learned by the CNN are minimizing the cost function of SegNet rather than the cost function of the polytree. Another line of future work can be extracting features by neural networks that are specifically minimizing the cost of polytree. In predicting the segmentation error, however, polytrees significantly outperform trees (see Fig. 3.19 and 3.20). The lower variance of the average DSC in Table 3.2 when using DeepLab and PSPNet is due to the additional network layers that that improve the localization of boundaries for a cost of adding to the computational complexity. An extension of current work can be employing deep representations extracted by these two networks with the use of polytrees for incorporating prior knowledge for possible improvements in the segmentation results.

The proposed application of the directed graphical models facilitates extracting statistics of relationships between class labels from the graph, in addition to the current use of the graph for imposing prior knowledge. For example, using the proposed method for the segmentation of host and pathogen cells, the proportions of intracellular and extracellular pathogen cells, infected and healthy host cells can be calculated from the labeled graph, both at a specific time point and over time for disease progression monitoring. Such applications introduce new capabilities of graph-based segmentation for the behavioral analysis of diseases and biological systems. Other than their use in the image analysis, polytrees can model phenomena involving the interrelations of different objects with underlying dependencies. One example can be the genetic networks where polytrees can model relationships between different entities including genes or individuals and the expression of certain genes in different generations. The inference platform presented here can be extended to the case where each node can have more than two and generally an arbitrary number of descendant nodes to improve its adaptation to the problem being modeled.

3.6 Appendix 1: Proofs of equations

- Expansion of Eq. 3.10 (top-down)

$$\begin{aligned}
p(x_s|\mathcal{Y}_{a(s)}) &\propto p(x_s, \mathcal{Y}_{a(s)}) \\
&= \sum_{x_{s^-}, x_{s'}} p(\mathcal{Y}_{a(s)}, x_{s^-}, x_s, x_{s'}) \\
&= \sum_{x_{s^-}, x_{s'}} p(\mathcal{Y}_{a(s)}|x_{s^-}, x_s, x_{s'}) p(x_{s^-}, x_s, x_{s'}) \\
&= \sum_{x_{s^-}, x_{s'}} p(\boldsymbol{y}_s|x_s) p(\boldsymbol{y}_{s'}|x_{s'}) p(\mathcal{Y}_{d(s')}|x_{s'}) p(\mathcal{Y}_{a(s^-)}|x_{s^-}) p(x_{s^-}, x_s, x_{s'}) \\
&\propto \sum_{x_{s^-}, x_{s'}} p(\boldsymbol{y}_s|x_s) p(\boldsymbol{y}_{s'}|x_{s'}) p(x_{s'}|\mathcal{Y}_{d(s')}) \frac{p(x_s, x_{s'}, x_{s^-})}{p(x_{s^-}) p(x_{s'})} p(x_{s^-}|\mathcal{Y}_{a(s^-)})
\end{aligned} \tag{3.17}$$

- Expansion of Eq. 3.12 (top-down)

$$\begin{aligned}
p(x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s_1^+, s_2^+)}) &= p(x_{s_1^+}, x_{s_2^+}|\boldsymbol{y}_{s_1^+}, \boldsymbol{y}_{s_2^+}, \mathcal{Y}_{a(s)}) \\
&= \sum_{x_s} p(x_s, x_{s_1^+}, x_{s_2^+}|\boldsymbol{y}_{s_1^+}, \boldsymbol{y}_{s_2^+}, \mathcal{Y}_{a(s)}) \\
&\propto \sum_{x_s} p(\boldsymbol{y}_{s_1^+}, \boldsymbol{y}_{s_2^+}, \mathcal{Y}_{a(s)}|x_s, x_{s_1^+}, x_{s_2^+}) p(x_s, x_{s_1^+}, x_{s_2^+}) \\
&= \sum_{x_s} p(\boldsymbol{y}_{s_1^+}|x_{s_1^+}) p(\boldsymbol{y}_{s_2^+}|x_{s_2^+}) p(\mathcal{Y}_{a(s)}|x_s) p(x_s, x_{s_1^+}, x_{s_2^+}) \\
&\propto \sum_{x_s} p(\boldsymbol{y}_{s_1^+}|x_{s_1^+}) p(\boldsymbol{y}_{s_2^+}|x_{s_2^+}) p(x_{s_1^+}, x_{s_2^+}|x_s) p(x_s|\mathcal{Y}_{a(s)})
\end{aligned} \tag{3.18}$$

- Expansion of Eq. 3.13 (bottom-up)

$$\begin{aligned}
p(x_s|\mathcal{Y}_{d(s)}) &= \sum_{x_{s_1^+},x_{s_2^+}} p(x_s, x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{d(s)}) \\
&\propto \sum_{x_{s_1^+},x_{s_2^+}} p(\boldsymbol{y}_{s_1^+}, \mathcal{Y}_{d(s_1^+)}, \boldsymbol{y}_{s_2^+}, \mathcal{Y}_{d(s_2^+)}|x_s, x_{s_1^+}, x_{s_2^+})p(x_s, x_{s_1^+}, x_{s_2^+}) \\
&= \sum_{x_{s_1^+},x_{s_2^+}} p(\boldsymbol{y}_{s_1^+}|x_{s_1^+})p(\mathcal{Y}_{d(s_1^+)}|x_{s_1^+})p(\boldsymbol{y}_{s_2^+}|x_{s_2^+})p(\mathcal{Y}_{d(s_2^+)}|x_{s_2^+})p(x_s, x_{s_1^+}, x_{s_2^+}) \\
&\propto \sum_{x_{s_1^+},x_{s_2^+}} p(\boldsymbol{y}_{s_1^+}|x_{s_1^+})p(\boldsymbol{y}_{s_2^+}|x_{s_2^+})p(x_{s_1^+}|\mathcal{Y}_{d(s_1^+)})p(x_{s_2^+}|\mathcal{Y}_{d(s_2^+)})p(x_s|x_{s_1^+}, x_{s_2^+})
\end{aligned} \tag{3.19}$$

3.7 Appendix 2: Tools employed

The main implementation platform used was C++, where various functions and classes are defined. Different modules were implemented for reading the image, calculating and ranking features, generating, writing and reading graphs, labeling the graph based on the training data, inferring posteriors on the graph and generating the segmented image according to a labeled graph. The main open-source libraries used include

- **OpenCV (Open Source Computer Vision)** [201] libraries for generating superpixels and performing matrix operations.
- **ITK (Insight Segmentation and Registration Toolkit)** [202] library functions have been used for feature extraction and mathematical operations.
- **BOOST** library [203] was used for efficient storage and loading of the graph structures.
- **FSL** [204], **GSL** (GNU Scientific Library) [205] have been used for numerical and mathematical routines.

Two types of features have been extracted from the images: 1) intensity values, the absolute value of the gradient and determinant and trace of the Hessian matrix, known as scale-space differential invariant features [198], 2) features extracted by a convolutional neural network, namely SegNet [78]. Fisher discriminant score [66] has been implemented for ranking the features and employing the most relevant features for segmentation. This reduces the unnecessary computational complexity of the method and helps to avoid overfitting.

An algorithm has been designed for graph generation, that generates a connected hierarchical model by recursively merging the two most similar superpixels in the image based on a similarity metric. An adaptive similarity metric was implemented to ensure firstly connecting subareas of the objects and then connecting objects of the same type in the hierarchy. Finally, MATLAB and Python were employed for batch execution of the binaries as well as paralleling them.

Chapter 4

Hierarchical Bayesian Networks for Weakly Supervised Cell Segmentation in Light Microscopy

4.1 Introduction

Cell image analysis is crucial in disease diagnosis and treatment, for example, abnormal nuclei shapes in histology [39], differential cell counts in blood smears [10] or the proportion of the viremic cells during infection [11, 12]. On the other hand, morphological characteristics of cells and their compartments have been shown to determine the controllability and the outcome of disease [1]. Analyzing cellular objects in an image relies on a segmentation of objects of multiple classes that usually have known interrelations, for example pathogens that can reside within or outside host cells, nuclei within cells, and sub-nuclear compartments inside cell nuclei.

Two main challenges are involved in cell segmentation. First is the limited availability of ground truth due to the increasing number of images, large number of object samples in each image, and manual annotation being arduous, time-consuming and, in turn, expensive. Second is the effective modeling of the inter-class dependencies that determines the accuracy and meaningfulness of the final segmentation. Therefore, methods that can effectively incorporate the class dependencies using a limited set of annotations are highly desirable.

In this work, four microscopy image datasets have been used for the evaluation of the proposed method. The first dataset is a synthetic dataset with 25 simulated images of HL60 cell nuclei and sub-nuclear particles, namely nucleoli, generated by the Mitogen software [26]. The second one is the zebrafish dataset having 10 random two-channel microscopy images of host and pathogen cells, viz. macrophages and the fungal pathogen *Cryptococcus neoformans* published in [1]. Thirdly, BBBC020 dataset has been used containing images depicting bone-marrow derived macrophages from C57BL/6 mice stained with DAPI to label the nuclei and CD11b/APC to label the cell surface, to investigate the role of toll-like receptors for macrophage spreading [25]. Automatic segmentation of the image data is required for measurement of experimental parameters. Finally, the ISBI2015 Challenge dataset with 900 images of cytoplasms and nuclei of cervical cells obtained from Pap smear [27] was used. Manual annotations are available for all of the four datasets. Figure 4.1 shows samples from the four datasets.

Inter-class relations have been employed in segmentation through discriminative and generative probabilistic models. Convolutional neural networks (CNNs) are an example of discriminative models that can learn dependencies between different classes through their

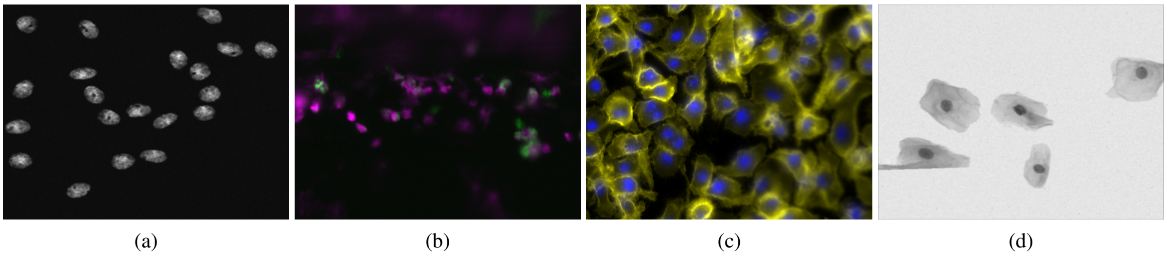


Figure 4.1: Samples from the synthetic [26] (a), zebrafish [1] (b), BBBC020 [25] (c) and ISBI2015 [27] datasets.

cascade network architecture [118, 124]. These models perform well, and rely on a set of training samples, with a sufficient quantity and variance, to learn the underlying constraints. On the other hand, generative models, such as Bayesian networks (BNs), impose the interrelations by utilizing a prior term designed based on the problem. Incorporation of prior knowledge for deriving model posteriors reduces the dependency of BNs on the training data, compared to CNNs. Directed acyclic graphs (DAGs) are a type of BNs that provide closed-form posteriors through efficient non-iterative inferences on the graph. Trees and polytrees are two common DAGs used for modeling class dependencies on the graph [206,207]. These models are similar in having a unique path between every two nodes, with the distinction that nodes in trees (except the highest-level parent node) have one parent node, while nodes in polytrees can have multiple parents.

In this paper, we propose Expectation Maximization or EM-based polytrees for the segmentation of cells of multiple classes, under weak supervision (Fig. 4.2). The polytree structure models between-class constraints on the graph labels that are efficiently and exactly inferred through a two-pass inference in the Expectation (E) step. Employing the inferred posteriors of the labels, the model parameters are optimized in the Maximization (M) step of the algorithm. E and M steps are iterated until the optimal labels for the graph are inferred. We first assess the sensitivity of EM-polytree to the training size and the initial values of model parameters. We then apply the proposed method to the segmentation of four microscopy image datasets and compare its performance to SegNet [78], EM-tree and EM-GMM. Furthermore, we show that EM-polytrees outperform EM-trees in predicting the segmentation error which has possible applications in interactive segmentation. Finally, we

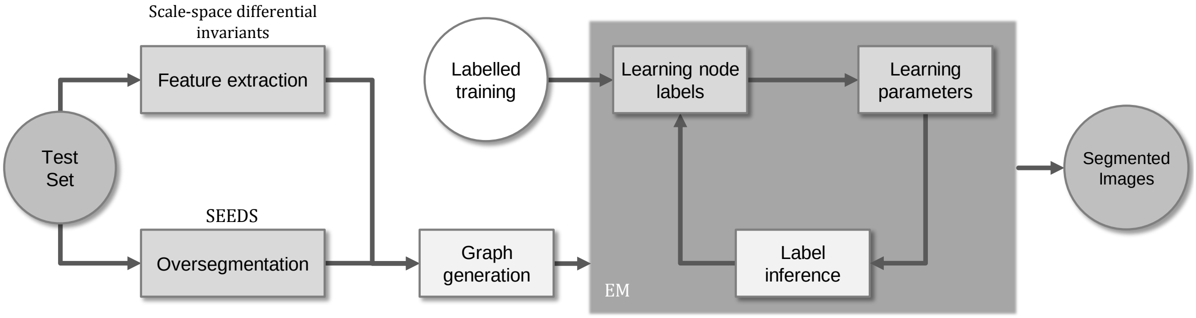


Figure 4.2: Block diagram of weakly supervised image segmentation using EM-based inference. The same oversegmentation, feature extraction and graph generation algorithms were used for EM-polytree, EM-tree, and EM-GMM. Different label inference methods were employed that utilize different inter-class dependencies.

examine the reliability of the EM-polytree segmentations by comparing the semantic measures extracted from them to the ground truth values.

4.2 Related work

Incorporating *a priori* information into segmentation generates topologically accurate and semantically meaningful segmentations by decreasing the sensitivity to image acquisition artifacts. Various types of prior information have been employed in segmentation by enforcing shape, appearance, adjacency and inclusion constraints. Shape constraints have been enforced through the classic approach of deforming a model to a target structure by regularizing deformable contours [95–97], or by fitting the current segmentation to a template to enforce the underlying structures [98, 99]. Graph-cuts were also employed to improve segmentation by imposing shape [100, 101], ordering [102, 103] and nesting [104] priors.

Interactive segmentation systems were developed that consider weaker priors during segmentation, but detect inconsistencies in the segmented images and aim to correct them [120, 121]. These methods utilize user input to encode the implications of the prior knowledge through energy terms in their cost functions [122, 123].

In neural networks, shape priors have been imposed through Shape Boltzmann Machines (SBM) [114, 115]. Successive works enforced topology and geometric constraints [116, 117] in fully convolutional networks (FCNs). Cascaded CNNs [78] were also shown to learn priors using more complex models and with increasing computational expense [118, 119]. However, these require a training of a large set of parameters, which may result in intractable training or in inaccurate segmentation.

Graphical models provide a simple way to visualize probabilistic models by expressing random variables as graph nodes and defining the relationships between them through graph edges [66]. Resolving the underflow implementation problem of the early factorization methods for Bayesian networks [166], Laferte *et al.* proposed a two-pass inference for the exact calculation of posteriors on quadtrees [167], extensions of which were used for other applications [168–170]. Combining tree-structured belief networks as a prior model with neural networks, Feng *et al.* [171] proposed a multi-label scene segmentation method. To address the block artifact of trees [172], stemming from the statistical independence of the descendant nodes in tree structures (Fig. 4.3a), more complex structures have been proposed at the expense of higher computational complexity [173,176]. Extending the inference to the graph structure, dynamic trees were introduced with structures being adapted to the images [178, 179, 181]. Incorporation of prior knowledge was also performed in a group of trees called Hierarchically-Structured Interacting Segments [183], providing approximate solutions with iterative algorithms [182, 184].

We propose a novel graph-based segmentation algorithm that can effectively incorporate prior knowledge. To ameliorate the intrinsic limitation of trees in capturing same-level dependencies, we employ polytrees (Fig. 4.3b) for multi-label segmentation. Polytree graphical models have been proposed for supervised graph-based image segmentation [193], where a hierarchical graph is generated for each image and the segmentation is obtained by labeling the graph nodes. However, that method performs *off-line* training (without optimization), requiring a set of fully segmented images, which can be difficult or impossible to obtain. Additionally, the trained classifier can be suboptimal for segmentation of new images, due to the inherent differences between training and test images. In this paper, we propose an Expectation-Maximization algorithm, that jointly calculates the labels of the nodes and estimates the parameters of the model according to a set of sparse training labels available on the graph. Depending on the segmentation problem, inclusion based constraints are imposed through setting conditional probabilities that eliminate implausible label configurations. Us-

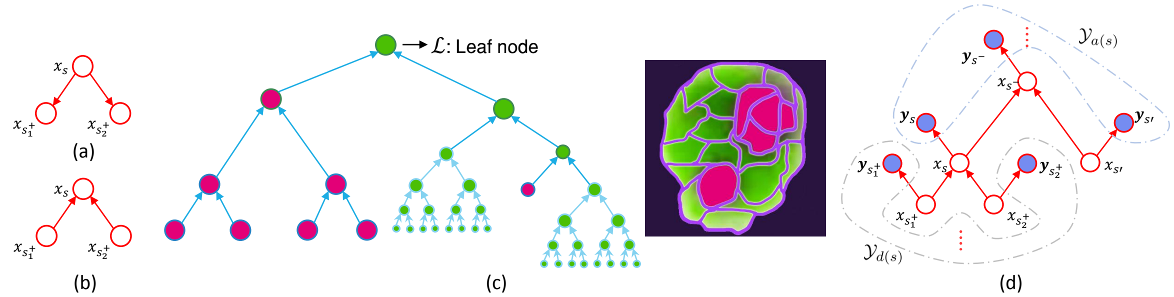


Figure 4.3: Graphical models. Panel (a) shows a clique in trees, where nodes $x_{s_1^+}$ and $x_{s_2^+}$ both have one parent, x_s. Panel (b) shows a clique in polytrees, where node x_s has two parents $x_{s_1^+}$ and $x_{s_2^+}$. A sample merge-polytree is depicted for an oversegmented image of a host cell (green) containing two pathogens (magenta), in (c). Panel (d) shows the notation of nodes neighboring node s and the graphical representations of $\mathcal{Y}_{a(s)}$ and $\mathcal{Y}_{d(s)}$ observation node sets.

ing *on-line* training, the proposed method has a lighter dependency on the training data and does not require distinct train and test phases.

4.3 Method

For image segmentation, features are first calculated and a polytree is generated by grouping similar pixels through superpixel generation, and regarding them as nodes. Given a sparse training set, an EM algorithm refines the inferred labels through optimization of model parameters. In the E step, an exact non-iterative two-pass procedure infers the posteriors of the nodes, given the parameter updates from the M step. Finally, the segmented image is constructed based on the optimal node labels. Figure 4.2 shows the block diagram of the proposed EM-based segmentation.

4.3.1 Data-driven irregular polytree

We first adopt a superpixel representation of the image, to group locally similar pixels using SEEDS algorithm [196]. These superpixels are recursively merged to obtain a hierarchy of locally coherent areas, each corresponding to a node in a merge-polytree [208] (see Fig. 4.3c). Depending on the level of a superpixel in the hierarchy, different criteria are employed for merging superpixels. Accordingly, the lower level nodes, that correspond to smaller superpixels (i.e. parts of the same objects), can only be merged with their neighboring superpixels. The higher level nodes, on the other hand, that correspond to larger superpixels (i.e. objects or groups of objects), can be merged with similar superpixels, regardless of their location in the image. This scheme makes merging neighboring nodes that correspond to parts of the same object more probable. Starting from nodes that correspond to the finest superpixels, nodes are recursively merged until a connected graph is generated. The generated graph has an irregular structure that adapts to the image leading to more naturally preserved boundaries and has no loops. We merge two nodes at each merging step that urges each non-leaf node to have two descendant nodes (Fig. 4.3b). This three-wise clique is denoted by $descendant1 - node - descendant2$.

4.3.2 Problem definition

Herewith, we describe how the posteriors of nodes are exactly computed, given the parameters of the likelihood functions. This is implemented in the E step of the proposed EM algorithm (see section 4.3.4). Sets of labels and observed features are denoted by $\mathbb{X} = \{z_s\}$ and $\mathbb{Y} = \{\boldsymbol{y}_s\}$, respectively. The set of graph nodes and edges is denoted by $\mathcal{G}$, and Λ is the set of all class labels. For an internal node s (neither in the lowest level nor the leaf node), s^-, s^+ and s' denote nodes in higher, lower and same layers, respectively (Fig. 4.3d).

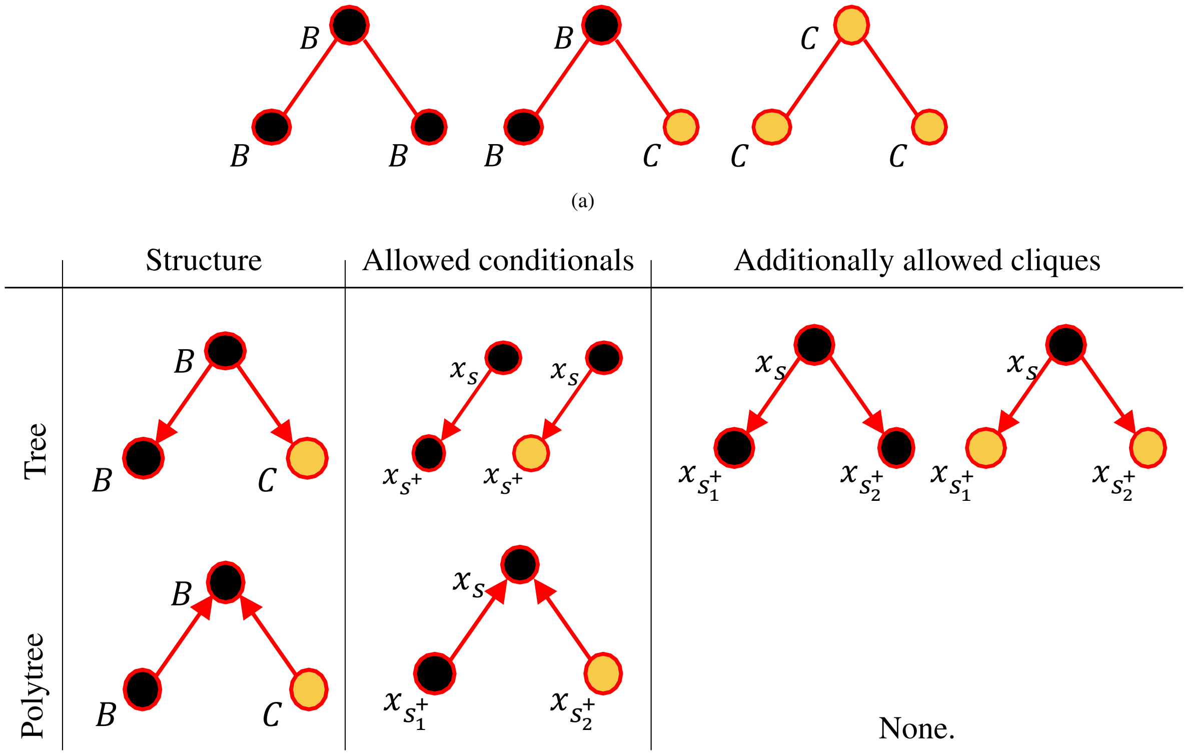
Factorization of the joint probability of a clique in directed trees (Fig. 4.3a);

$$p(\mathbb{X}) = p(x_{s_1^+}|x_s)p(x_{s_2^+}|x_s)p(x_s), \tag{4.1}$$

involves across-level constraints $p(x_{s_1^+}|x_s)$ and $p(x_{s_2^+}|x_s)$. This implies that the labels $x_{s_1^+}$ and $x_{s_2^+}$ are independent, given x_s. However, this conditional independence does not always hold (e.g. when the descendants are nodes corresponding to neighboring superpixels) and is a source of error in tree-based segmentation methods [209]. In polytrees, however, the joint probability of the clique (Fig. 4.3b);

$$p(\mathbb{X}) = p(x_s|x_{s_1^+}, x_{s_2^+})p(x_{s_1^+})p(x_{s_2^+}), \tag{4.2}$$

comprises of three-wise constraints $p(x_s|x_{s_1^+}, x_{s_2^+})$, that capture both across-level and same-



(a)

(b)

Figure 4.4: A two-class example with the background (B) and cell (C) labels for each node, for segmenting cells from the background. The feasible label configurations for three-wise cliques are shown in (a). As shown in (b), setting constraints to allow $B - B - C$ cliques in trees leads to allowing the unwanted clique $C - B - C$, while this does not occur in polytrees.

level dependencies. This feature enables polytrees to eliminate infeasible label configurations on the graph more precisely, compared to trees.

Figure 4.4 shows an example with two classes of background and cell, modeled by trees and polytrees. The three feasible configurations in this problem are shown in Fig. 4.4a. According to Eq. 4.1, allowing the *background* – *background* – *cell* clique on the tree requires setting both $p(x_{s^+} = background|x_s = background)$ and $p(x_{s^+} = cell|x_s = background)$ conditional probabilities to nonzero values (Fig. 4.4b). Doing so will also allow the *background* – *background* – *background* and *cell* – *background* – *cell* cliques, the latter of which is not feasible (see Fig. 4.4a). Based on Eq. 4.2, allowing *background* – *background* – *cell* for cliques on the *polytree* requires setting $p(x_s = background|x_{s_1^+} =$

background, $x_{s_2^+} = cell$) to nonzero values (Fig 4.4b). However, despite trees, allowing these label configurations on the polytree does not lead to the emergence of semantically infeasible cliques.

To incorporate priors in polytree segmentation, one needs to set the conditional probabilities $p(x_s|x_{s_1^+}, x_{s_2^+})$, based on the known label interrelations. Given the hierarchical structure of the graph with an upward fine-to-coarse resolution, each node s corresponds to the union of areas corresponding to s_1^+ and s_2^+. Therefore, setting $p(x_s|x_{s_1^+}, x_{s_2^+})$ is equivalent to determining the inclusion rules that work as a prior during segmentation.

4.3.3 Inference

We now explain the calculation of posteriors on the polytree. Given the observations $\mathbb{Y}$, finding the best segmentation is equivalent to associating the most probable label to each node (Bayesian inference)

$$\forall s \in \mathcal{G}, \hat{x}_s = \arg\max_{x_s \in \Lambda} p(x_s|\mathbb{Y}). \tag{4.3}$$

Here, the joint posterior $p(x_s|\mathbb{Y})$ is computed by marginalizing the probability of the clique $s_1^+ - s - s_2^+$, given $\mathbb{Y}$.

$$p(x_s|\mathbb{Y}) = \sum_{x_{s_1^+}, x_{s_2^+}} p(x_s, x_{s_1^+}, x_{s_2^+}|\mathbb{Y}). \tag{4.4}$$

This way of marginalizing the node's posterior reveals the contribution of three-wise constraints on cliques. Using the D-separation rule [140], the joint posterior is expanded as follows.

$$\begin{aligned} p(x_s, x_{s_1^+}, x_{s_2^+}|\mathbb{Y}) &= p(x_s|x_{s_1^+}, x_{s_2^+}, \mathbb{Y})p(x_{s_1^+}, x_{s_2^+}|\mathbb{Y}) \\ &= p(x_s|x_{s_1^+}, x_{s_2^+}, \mathcal{Y}_{a(s)})p(x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s_1^+, s_2^+)}, \mathcal{Y}_{d(s_1^+, s_2^+)}). \end{aligned} \tag{4.5}$$

Note that $\mathcal{Y}_{a(.)}$ and $\mathcal{Y}_{d(.)}$ refer to sets of ascendant and descendant observation nodes, respectively. For each node s (or a set of nodes $\mathcal{S}$), ascendant nodes are connected to s ($\mathcal{S}$) through edges with outward directions (nodes $\boldsymbol{y}_s$, x_{s^-}, $\boldsymbol{y}_{s^-}$, $x_{s'}$ and $\boldsymbol{y}_{s'}$ in Fig. 4.3d). Similarly, descendant nodes include the nodes connected to s ($\mathcal{S}$) through inward edge directions (nodes $x_{s_1^+}$, $\boldsymbol{y}_{s_1^+}$, $x_{s_2^+}$ and $\boldsymbol{y}_{s_2^+}$ in Fig. 4.3d).

We first elaborate the factor $p(x_s|x_{s_1^+}, x_{s_2^+}, \mathcal{Y}_{a(s)})$ on the right side of Eq. 4.5. This factor enforces posteriors of infeasible configurations to zero, as it is a product of the joint probability of a child node and its two parent nodes.

$$p(x_s|x_{s_1^+}, x_{s_2^+}, \mathcal{Y}_{a(s)}) = \frac{p(x_s, x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s)})}{\sum_{x'_s} p(x'_s, x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s)})}, \tag{4.6}$$

where the numerator can be expanded as follows.

$$\begin{aligned} p(y_s, y_{s_1^+}, y_{s_2^+}|\mathcal{Y}_{a(s)}) &= p(x_{s_1^+}, x_{s_2^+}|x_s)p(x_s|\mathcal{Y}_{a(s)}) \\ &= \frac{p(x_s, x_{s_1^+}, x_{s_2^+})}{p(x_s)} p(x_s|\mathcal{Y}_{a(s)}). \end{aligned} \tag{4.7}$$

The factor $p(x_s, x_{s_1^+}, x_{s_2^+})$ in Eq. 4.7 controls the occurrence of feasible and infeasible configurations on the graph, by setting zero and nonzero probabilities, respectively. Factor $p(x_s|\mathcal{Y}_{a(s)})$ in Eq. 4.7 is the posterior of node s having observed all its ascendant nodes and its own observation. This *top-down posterior* is expanded as

$$p(x_s|\mathcal{Y}_{a(s)}) \propto \sum_{x_{s^-}, x_{s'}} p(\boldsymbol{y}_s|x_s)p(\boldsymbol{y}_{s'}|x_{s'})p(x_{s'}|\mathcal{Y}_{d(s')})\frac{p(x_s, x_{s'}, x_{s^-})}{p(x_{s^-})p(x_{s'})}p(x_{s^-}|\mathcal{Y}_{a(s^-)}). \tag{4.8}$$

Equation 4.8 indicates that having calculated the likelihoods $p(\boldsymbol{y}_s|x_s)$ and $p(\boldsymbol{y}_{s'}|x_{s'})$ and the posterior $p(x_{s'}|\mathcal{Y}_{d(s')})$, the top-down posterior of node s is calculated based on the top-down posterior of node s^-. This implies that a top-down recursion calculates the top-down posteriors for all nodes.

The factor $p(x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s_1^+,s_2^+)}, \mathcal{Y}_{d(s_1^+,s_2^+)})$ on the right-hand side of Eq. 4.5 is factorized by several usages of the D-separation rule. This factorization separates parts calculated from the ascendant and descendant nodes.

$$\begin{aligned} p(x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s_1^+,s_2^+)}, \mathcal{Y}_{d(s_1^+,s_2^+)}) &\propto p(\mathcal{Y}_{a(s_1^+,s_2^+)}, \mathcal{Y}_{d(s_1^+,s_2^+)}|x_{s_1^+}, x_{s_2^+})p(x_{s_1^+}, x_{s_2^+}) \\ &= p(\mathcal{Y}_{a(s_1^+,s_2^+)}|x_{s_1^+}, x_{s_2^+})p(\mathcal{Y}_{d(s_1^+,s_2^+)}|x_{s_1^+}, x_{s_2^+})p(x_{s_1^+}, x_{s_2^+}) \\ &= p(\mathcal{Y}_{a(s_1^+,s_2^+)}|x_{s_1^+}, x_{s_2^+})p(\mathcal{Y}_{d(s_1^+)}|x_{s_1^+})p(\mathcal{Y}_{d(s_2^+)}|x_{s_2^+})p(x_{s_1^+}, x_{s_2^+}) \\ &\propto p(x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s_1^+,s_2^+)})\frac{p(x_{s_1^+}|\mathcal{Y}_{d(s_1^+)})}{p(x_{s_1^+})}\frac{p(x_{s_2^+}|\mathcal{Y}_{d(s_2^+)})}{p(x_{s_2^+})}. \end{aligned} \tag{4.9}$$

Similar to Eq. 4.8, factor $p(x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s_1^+,s_2^+)})$ on the right-hand side of Eq. 4.9 is calculated through a top-down recursion

$$p(x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s_1^+,s_2^+)}) \propto \sum_{x_s} p(\boldsymbol{y}_{s_1^+}|x_{s_1^+})p(\boldsymbol{y}_{s_2^+}|x_{s_2^+})p(x_{s_1^+}, x_{s_2^+}|x_s)p(x_s|\mathcal{Y}_{a(s)}). \tag{4.10}$$

Using a definition similar to that of the top-down posterior, the factors $p(x_{s_1^+}|\mathcal{Y}_{d(s_1^+)})$ and $p(x_{s_2^+}|\mathcal{Y}_{d(s_2^+)})$ in Eq. 4.9 are called *bottom-up* posteriors as they are calculated based on posteriors of their descendant nodes. For each node s in the graph, the bottom-up posterior is written as

$$p(x_s|\mathcal{Y}_{d(s)}) \propto \sum_{x_{s_1^+},x_{s_2^+}} p(\boldsymbol{y}_{s_1^+}|x_{s_1^+})p(\boldsymbol{y}_{s_2^+}|x_{s_2^+})p(x_{s_1^+}|\mathcal{Y}_{d(s_1^+)})p(x_{s_2^+}|\mathcal{Y}_{d(s_2^+)})p(x_s|x_{s_1^+}, x_{s_2^+}). \tag{4.11}$$

Derivations of Eq. 4.8, 4.10 and 4.11 are not included due to the shortage of space.

Using Eq. 4.5, 4.6, 4.7 and 4.9 the node's posterior in Eq. 4.4 given all the observations is written as follows.

$$p(x_s|\mathbb{Y}) \propto \sum_{x_{s_1^+},x_{s_2^+}} \frac{p(x_s, x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s)})}{\sum_{x_s'} p(x_s', x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s)})} p(x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s_1^+,s_2^+)}) \frac{p(x_{s_1^+}|\mathcal{Y}_{d(s_1^+)})}{p(x_{s_1^+})} \frac{p(x_{s_2^+}|\mathcal{Y}_{d(s_2^+)})}{p(x_{s_2^+})}. \tag{4.12}$$

Two types of terms (messages) emerge on the right-hand side of Eq. 4.12 in the factorization of the joint posterior for the node s: 1) *top-down messages* $p(.|\mathcal{Y}_{a(.)})$: posterior marginals that depend merely on ascendant observation nodes $\mathcal{Y}_{a(s)}$, and 2) *bottom-up messages* $p(.|\mathcal{Y}_{d(.)})$: posterior marginals that are merely dependent on descendant observation nodes $\mathcal{Y}_{d(s)}$. We derived recursive procedures that calculate the top-down messages from child nodes for each node, creating a top-down inference pass from the leaf to the root nodes. Similarly, bottom-up messages are calculated from parent nodes of each node creating a bottom-up pass from roots to the leaf. Sweeping the two passes, top-down and bottom-up messages are calculated for each node, based on which the posterior of that node (Eq. 1) is computed for a given set of parameters.

4.3.4 Parameter Estimation

We use an Expectation-Maximization (EM) algorithm for parameter estimation. The EM algorithm [148] is an extension of the maximum likelihood approach dealing with incomplete data. EM iteratively increases the likelihood of the observed data based on the posterior probabilities. Since the parameter values can critically influence the results, automatic methods for parameter estimation are highly desirable to adapt to the characteristics of each microscopy set.

Let $\mathbf{\Theta}$ be the set of parameters involved in the joint distribution $p(\mathbb{Y}, \mathbb{X}|\mathbf{\Theta})$. Given the set of possible node labels Λ and considering the dependencies on the hierarchy, the joint probability distribution of the model is expressed as

$$p(\mathbb{Y}, \mathbb{X}|\mathbf{\Theta}) = \prod_{s \in \mathcal{R}} p(x_s|\mathbf{\Theta}) \prod_{s \notin \mathcal{R}, s \in \mathcal{G}} p(x_s|x_{s_1^+}, x_{s_2^+}, \mathbf{\Theta}) \prod_{s \in \mathcal{G}} p(\boldsymbol{y}_s|x_s, \mathbf{\Theta}), \tag{4.13}$$

where $\mathcal{R}$ is the set of root nodes in graph $\mathcal{G}$. Given the current estimate of model parameters $\mathbf{\Theta}^{(n)}$ at the n^{th} step, expectation of the joint log-likelihood function is calculated (E-step)

$$\begin{aligned} Q(\mathbf{\Theta}|\mathbf{\Theta}^{(n)}) = & \sum_{s \in \mathcal{R}} \sum_{i \in \Lambda} p(x_s = i|\mathbb{Y}, \mathbf{\Theta}^{(n)}) \ln p(x_s = i|\mathbf{\Theta}) \\ & + \sum_{s \notin \mathcal{R}} \sum_{(i,j,k) \in \mathbb{F}} p(x_s = i, x_{s_1^+} = j, x_{s_2^+} = k|\mathbb{Y}, \mathbf{\Theta}^{(n)}) \ln p(x_s = i|x_{s_1^+} = j, x_{s_2^+} = k, \mathbf{\Theta}) \\ & + \sum_{s \in \mathcal{G}} \sum_{i \in \Lambda} p(x_s = i|\mathbb{Y}, \mathbf{\Theta}^{(n)}) \ln p(\boldsymbol{y}_s = \boldsymbol{l}|x_s = i, \mathbf{\Theta}), \end{aligned} \tag{4.14}$$

where $\mathbb{F} \subset \Lambda^3$ is the set of feasible configurations and $\boldsymbol{l}$ is the observed feature vector for node s from the set of all possible data vectors $\boldsymbol{L}$. The parameter vector $\mathbf{\Theta}$ to be estimated is

$$\mathbf{\Theta} = (p(x_s = i)_{s \in \mathcal{R}}, p(x_s = i|x_{s_1^+} = j, x_{s_2^+} = k)_{s \notin \mathcal{R}}, p(\boldsymbol{y}_s = \boldsymbol{l}|x_s = i)_{s \in \mathcal{G}})^T, \tag{4.15}$$

which is subject to these constraints

$$\begin{aligned} & \sum_{i \in \Lambda} p(x_s = i) = 1, \\ & \sum_{i \in \Lambda} p(x_s = i|x_{s_1^+} = j, x_{s_2^+} = k) = 1, \\ & \sum_{\boldsymbol{l} \in \boldsymbol{L}} p(\boldsymbol{y}_s = \boldsymbol{l}|x_s = i) = 1. \end{aligned} \tag{4.16}$$

Using Lagrange multipliers for maximizing these parameters with the constraints, the following update equations are derived for the M step.

$$\begin{aligned} p(x_s = i|\boldsymbol{\Theta}^{(\boldsymbol{n}+1)}) &= \frac{1}{|\mathcal{R}|}\sum_{s\in\mathcal{R}}\sum_{(j,k)\in\Lambda^2} p(x_s = i, x_{s_1^+} = j, x_{s_2^+} = k|\mathbb{Y}, \boldsymbol{\Theta}^{(\boldsymbol{n})}), \\ p(x_s = i|x_{s_1^+} = j, x_{s_2^+} = k, \boldsymbol{\Theta}^{(\boldsymbol{n}+1)}) &= \frac{\sum_{s\notin\mathcal{R}} p(x_s = i, x_{s_1^+} = j, x_{s_2^+} = k|\mathbb{Y}, \boldsymbol{\Theta}^{(\boldsymbol{n})})}{\sum_{s\notin\mathcal{R}}\sum_{i\in\Lambda} p(x_s = i, x_{s_1^+} = j, x_{s_2^+} = k|\mathbb{Y}, \boldsymbol{\Theta}^{(\boldsymbol{n})})}. \end{aligned} \tag{4.17}$$

Assuming Gaussian distributions for class-conditional likelihood functions, $p(\boldsymbol{y}_s = \boldsymbol{l}|x_s = i, \boldsymbol{\Theta}) = \mathcal{N}(\boldsymbol{y}_s = \boldsymbol{l}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, the mean and covariance of each class i are updated by

$$\boldsymbol{\mu}_i^{(n+1)} = \frac{\sum_{s\in\mathcal{G}} p(x_s = i|\mathbb{Y}, \boldsymbol{\Theta})\boldsymbol{y}_s}{\sum_{s\in\mathcal{G}} p(x_s = i|\mathbb{Y}, \boldsymbol{\Theta})}, \tag{4.18}$$

$$\boldsymbol{\Sigma}_i^{(n+1)} = \frac{\sum_{s\in\mathcal{G}} p(x_s = i|\mathbb{Y}, \boldsymbol{\Theta})[\boldsymbol{y}_s - \boldsymbol{\mu}_i^{(n+1)}][\boldsymbol{y}_s - \boldsymbol{\mu}_i^{(n+1)}]^T}{\sum_{s\in\mathcal{G}} p(x_s = i|\mathbb{Y}, \boldsymbol{\Theta})}. \tag{4.19}$$

Posterior probabilities $p(x_s = i|\mathbb{Y}, \boldsymbol{\Theta})$ that appear in the two update equations 4.18 and 4.19 are computed using Eq. 4.5. The algorithm iterates between E and M steps until a convergence is reached (usually within 10 EM iterations). To reconstruct the segmented image, labels of nodes are set as classes for which the posterior probability of that node is the maximum. See Algorithm 2 for an overview of the segmentation steps.

4.4 Experiments and Results

In this section, we first assess the sensitivity of the algorithm to the training size and visualize the evolution of the model parameters towards convergence. This is done through solving the inference problem for randomly generated graphs. We also evaluate the proposed EM-polytree algorithm on image segmentation and interpretation, and prediction of segmentation error. Four image datasets were used that contain cell microscopy images; 1) the synthetic dataset with 25 simulated images of HL60 cell nuclei and sub-nuclear particles, namely nucleoli, generated by the Mitogen software [26], 2) the zebrafish dataset having 10 random two-channel microscopy images of host and pathogen cells, viz. macrophages and the fungal pathogen *Cryptococcus neoformans* published in [1], 3) the BBBC020 set of 20

Algorithm 2 Segmentation using EM-polytrees

```
procedure Graph generation.
▷ Oversegment the image using SEEDS [196]
▷ Extract scale-space differential invariants as features
▷ Insert a node in the graph corresponding to each superpixel
while $\#(orphan\ nodes) > 1$ do
    Create a new parent node n
    Connect the two most similar orphan nodes to n
end while
procedure Initialization.
▷ Set the constraints $p(x_s | x_{s_1^+}, x_{s_2^+})$ based on the problem
▷ Set labels of nodes in the sparse training set $\mathcal{T}$
▷ Calculate initial likelihood parameter values
for all $i \in \Lambda$ do
    $$\boldsymbol{\mu}_i^{(0)} = \frac{\sum_{s \in \mathcal{T}} p(x_s = i) \boldsymbol{y}_s}{\sum_{s \in \mathcal{T}} p(x_s = i)}, \quad \boldsymbol{\Sigma}_i^{(0)} = \frac{\sum_{s \in \mathcal{T}} p(x_s = i) [\boldsymbol{y}_s - \boldsymbol{\mu}_i^{(0)}][\boldsymbol{y}_s - \boldsymbol{\mu}_i^{(0)}]^T}{\sum_{s \in \mathcal{T}} p(x_s = i)}.$$
end for
procedure Expectation-Maximization.
▷ Oversegment the image using SEEDS [196]
▷ Extract scale-space differential invariants as features
▷ Insert a node in the graph corresponding to each superpixel
while Convergence is not reached do
    E-Step
    for all $s \in \mathcal{G}$ do
        Calculate the posterior for node s through the two-pass inference
        $$p(x_s | \mathbb{Y}) \propto \sum_{x_{s_1^+}, x_{s_2^+}} \frac{p(x_s, x_{s_1^+}, x_{s_2^+} | \mathcal{Y}_{a(s)})}{\sum_{x'_s} p(x'_s, x_{s_1^+}, x_{s_2^+} | \mathcal{Y}_{a(s)})} p(x_{s_1^+}, x_{s_2^+} | \mathcal{Y}_{a(s_1^+, s_2^+)}) \frac{p(x_{s_1^+} | \mathcal{Y}_{d(s_1^+)})}{p(x_{s_1^+})} \frac{p(x_{s_2^+} | \mathcal{Y}_{d(s_2^+)})}{p(x_{s_2^+})}.$$
    end for
    M-Step
    for all $i \in \Lambda$ do
        Calculate parameter values for class i based on the current posteriors
        $$\boldsymbol{\mu}_i^{(n+1)} = \frac{\sum_{s \in \mathcal{G}} p(x_s = i | \mathbb{Y}, \boldsymbol{\Theta}) \boldsymbol{y}_s}{\sum_{s \in \mathcal{G}} p(x_s = i | \mathbb{Y}, \boldsymbol{\Theta})}, \quad \boldsymbol{\Sigma}_i^{(n+1)} = \frac{\sum_{s \in \mathcal{G}} p(x_s = i | \mathbb{Y}, \boldsymbol{\Theta}) [\boldsymbol{y}_s - \boldsymbol{\mu}_i^{(n+1)}][\boldsymbol{y}_s - \boldsymbol{\mu}_i^{(n+1)}]^T}{\sum_{s \in \mathcal{G}} p(x_s = i | \mathbb{Y}, \boldsymbol{\Theta})}.$$
    end for
    Create a new parent node n
    Connect the two most similar orphan nodes to n
end while
procedure Segmented image reconstruction.
▷ Generate a blank image with the same size as the input image
for all $s \in \mathcal{G}$ do
    $label_s = \arg\max_i p(x_s = i | \mathbb{Y})$
    Color the image based on the pixel list and label of s
end for
```

two-channel microscopy images of murine bone marrow macrophages and their nuclei from the Broad Bioimage Benchmark Collection [25], and 4) the ISBI2015 Challenge dataset with 900 images of cytoplasms and nuclei of cervical cells obtained from Pap smear [27]. Segmentation of the images in the four datasets involves multi-class segmentation problems with constraints on structural embedding. These constraints, e.g. nuclei being within the cytoplasms and pathogens being engulfed by immune cells, can be enforced by setting the conditional probabilities, making up clique posteriors in Eq. 4.4.

We compare the performance of EM-polytree to those of EM-tree, EM-GMM, and SegNet using Dice Similarity Coefficients (DSC) [199]. To measure the segmentation performance, DSC calculates the ratio between the intersection and the union of the automatic and ground truth segmentations. For an image I, with the ground truth segmentation I_{GT} and a segmentation I_S, DSC is calculated as

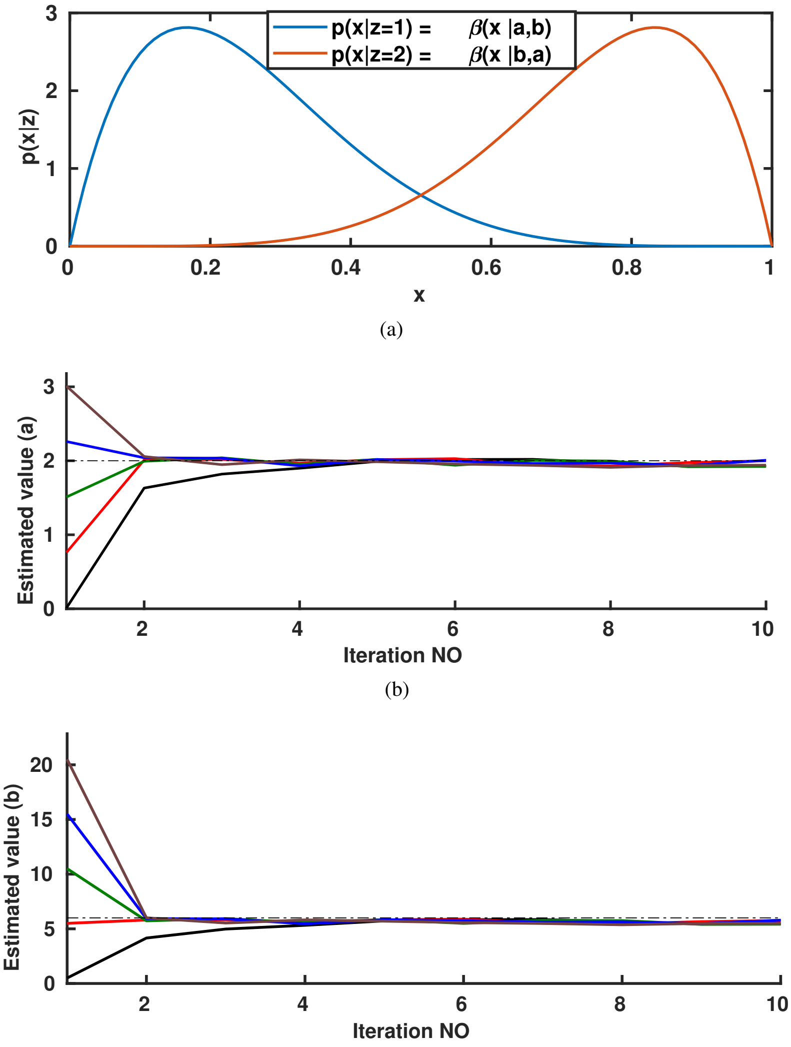
$$DSC = \frac{2|I_{GT} \cap I_S|}{|I_{GT}| + |I_S|}, \tag{4.20}$$

where $|.|$ is the number of nonzero pixels in the image and $I_{GT} \cap I_S$ is the intersection of the nonzero areas in the two images. Given Eq. 4.20, higher values of DSC correspond to more accurate segmentations.

4.4.1 Sensitivity and convergence analysis

The sensitivity of the proposed algorithm to the initial parameter values and to the proportion of the training nodes was evaluated using randomly generated graphs. We considered the classification of synthetic data generated by the ancestral sampling technique [66]. We draw samples for x_s variables to represent ground truth data. Based on this, the $\boldsymbol{y}_s$ variables are then drawn according to the presumed class-conditional distributions. Next, retaining the labels x_s for a random selection of 10% of the graph root nodes (corresponding to 5% of the whole nodes), model parameters are estimated (M step) and new values are inferred for x_s from the observed $\boldsymbol{y}_s$ variables (E step). Iterating between E and M continues until a convergence is reached. We compare the inferred x_s variables to the ground truth and experimentally validate the viability of the algorithm.

For a graph with N nodes, to draw samples $\hat{x}_1, \hat{x}_2, ..., \hat{x}_N$ from the joint distribution $p(\mathbb{Y}, \mathbb{X})$,



(a)

(b)

(c)

Figure 4.5: The likelihood distributions of the two classes (a) and the evolution of *a* and *b* parameters in beta likelihood functions ((b) and (c), respectively). The actual parameter values are shown by dashed lines in (b) and (c). 10 EM iterations are shown, starting from different sets of initial values (shown by different colors). The inference and parameter optimization were done on random graphs generated with ten thousand root nodes.

we first sample from the probability distribution $p(x_s)\big|_{s\in\mathcal{R}}$ for all root nodes. Visiting each internal node in an upward recursion, we sample from the conditional distribution $p(x_s|x_{s_1^+}, x_{s_2^+})$,

Table 4.1: Percentages of errors for different proportions of training nodes in random graphs with 1000 root nodes.

Training size (%)	10	20	30	40	50	60	70	80	90	100
Error (%)	3.17	2.66	2.38	1.98	1.87	1.46	1.06	0.68	0.28	0

where the parent labels $\hat{x}_{s_1^+}$ and $\hat{x}_{s_2^+}$ have been sampled in previous steps. Once we have sampled from the leaf node of the graph, $\hat{x}_N$, we will have obtained a sample from the joint distribution $p(\mathbb{Y}, \mathbb{X})$.

In this section only, we considered two classes for x_s for simplicity and selected y_s from the continuous range of $[0, 1]$. class-conditional likelihood functions, $p(y_s|x_s)$ were beta distributions shown in Fig. 4.5a. For different initial values of the likelihood parameters, the convergence of the algorithm to the optimal values is in Fig. 4.5b and 4.5c. These figures show that the algorithm converged after about 4 EM iterations, with steep leaps toward the optimal values in the early steps.

We have also studied the sensitivity of the method to the size of the training nodes. Table 4.1 shows the accuracy of the inferred labels when 10% to 100% of the root nodes were used as training. The accuracy of the segmentation improves with the size of the training set. However, we obtain a relatively small error rate (i.e. 3.17%) even for fairly sparse annotations (i.e. as small as 5% of the nodes).

4.4.2 Multi-class segmentation of synthetic/real microscopy data

Figure 4.2 shows the block diagram of EM-based inference using the three weakly supervised methods, i.e. EM-polytree, EM-tree, and EM-GMM. Scale-space differential invariants [198] were used as features, with their scales being selected based on the average size of the objects in each dataset. Multi-dimensional Gaussian distributions were used as class-conditional likelihood functions with diagonal covariance matrices to reduce the number of parameters. SEEDS [196] algorithm was used for generating the superpixels from the input images and the graph generation was done as explained in section 4.3.1. To label the training nodes in the generated graph, 5% of the nodes were randomly selected and their labels were set based on the annotated image. Starting from an initial set of parameter values based on

Table 4.2: Means and standard deviations of Dice similarity coefficients in segmentation.

Dataset	Synthetic		Zebrafish	
class	nucleus	nucleolus	macrophage	pathogen
SegNet	0.80 ± 0.11	0.05 ± 0.02	0.27 ± 0.12	0.01 ± 0.00
EM-GMM	0.70 ± 0.19	0.20 ± 0.15	0.70 ± 0.04	0.22 ± 0.38
EM-tree	0.70 ± 0.19	0.17 ± 0.14	0.68 ± 0.05	0.30 ± 0.21
EM-polytree	**0.81 ± 0.13**	**0.23 ± 0.15**	**0.77 ± 0.06**	**0.38 ± 0.25**

Dataset	BBBC020		ISBI2015	
class	cell	nucleus	cytoplasm	nucleus
SegNet	0.72 ± 0.07	**0.81 ± 0.03**	**0.90 ± 0.02**	0.63 ± 0.08
EM-GMM	0.61 ± 0.18	0.78 ± 0.07	0.63 ± 0.12	0.84 ± 0.17
EM-tree	0.78 ± 0.08	0.79 ± 0.06	0.80 ± 0.09	0.59 ± 0.23
EM-polytree	**0.86 ± 0.06**	0.80 ± 0.04	0.88 ± 0.06	**0.86 ± 0.15**

the labeled nodes, the labels are inferred (E-step), and this is followed by a new estimation of the parameter values based on the inferred labels (M-step). The algorithm iterates between E and M steps, until convergence is reached (mostly within 10 iterations).

The four image datasets were used to evaluate the performance of the method in multi-class segmentation compared to EM-tree, EM-GMM, and SegNet [78]. Fig. 4.6 shows samples from each dataset and their segmentations. For inferring the labels on tree, we adapted Laferte *et al.* [167] formulation to the irregular trees generated in this work.

To apply SegNet as a supervised segmentation method, cross-validation was done on the datasets to provide SegNet with the training set. Data augmentation and customized wighting were used to maximize the performance of SegNet.

Table 4.2 shows the average DSC and their standard deviations for the four methods on the four datasets. We randomly selected 50 images from the ISBI2015 dataset for comparing the four methods. Table 4.3 compares EM-polytree segmentation on the 900 images of the ISBI2015 dataset with the first and second ranks of the challenge. The results show that the proposed method outperforms the competing methods in the overall DSC in multi-class segmentation.

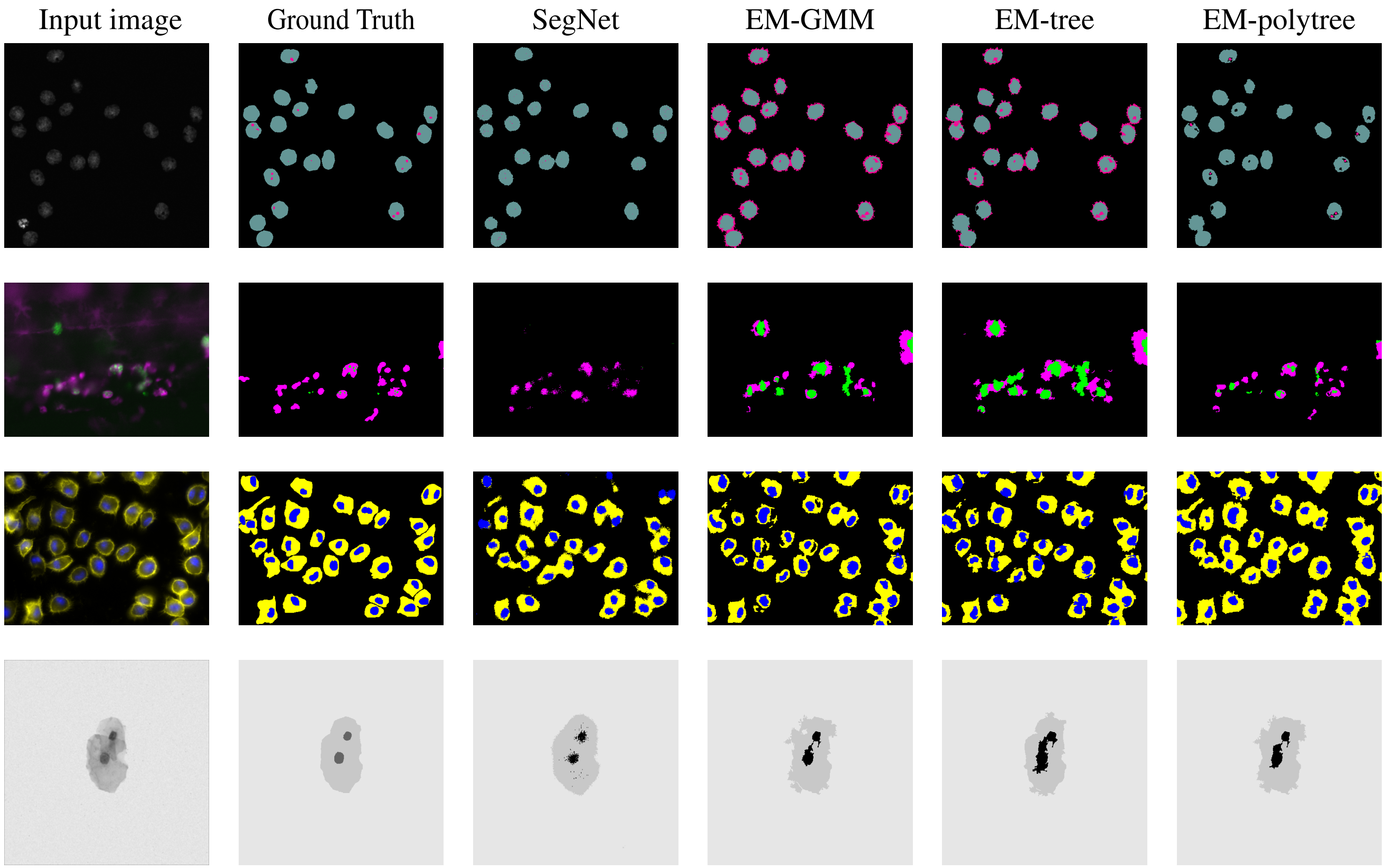


Figure 4.6: Evaluation on multi-class image segmentation. Samples from synthetic (light green nuclei and magenta nucleoli), zebrafish (magenta host and green pathogen cells), BBBC020 (yellow cells and blue nuclei) and ISBI2015 challenge (gray cytoplasms and black nuclei) datasets are shown in the first to fourth rows, respectively.

Table 4.3: Comparison on ISBI2015 cell segmentation challenge dataset.

Method	DSC	FNo	TPp	FPp
Phoulady *et al.*	0.831 ± 0.079	0.408 ± 0.163	0.927 ± 0.098	0.003 ± 0.002
Ushizima *et al.*	0.856 ± 0.078	0.501 ± 0.180	0.899 ± 0.113	0.002 ± 0.001
EM-polytree	0.902 ± 0.052	0.036 ± 0.078	0.918 ± 0.094	0.003 ± 0.001

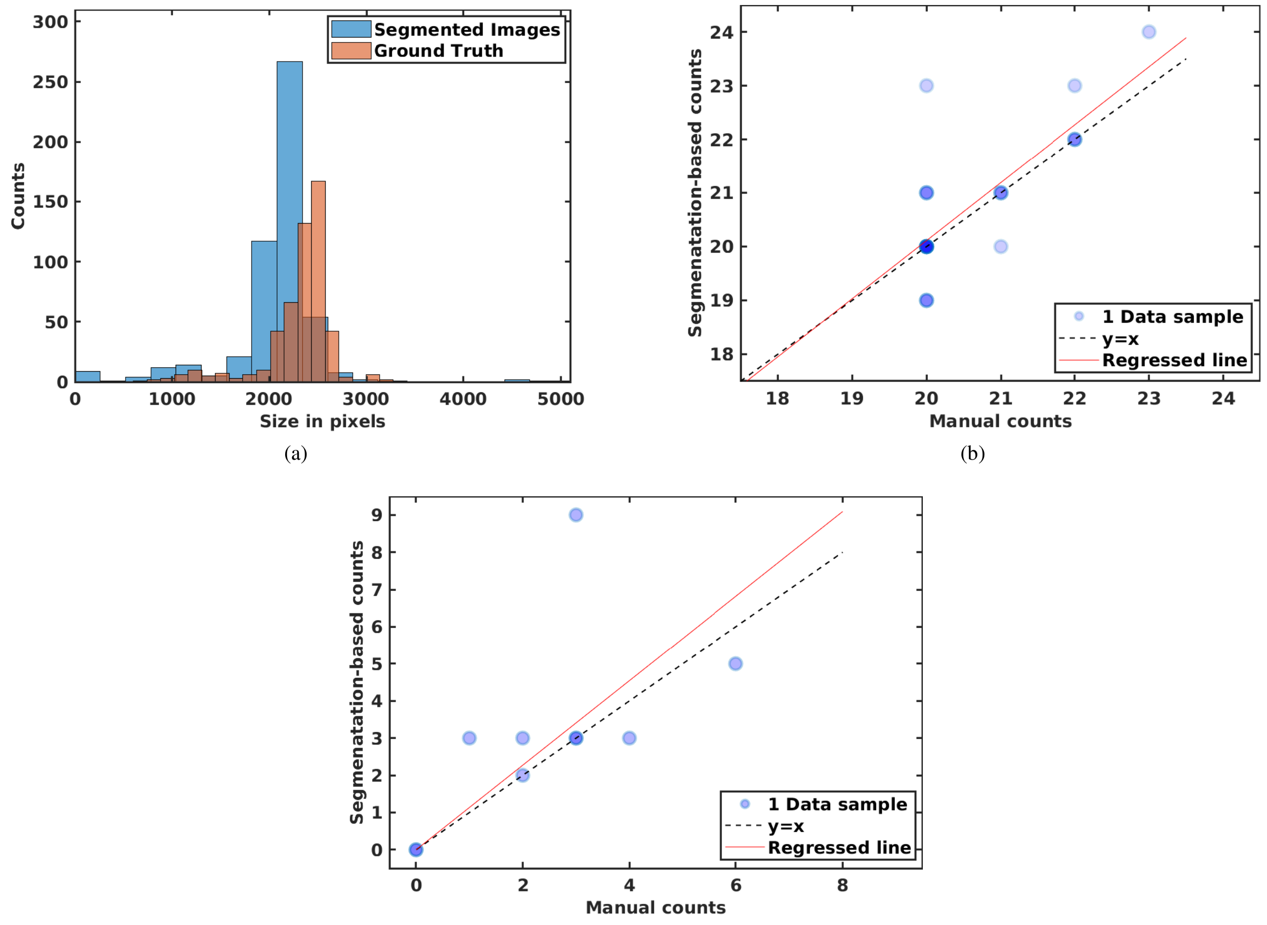
4.4.3 Segmentation-based assessment

To assess the reliability of EM-polytree segmentation, semantic information extracted from the images segmented by the proposed method were compared to the ground truth. For the synthetic dataset, numbers of nuclei and histograms of their sizes in the automatic and manual segmentation were drawn in Fig. 4.7a and 4.7b, respectively. For the zebrafish dataset, the number of infected macrophages, i.e. host cells with pathogens in them, were counted based on the two segmentations. The experiments show a significant correlation (p-value = 0) for the measures on the synthetic dataset and a correlation with p-value = 0.06 for the measure on the zebrafish dataset, which indicate the reliability of the EM-polytree for segmentation.

4.4.4 Segmentation error prediction

The proposed method can evaluate the compliance of the segmented image with the imposed priors. A strong non-compliance can indicate a faulty segmentation to be flagged up for subsequent correction. This was implemented by assessing the label configurations on the labeled graph that do not comply with the priors, i.e. $p(x_s|x_{s_1^+}, x_{s_2^+})$ in polytrees and $p(x_{s^+}|x_s)$ in trees (see Section 4.4.2). Areas of the segmented image that correspond to such label configurations were nominated as the predicted error and Dice similarity coefficients were calculated between the annotated areas and the actual segmentation error.

Figure 4.8a shows the predicted segmentation error by the red color for a sample image from BBBC020 dataset segmented by EM-polytree and EM-tree. The DSC between the predicted and actual segmentation error over the entire dataset in Fig. 4.8b shows that the EM-polytree outperforms the EM-tree in predicting the error.

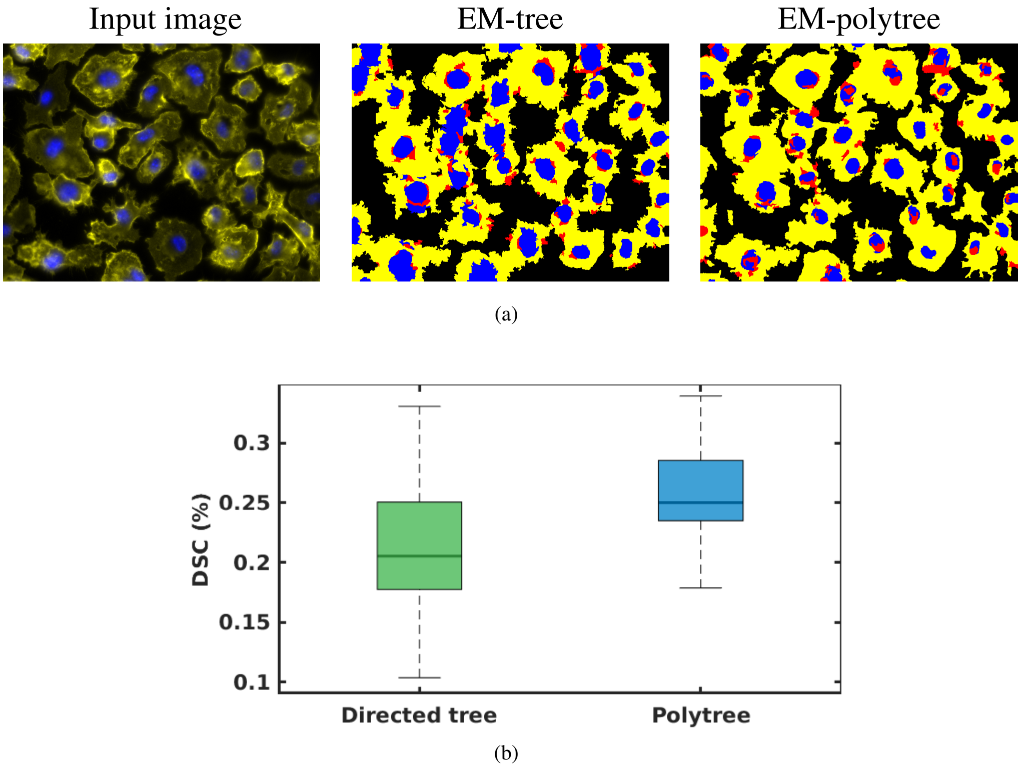


(a)

(b)

(c)

Figure 4.7: Comparing semantic interpretations based on the EM-polytree segmented images with the ground truth values. Panel (a) shows the size histograms for the nuclei in the manual and automatic EM-polytree segmentations of the synthetic image dataset. Panel (b) shows nuclei counts using the EM-polytree and manual segmentations of the synthetic image dataset. Panel (c) shows the counts of the infected host cells (macrophages) based on the manual and EM-polytree segmentations. Performing pair-wise correlation tests shows significant correlations with Pearson correlation coefficients of 0.9877 and 0.7584 (p-value = 0) in (b) and (c), respectively, and a Pearson correlation coefficient of 0.6050 (p-value = 0.0639) in (c).



(a)

(b)

Figure 4.8: Comparison of the error prediction in EM-tree and EM-polytree methods. Panel (a) shows the errors annotated by red on a sample image from BBBC020 dataset. Panel (b) shows the Dice similarity coefficients between the predicted and actual segmentation errors for the two methods (p-value of the pairwise t-test was 0.005).

4.5 Discussion

Here, we addressed the problem of a constrained segmentation of biological microscopy data, when limited training data is available. We have proposed EM-based polytrees, in which the priors are encoded in the conditional probabilities governing the label configurations on three-wise cliques. Using the EM algorithm, parameters of the model have been initially set and then optimized iteratively to generate the final segmentation. Label inference has been done efficiently through a two-pass algorithm that calculates closed-form posteriors on the polytree.

To investigate the effect of a comprehensive prior knowledge implementation in the inference, we compared EM-polytree with EM-GMM (having no constraints), EM-tree (with weaker constraints compared to polytrees) and SegNet (with implicitly learned constraints). Evaluating the methods on multi-class image segmentation shows that EM-polytree and SegNet produce the most accurate segmentations among others (Table 4.2). Even though SegNet performs better in two cases, nuclei in the BBBC020 dataset and cytoplasms in the ISBI2015 dataset, the overall DSC of the polytree approach is significantly better than all other methods.

To evaluate the segmentation results, we calculated three quantitative measures (section 4.4.3). The results indicate that two of these measures, acquired using EM-polytree, significantly correlate with the values for the ground truth.

Inspired by the interactive segmentation framework by Uzunbas *et al.* [129], we delineated areas of the segmented image that did not comply with the imposed constraints. This can provide an interactive segmentation method where the user is asked to correct the error-prone areas flagged up by the algorithm. This is particularly useful in applications where the existing ground truth is limited and/or unreliable, rendering the training of CNNs implausible. Alternatively, the optimization can start from an arbitrary set of parameters and iterate only relying on the interactions from the user to converge.

The sensitivity of an algorithm to the training set is one of its key features, especially when dealing with sparsely labeled data. We evaluated EM-polytrees by inference on arbitrary graphs. The experiments implicate the low dependency of the output on the proximity of the initial values to the final ones and the size of the training set. This emphasizes the possible future application of the algorithm in an unsupervised framework, where no label is available.

Limitations and future work

The present method generates the graph structure based on the features of the superpixels attached to the graph nodes. However, the inferred label information can also be useful in determining the graph structure. To infer the structure of the graph simultaneously with nodes' labels, one direction of future work could be to use dynamic polytrees, as dynamic trees [178]. Another space for future improvement resides in the generation of superpixels. We have used SEEDS algorithm due to its efficiency and independence from the initial grid. However, depending on the application, one can choose other superpixel generation algorithms with possible improvements in the segmentation results.

One possible future application of the proposed method is the evaluation of the priors considered for segmentation. This can be done by comparing the values of the initial conditional probabilities with their optimal values. On the other hand, as the EM algorithm can be trapped in local minima [210], the proposed algorithm cannot be readily applied to the unsupervised analysis. This can be achievable where either a global optimization approach is employed, or an interactive version of the method is implemented.

The results of this work verify that polytrees can incorporate prior knowledge more effectively compared to trees. This finding opens new horizons for polytrees to replace trees in their applications for enforcing priors.

Chapter 5

Discussion and conclusions

5.1 Summary

Microscopy images play a critical role in biomedical research, disease diagnosis, and treatment. They can help to understand the healthy and pathological behaviors in order to develop effective treatments and drugs. To this end, methods and tools are required to quantify the images by calculating different metrics such as indicators of disease grades [1]. This need is emphasized by the large cohorts of data generated by advanced microscopes with increased spatial and temporal resolutions. Therefore, faster, more accurate and cheaper methods are desired that are also scalable to large data analysis and can compete with the human precision.

In this thesis, I addressed the segmentation of images containing objects of different types, which is a common problem in the quantification of microscopy images. I investigated mathematical modeling of semantic constraints employed by human analyzers through probabilistic graphical models to improve the segmentation accuracy. More specifically, I looked at modeling between-class relationships for improving the segmentation performance. Tree graphical models can capture between-class relationships but are limited in a sense that only across-level dependencies can be modeled. However, same-level dependencies are also important in modeling images by graphical models and ignoring them in tree models leads to blocky artifacts in segmentations [181]. Polytrees can capture both across-level and same-level dependencies which can lead to a more accurate matching of the segmented image with the prior knowledge. With the inference algorithm presented in this thesis, these models can be applied to image segmentation, addressing the limitations of tree graphical models. On the other hand, CNNs can provide high segmentation accuracies, but they usually require large training datasets, which is expensive and not always available. I investigated methods that can mainly address the two aspects of capturing inter-class relationships and having a low dependency on training.

5.1.1 Achievements

In addressing the problem mentioned above, this thesis has had multiple contributions that are summarized as follows.

After reviewing the state-of-the-art models in chapter 3, polytree graphical models were introduced to the problem of multi-class segmentation in chapter 4. A polytree graphical model was generated for each image for which a probabilistic model was made. Two types of features were extracted for the images, including the scale-space invariant and deep features. The features are ranked using the Fisher score and the most relevant features were used for segmentation. Finding the optimal labeling of the graphical model equals performing an optimal segmentation.

An exact method for finding the posterior probability distributions on the polytree was presented in chapter 4. Using the Bayes law and factorizing posterior distribution functions using the d-separation rule, a message passing formulation has been derived that calculates the closed-form posteriors in two passes on the graph. Applying the proposed algorithm to the classification of synthetic data and the multi-class segmentation problem in chapter 4 shows the effectiveness of incorporation of inter-class relations using polytrees.

Using the Expectation-Maximization algorithm for optimizing the parameters of the polytree model, chapter 5 presented a weakly supervised segmentation approach that can operate with a lowered reliance on the training data. Assessing the sensitivity of the proposed method to the initial parameter values and the size of the training indicated the low dependency of the output on the proximity of these two elements. The method was also evaluated with the training available for 5% of the nodes on the graph and was applied to the multi-class segmentation problem to multiple microscopy datasets. Finally, the reliability of the segmented images was assessed by extracting quantitative measures from the image that were shown to significantly correlate with the ground truth values.

The ability of the proposed method in explicitly modeling the constraints allows assessing the segmented images in matching the enforced label configurations. Using this feature in chapters 4 and 5, I showed that polytree segmentation can predict possible locations of error in the segmented images. Comparing the results with tree graphical models shows that polytrees can more comprehensively predict the possible error in the segmented images, which is due to their superior ability to capture the interrelations.

5.2 Applications

The method developed in this project was motivated by the existing challenges in microscopy image analysis, mainly being the presence of objects of multiple types and the limited availability of the ground truth. Apart from the derived formulation for inference on polytree graphical models and the designed framework for employing multi-class segmentation, the implemented software can be made available for future image analysis applications. Even though the motivation and evaluation of the proposed method have been on microscopy images, it can be applied to other types of images where incorporating class dependencies can improve the overall performance of segmentation. For example, a new collaboration has been set up to apply polytree segmentation to the problem of segmenting femur CT images, where the dense bone resides in the bone with a lower density, and the whole bone is inside the muscle, defining a multi-class segmentation problem. Following validation steps, this work will be performed following this Ph.D.

Depending on the images the algorithm is being applied to, the proposed method might need to operate on 3D images. To change the input and output images from 2D to 3D, the first and last steps of the algorithm (as shown in Figure 4.2) need to be altered. More specifically, the oversegmentation step should be done by an algorithm that generates supervoxels instead of superpixels and the features have to be calculated in 3D. The graph generation algorithm and the inference do not require changes. The reconstruction of the segmented image should also be performed in 3D.

In this thesis, the proposed methods have been evaluated on stained transmitted light and fluorescence microscopy images. However, the same algorithm can be applied to other types of microscopy images, such as confocal, DIC or electron microscopy. Doing so might need changing the features based on the method. For example, for DIC images gradient-based features can better describe the object boundaries. Having that said, since the proposed method features an automatic feature selection step, applying it to different types of images with distinct characteristics requires expanding the pool of features that are initially calculated. The algorithm will then be able to find the most discriminant features for the problem using Fisher scores.

Identification of multiple objects and their nuclei is useful in analyzing 3D fluorescence breast cancer histopathological images. Most of the existing techniques, such as the Euclidean Distance Transform are effective in for the regular shapes of nuclei in these images and fail in the more complex shapes. Aiding the segmentation by employing the prior knowledge and performing a joint segmentation of cells and their nuclei can address this problem.

Performing a constrained segmentation can be beneficial in the analysis of touching or overlapping objects in noisy images. This is considered as the challenging part of segmenting images containing large numbers of cells, e.g. for the nucleus-like nano-particles in electron microscopy images. Given that nuclei have elliptical shapes, this information can be mathematically formulated and be used as a morphological constraint to be satisfied for the segmentation of nuclei in fluorescence microscopy and Pap smear images, and mitotic cells in H&E-stained multi-spectral microscopy images.

Another type of morphological constraints are the existence of local minima for nucleus/cell segmentation. Finding the locations of the local minima and considering it as the prior probability, the connected areas surrounding them could be assessed to located the desired objects. This technique facilitates the segmentation of pathological and biomedical microscopy images, such as overlapping cells in Pap smear images, IHC-stained breast cancer images and phased contrast microscopy images. Constraints on nucleus size can increase the accuracy of analyzing cells in endometrial histopathology images, like the uterine natural killer cell nuclei. A combination of local minima information and nucleus size information can be used in detecting stromal cell nuclei of cervical and breast cancer cells. These images have been previously analyzed using watershed segmentation techniques, without the imposing the prior knowledge in the segmentation [211].

The circular shape of the nuclei in 2D, or their spherical shapes in 3D, has been used in the literature as a clue for their detection and segmentation (e.g. [212]). This morphological information can be incorporated in the generation of superpixels in the first step of the method proposed in this thesis. This way the superpixels (or supervoxels) will be initially formed in a way to follow the boundaries of nuclei. Starting from this improved oversegmentation, the segmentation can be performed using the shapes of the nuclei as a prior. This approach can be useful in the segmentation of haematopoietic cells in confocal microscopy images of zebrafish [213], as well as cervical images [214], where the existing noise can make the analysis challenging.

Due to the existence of a large variety of methods for cell image segmentation, many of them providing acceptable performances, the proposed method can be used in conjunction with another method to enhance their performance. The constraints existing based on the prior knowledge can be evaluated on the segmented image provided by that method using the polytree technique. This can be done employing simple yet effective methods such as thresholding methods that have been applied to the segmentation of breast cancer TMA images, cervical images, HeLa cell line images, 3D confocal microscopy images and whole-slide lung cytology images, to name a number from the many examples. Using the images segmented by another method as the input, the oversegmentation step would be more accurate due to the homogeneity of the areas in the labelled image. However, it might be biased on the prior segmentation. Considering the segmented image along with the raw images and optimizing their weighting should be taken into account to minimize the bias on the new segmentation results.

Similarly, deformable models, such as active contours, have been popular in nucleus and cell segmentation. Contrary to the majority of superpixel generation algorithms, they can create concave areas by adapting their initial shape to the area they are surrounding. Replacing the superpixel generation by deformable models, the polytree inference framework developed in this thesis can be applied to different images, including drosophila cell segmentation in RNAi images [215], red blood cells in phase contrast images [216] and eukaryotic cells in confocal microscopy images [217], as examples previously analyzed be merely deformable models.

Many other graph based techniques have been applied to the cell segmentation problem, starting the analysis from making superpixels. One can replace the graph analysis parts in the previous methods by the graph modelling and inference presented in this thesis, for improved performance. Doing so, will broaden the application of the proposed method to other image datasets not evaluated in this work, such as colon fluorescence images [218], aortic endothelial cells in phase-contrast microscopy images [219] and yeast cells in brightfield microscopy images [220].

As described in the previous chapters, polytrees can model the prior knowledge to generate a constrained segmentation that follows the enforced rules. Other than improving the segmentation by incorporating the prior knowledge through this feature, these models can be used to evaluate biological assumptions about the objects being segmented. This is possible by setting the parameters of the model initially based on assumptions and comparing them with their values after optimization by the EM algorithm.

Using the weakly supervised version of the algorithm, a GUI-based framework can be designed such that the training phase of the algorithm is done based on a few annotations from the operator, making an interactive segmentation. Furthermore, samples from areas in the segmented images that are prone to error according to the labeling on the graphical model can be nominated to be verified/corrected by the operator, so that all the similar errors in the segmentation will be updated accordingly. This can contribute to addressing the problem of unavailability or limited availability of reliable training data that exists for many image datasets.

Further to their application to the image analysis problems, polytree graphical models can be used for modeling other phenomena that can be modeled as hierarchical structures. One possible application is the use in genetic networks with relationships between different entities including genes or individuals and the expression of certain genes in different generations.

5.3 Limitations and future work

In the current application of the method, the graph is generated according to the features taken from the superpixels and is fixed during the inference. This way the accuracy of the final output might be confined to the graph generation and errors made in the graph generation can propagate to the next stages of the method. To address this limitation, one can think of estimating the graph structure and considering the initial graph structure as *a priori*. Alternatively, the graph structure can be updated based on a first round of labeling and the labeling and graph tuning can iteratively continue to reach a convergence. Doing so, the graph structure and the inferred labels will be jointly optimized for a given image.

The accuracy of the superpixels in following the boundaries of the objects in the image can also reduce the overall segmentation performance. In chapter 4, it has been shown that this effect confines the maximum achievable Dice similarity coefficients to values less than $\%100$. Furthermore, performing the segmentation using other superpixel generation algorithms can be considered to reduce this effect. Benchmarking different superpixel generation algorithms and choosing the most suitable one for each application can improve the performance of the segmentation algorithm. This is because the cost functions of different oversegmentation algorithms, optimize different characteristics of the generated superpixels, which might or might not be suitable for a specific application.

The polytree structure proposed in this thesis is binary, that results in each node having two descendants (except the root nodes located in the bottommost layer of the graph). Using such a structure might not necessarily describe the connections between the superpixels in the image and results in the maximum number of layers in the graph and the maximum number of equations in the inference formulation. One possibility of future work is considering more descendants for each node that can make a better match with the image. A new inference algorithm will have to be derived for such a structure that can have the potential to be implemented for parallel processing. A more adaptive structure is having an arbitrary number of descendant nodes based on the spatial location of the superpixels in the image. This can be made possible by starting from an equal number of descendants for all nodes and tune the individual node connections through iterations that optimize the graph and inference.

It has been shown in this thesis that employment of the prior knowledge can improve segmentation. Results of comparing polytrees to trees also emphasize that more accurate modeling of the prior knowledge can generate more accurate segmentation results. However, it should be noted that this knowledge might not be available in some applications. A possible application of the graphical-model-based segmentation approach is validating biological hypotheses by comparing their effect in the segmentation results. If making use of a hypothesized prior knowledge improves the segmentation accuracy, this can be an indication of its correctness and vice versa. It should also be noted that different prior knowledges might be challenging or impossible to be modeled by polytrees and may need more complex structures or a combination of different types of graphs to be modeled. The limitation in enforcing the plausible label configurations all over the graph in current work is that labelings at certain areas of the graph might not obey the presumed configurations. This can be due to the imaging artifacts or staining errors that result in implausible object appearances in the image. Examples of this in cell images are cases where a nucleus can be seen within the background without the cell cytoplasm being visible. Imposing priors to such areas of the image introduces error to the segmentation which can result in an unacceptable performance in low quality or noisy images.

The probability distributions of classes of objects have been assumed to be Gaussian in this work. While this is a valid assumption for many cell types, more complex distributions can be employed to better explain the cellular objects. Dirichlet distributions have been implemented and experimented for modeling the intensity distribution in the superpixels while considering their sizes as part of this work. However, due to the large number of parameters that needed to be set and optimized for the model, the segmentation results have not been convincing.

Using deep features calculated by SegNet in chapter 4, the proposed algorithm could marginally outperform SegNet. This is due to the features and weights calculated by SegNet being optimized to minimize its cost. To address this problem, one line of future work can be designing a deep network with a cost function that is derived for the polytree model. Such a model will have the advantage of *explicitly* modeling and imposing the prior knowledge to the segmentation problem. This is possible due to performing a model-based segmentation, which is also less reliant on the training data, more efficient and therefore faster than using convolutional neural networks.

Another extension of the use of the model can be in studying the cell interactions over time. For example, by segmenting the hosts and pathogens and finding their inclusions in single time frames and looking at the evolution of the corresponding graphical models over time one can investigate the behavior of the immune system during infectious disease. The joint segmentation and tracking method can be also applied to other applications that involve migration and deformation of cells, for example in organ development [13].

Further to the analysis of cell interactions, the trajectories of cells of different types in microscopy videos could be extracted. Extracting patterns from the cell trajectories can be used for the classification of healthy and diseased samples. In addition, an atlas can be generated to predict the outcome of a given infection based on the existing labeled samples.

In this work, I have considered the inclusion of objects of different classes in the image, which is modeled through a child-parent relationship in the graph. However, different types of cells can have more complex relationships in their interactions, including being in another cell's proximity to exchange chemicals [221]. Also, there are interactions that can be identified by looking at the cells over time, such as cells being engulfed by or jumping out of, other cells or being exploded by the engulfed cells [222], some of which having unknown mechanisms. In order to account for these interactions, features other than the inclusion has to be considered in the graph generation leading to graphs with more complex structures and that possibly involve loops. Also, interactions over time can be modeled by mappings of the graph structures between successive time points. Even though the inference algorithm developed in this thesis is proposed for polytrees without loops, it can be extended to polytrees that have loops in them. Furthermore, the polytrees can be used for modeling the interrelations between cells and the factor graph [142] theory can be employed for the inference part.

The proposed model for segmentation was motivated by the need for modeling interactions between host and pathogen cells during infectious disease. Considering different types of cell interactions and other parameters contributing to the infection, an *in silico* version of the model of infectious disease can be implemented. Extension of such a system can be used for simulating the effects of drugs in infectious disease. Given the existing laboratory work and imaging being on animal models, the developed model can be extended to a pharmaceutical model to predict the drug effects on humans.

Other possible applications of the polytree structure for capturing prior knowledge can be emerged by changing the interpretation of the graph connections. Examples can vary from the causal effects of political and social circumstances on the economy to the emotional inputs of an individual and their relation to mental health. Another interesting example for the use of polytrees is modeling business hierarchies in organizations to study the influences of the members on one another.

Bibliography

[1] Aleksandra Bojarczuk, Katie A. Miller, Richard Hotham, Amy Lewis, Nikolay V. Ogryzko, Alfred A. Kamuyango, Helen Frost, Rory H. Gibson, Eleanor Stillman, Robin C. May, Stephen A. Renshaw, and Simon A. Johnston. Cryptococcus neoformans Intracellular Proliferation and Capsule Size Determines Early Macrophage Control of Infection. *Scientific Reports*, 6:21489, February 2016.

[2] David J Foran, Lin Yang, Wenjin Chen, Jun Hu, Lauri A Goodell, Michael Reiss, Fusheng Wang, Tahsin Kurc, Tony Pan, Ashish Sharma, et al. Imageminer: a software system for comparative analysis of tissue microarrays using content-based image retrieval, high-performance computing, and grid technology. *Journal of the American Medical Informatics Association*, 18(4):403–415, 2011.

[3] Daniel J Pambianco, John J Vargo, Ronald E Pruitt, Robert Hardi, and James F Martin. Computer-assisted personalized sedation for upper endoscopy and colonoscopy: a comparative, multicenter randomized study. *Gastrointestinal Endoscopy*, 73(4):765–772, 2011.

[4] H. Irshad, A. Veillard, L. Roux, and D. Racoceanu. Methods for Nuclei Detection, Segmentation, and Classification in Digital Histopathology: A Review-Current Status and Future Potential. *Biomedical Engineering, IEEE Reviews in*, 7:97–114, 2014.

[5] M. N. Gurcan, L. E. Boucheron, A. Can, A. Madabhushi, N. M. Rajpoot, and B. Yener. Histopathological Image Analysis: A Review. *IEEE Reviews in Biomedical Engineering*, 2:147–171, 2009.

[6] Gaorav P Gupta and Joan Massagué. Cancer metastasis: building a framework. *Cell*, 127(4):679–695, 2006.

[7] W. H. De Vos, L. Van Neste, B. Dieriks, G. H. Joss, and P. Van Oostveldt. High content image cytometry in the context of subnuclear organization. *Cytometry Part A*, 77A(1):64–75, January 2010.

[8] Murray B. Resnick, Tamako Konkin, Justin Routhier, Edmond Sabo, and Victor E. Pricolo. Claudin-1 is a strong prognostic indicator in stage II colonic cancer: a tissue microarray study. *Modern Pathology*, 18(4):511, April 2005.

[9] Thomas Misteli and Karen Meaburn. Prostate Cancer Diagnostics and Prognostics Based on Interphase Spatial Genome Positioning. Technical report, The Geneva Foundation Tacoma United States, The Geneva Foundation Tacoma United States, March 2016.

[10] Elizabeth R. Smith, Sophia H. George, Erin Kobetz, and Xiang-Xi Xu. New biological research and understanding of Papanicolaou's test. *Diagnostic Cytopathology*, 46(6):507–515, June 2018.

[11] Claudio Procaccini, Claudia La Rocca, Fortunata Carbone, Veronica De Rosa, Mario Galgani, and Giuseppe Matarese. Leptin as immune mediator: Interaction between neuroendocrine and immune system. *Developmental & Comparative Immunology*, 66:120–129, January 2017.

[12] Ali Tabesh, Mikhail Teverovskiy, Ho-Yuen Pang, Vinay P. Kumar, David Verbel, Angeliki Kotsianti, and Olivier Saidi. Multifeature prostate cancer diagnosis and Gleason grading of histological images. *IEEE transactions on medical imaging*, 26(10):1366–1378, 2007.

[13] Ting Wang, Yan Cui, Penggang Liu, Junfeng He, and Qian Zhang. Distribution of EGFR, BMP-2, and p53 in kidney of healthy newborn, adult and old highland-plateau yaks. *Folia Morphologica*, 2018.

[14] Luciano da Fontoura Costa, Leandro Carrijo Cintra, and David Schubert. An integrated approach to the characterization of cell movement. *Cytometry Part A: The Journal of the International Society for Analytical Cytology*, 68(2):92–100, 2005.

[15] Meghan K Driscoll and Gaudenz Danuser. Quantifying modes of 3d cell migration. *Trends in cell biology*, 25(12):749–759, 2015.

[16] Annemarie H Meijer, Astrid M van der Sar, Cristiana Cunha, Gerda EM Lamers, Mary A Laplante, Hiroshi Kikuta, Wilbert Bitter, Thomas S Becker, and Herman P Spaink. Identification and real-time imaging of a myc-expressing neutrophil population involved in inflammation and mycobacterial granuloma formation in zebrafish. *Developmental & Comparative Immunology*, 32(1):36–49, 2008.

[17] Susanne M Rafelski and Julie A Theriot. Crawling toward a unified model of cell motility: spatial and temporal regulation of actin dynamics. *Annual review of biochemistry*, 73(1):209–239, 2004.

[18] Yanhua Zheng and Zhimin Lu. Regulation of tumor cell migration by protein tyrosine phosphatase (PTP)-proline-, glutamate-, serine-, and threonine-rich sequence (PEST). *Chinese journal of cancer*, 32(2):75, 2013.

[19] Michael R Duchen and Gyorgy Szabadkai. Roles of mitochondria in human disease. *Essays in biochemistry*, 47:115–137, 2010.

[20] Sarah R Elkin, Ashley M Lakoduk, and Sandra L Schmid. Endocytic pathways and endosomal trafficking: a primer. *Wiener Medizinische Wochenschrift*, 166(7-8):196–204, 2016.

[21] Alberto Mantovani, Federica Marchesi, Alberto Malesci, Luigi Laghi, and Paola Allavena. Tumour-associated macrophages as treatment targets in oncology. *Nature reviews Clinical oncology*, 2017.

[22] JC Kagan and GM Barton. Emerging principles governing signal transduction by pattern-recognition receptors. *Cold Spring Harbor perspectives in biology*, 7(3):a016253–a016253, 2014.

[23] Xuan Xiao, Xiang Cheng, Shengchao Su, Qi Mao, and Kuo-Chen Chou. pLoc-mGpos: Incorporate Key Gene Ontology Information into General PseAAC for Predicting Subcellular Localization of Gram-Positive Bacterial Proteins. *Natural Science*, 9(9):330–349, 2017.

[24] A Fatica and I Bozzoni. Long non-coding RNAs: new players in cell differentiation and development. *Nature reviews. Genetics*, 15(1):7–21, 2014.

[25] Vebjorn Ljosa, Katherine L. Sokolnicki, and Anne E. Carpenter. Annotated high-throughput microscopy image sets for validation. *Nat Methods*, 9(7):637, 2012.

[26] D. Svoboda and V. Ulman. MitoGen: A Framework for Generating 3D Synthetic Time-Lapse Sequences of Cell Populations in Fluorescence Microscopy. *IEEE Transactions on Medical Imaging*, 36(1):310–321, January 2017.

[27] Zhi Lu, Gustavo Carneiro, and Andrew P. Bradley. An improved joint optimization of multiple level set functions for the segmentation of overlapping cervical cells. *IEEE Transactions on Image Processing*, 24(4):1261–1272, 2015.

[28] Saskia JE Suijkerbuijk and Jacco van Rheenen. From good to bad: Intravital imaging of the hijack of physiological processes by cancer cells. *Developmental biology*, 428(2):328–337, 2017.

[29] David C Benjamin and Richard O Hynes. Intravital imaging of metastasis in adult zebrafish. *BMC cancer*, 17(1):660, 2017.

[30] Cheng Huang, Jessica R Maxey, Supriyo Sinha, Joan Savall, Yiyang Gong, and Mark J Schnitzer. Long-term optical brain imaging in live adult fruit flies. *Nature Communications*, 9(1):872, 2018.

[31] Jun Chu, Russell D Haynes, Stéphane Y Corbel, Pengpeng Li, Emilio González-González, John S Burg, Niloufar J Ataie, Amy J Lam, Paula J Cranfill, Michelle A Baird, et al. Non-invasive intravital imaging of cellular differentiation with a bright red-excitable fluorescent protein. *Nature methods*, 11(5):572, 2014.

[32] David Entenberg, Jessica M Pastoriza, Maja H Oktay, Sonia Voiculescu, Yarong Wang, Maria Soledad Sosa, Julio Aguirre-Ghiso, and John Condeelis. Time-lapsed,

large-volume, high-resolution intravital imaging for tissue-wide analysis of single cell dynamics. *Methods*, 128:65–77, 2017.

[33] Ping Hu, Wenhua Zhang, Hongbo Xin, and Glenn Deng. Single cell isolation and analysis. *Frontiers in cell and developmental biology*, 4:116, 2016.

[34] Sarah K Trisdale, Nicolette M Schwab, Xiaoying Hou, John S Davis, and David H Townson. Molecular manipulation of keratin 8/18 intermediate filaments: modulators of FAS-mediated death signaling in human ovarian granulosa tumor cells. *Journal of ovarian research*, 9(1):8, 2016.

[35] TJ Mitchison and ED Salmon. Mitosis: a history of division. *Nature cell biology*, 3(1):E17, 2001.

[36] Raju Tomer, Khaled Khairy, Fernando Amat, and Philipp J Keller. Quantitative high-speed imaging of entire developing embryos with simultaneous multiview light-sheet microscopy. *Nature methods*, 9(7):755, 2012.

[37] Sebastian Bhakdi and Ponpan Thaicharoen. Easy employment and crosstalk-free detection of seven fluorophores in a widefield fluorescence microscope. *Methods and Protocols*, 1(2):20, 2018.

[38] Hossam Isack, Olga Veksler, Ipek Oguz, Milan Sonka, and Yuri Boykov. Efficient optimization for Hierarchically-structured Interacting Segments (HINTS). *arXiv:1703.10530 [cs]*, March 2017. arXiv: 1703.10530.

[39] Lysandra Voltaggio, Ashley Cimino-Mathews, Justin A Bishop, Pedram Argani, Jonathan D Cuda, Jonathan I Epstein, Ralph H Hruban, George J Netto, Mark H Stoler, Janis M Taube, et al. Current concepts in the diagnosis and pathobiology of intraepithelial neoplasia: A review by organ system. *Ca-A Cancer Journal for Clinicians*, 66(5):408–436, 2016.

[40] Gianluca Pegoraro and Tom Misteli. High-throughput imaging for the discovery of cellular mechanisms of disease. *Trends in Genetics*, 33(9):604–615, 2017.

[41] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.

[42] TW Ridler, S Calvard, et al. Picture thresholding using an iterative selection method. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 8(8):630–632, 1978.

[43] Luc Vincent and Pierre Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(6):583–598, 1991.

[44] Satoshi Tashiro, Joachim Walter, Akira Shinohara, Nanao Kamada, and Thomas Cremer. Rad51 accumulation at sites of DNA damage and in postreplicative chromatin. *The Journal of cell biology*, 150(2):283–292, 2000.

[45] Fredric S Fay, Krishan L Taneja, Shailesh Shenoy, Lawrence Lifshitz, and Robert H Singer. Quantitative digital analysis of diffuse and concentrated nuclear distributions of nascent transcripts, sc35 and poly (a). *Experimental Cell Research*, 231(1):27–37, 1997.

[46] Richard A Russell, Niall M Adams, David A Stephens, Elizabeth Batty, Kirsten Jensen, and Paul S Freemont. Segmentation of fluorescence microscopy images for quantitative analysis of cell nuclear architecture. *Biophysical Journal*, 96(8):3379–3389, 2009.

[47] Theodosios Goudas and Ilias Maglogiannis. An advanced image analysis tool for the quantification and characterization of breast cancer in microscopy images. *Journal of Medical Systems*, 39(3):1–13, 2015.

[48] Joakim Lindblad, Carolina Wählby, Ewert Bengtsson, and Alla Zaltsman. Image analysis for automatic segmentation of cytoplasms and classification of rac1 activation. *Cytometry Part A: the journal of the International Society for Analytical Cytology*, 57(1):22–33, 2004.

[49] Mehmet Sezgin and Bülent Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1):146–166, 2004.

[50] Miss Hetal J Vala and Astha Baxi. A review on Otsu image segmentation algorithm. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 2(2):pp–387, 2013.

[51] Jos BTM Roerdink and Arnold Meijster. The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta Informaticae*, 41(1, 2):187–228, 2000.

[52] Xiaobo Zhou, Fuhai Li, Jun Yan, and Stephen TC Wong. A novel cell segmentation method and cell phase identification using Markov model. *IEEE Transactions on Information Technology in Biomedicine*, 13(2):152–157, 2009.

[53] Fuhai Li, Xiaobo Zhou, Jinwen Ma, and Stephen TC Wong. Multiple nuclei tracking using integer programming for quantitative cancer cell cycle analysis. *IEEE transactions on medical imaging*, 29(1):96–105, 2010.

[54] Gang Lin, Umesh Adiga, Kathy Olson, John F Guzowski, Carol A Barnes, and Badrinath Roysam. A hybrid 3d watershed algorithm incorporating gradient cues and object models for automatic segmentation of nuclei in confocal image stacks. *Cytometry Part A: the journal of the International Society for Analytical Cytology*, 56(1):23–36, 2003.

[55] Rafael C Gonzalez, Richard E Woods, Steven L Eddins, et al. *Digital image processing using MATLAB.*, volume 624. Pearson-Prentice-Hall Upper Saddle River, New Jersey, 2004.

[56] Hui Kong, Hatice Cinar Akakin, and Sanjay E Sarma. A generalized Laplacian of Gaussian filter for blob detection and its applications. *IEEE transactions on cybernetics*, 43(6):1719–1733, 2013.

[57] Min Zhang, Teresa Wu, and Kevin M Bennett. Small blob identification in medical images using regional features from optimum scale. *IEEE transactions on biomedical engineering*, 62(4):1051–1062, 2015.

[58] Cheng Chen, Wei Wang, John A Ozolek, and Gustavo K Rohde. A flexible and robust approach for segmenting cell nuclei from 2d microscopy images using supervised learning and template matching. *Cytometry Part A*, 83(5):495–507, 2013.

[59] Roland T Chin and Charles R Dyer. Model-based recognition in robot vision. *ACM Computing Surveys (CSUR)*, 18(1):67–108, 1986.

[60] Anil K Jain, Yu Zhong, and Marie-Pierre Dubuisson-Jolly. Deformable template models: A review. *Signal processing*, 71(2):109–129, 1998.

[61] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.

[62] Tony F Chan and Jianhong Jackie Shen. *Image processing and analysis: variational, PDE, wavelet, and stochastic methods*, volume 94. Siam, 2005.

[63] Filiz Bunyak, Kannappan Palaniappan, Sumit Kumar Nath, TL Baskin, and Gang Dong. Quantitative cell motility for in vitro wound healing using level set-based active contour tracking. In *Biomedical Imaging: Nano to Macro, 2006. 3rd IEEE International Symposium on*, pages 1040–1043. IEEE, 2006.

[64] Bo Zhang, Christophe Zimmer, and J-C Olivo-Marin. Tracking fluorescent cells with coupled geometric active contours. In *Biomedical Imaging: Nano to Macro, 2004. IEEE International Symposium on*, pages 476–479. IEEE, 2004.

[65] Sumit K Nath, Kannappan Palaniappan, and Filiz Bunyak. Cell segmentation using coupled level sets and graph-vertex coloring. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 101–108. Springer, 2006.

[66] Christopher M. Bishop and others. *Pattern Recognition and Machine Learning*, volume 4. Springer New York, 2006.

[67] Carlos Arteta, Victor Lempitsky, J Alison Noble, and Andrew Zisserman. Learning to detect cells using non-overlapping extremal regions. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 348–356. Springer, 2012.

[68] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004.

[69] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[70] Li Deng, Dong Yu, et al. Deep learning: methods and applications. *Foundations and Trends® in Signal Processing*, 7(3–4):197–387, 2014.

[71] Holger R Roth, Le Lu, Ari Seff, Kevin M Cherry, Joanne Hoffman, Shijun Wang, Jiamin Liu, Evrim Turkbey, and Ronald M Summers. A new 2.5 d representation for lymph node detection using random sets of deep convolutional neural network observations. In *International conference on medical image computing and computer-assisted intervention*, pages 520–527. Springer, 2014.

[72] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[73] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1915–1929, 2013.

[74] Yann LeCun, Koray Kavukcuoglu, Clément Farabet, et al. Convolutional networks and applications in vision. In *ISCAS*, volume 2010, pages 253–256, 2010.

[75] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

[76] Dan Ciresan, Alessandro Giusti, Luca M Gambardella, and Jürgen Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In *Advances in neural information processing systems*, pages 2843–2851, 2012.

[77] Dan C Cireşan, Alessandro Giusti, Luca M Gambardella, and Jürgen Schmidhuber. Mitosis detection in breast cancer histology images with deep neural networks. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, pages 411–418. Springer, 2013.

[78] V. Badrinarayanan, A. Kendall, and R. Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Scene Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(39):2481–2495, 2017.

[79] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[80] Stuart Lloyd. Least squares quantization in PCM. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[81] Neelam Sinha and AG Ramakrishnan. Automation of differential blood count. In *TENCON 2003. Conference on Convergent Technologies for the Asia-Pacific Region*, volume 2, pages 547–551. IEEE, 2003.

[82] James Monaco, Jason Hipp, D Lucas, S Smith, Ulysses Balis, and Anant Madabhushi. Image segmentation with implicit color standardization using spatially constrained expectation maximization: Detection of nuclei. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 365–372. Springer, 2012.

[83] Caroline Petitjean and Jean-Nicolas Dacher. A review of segmentation methods in short axis cardiac MR images. *Medical image analysis*, 15(2):169–184, 2011.

[84] Daniel García-Lorenzo, Simon Francis, Sridar Narayanan, Douglas L Arnold, and D Louis Collins. Review of automatic segmentation methods of multiple sclerosis white matter lesions on conventional magnetic resonance imaging. *Medical image analysis*, 17(1):1–18, 2013.

[85] Mohd Ali Balafar, Abdul Rahman Ramli, M Iqbal Saripan, and Syamsiah Mashohor. Review of brain MRI image segmentation methods. *Artificial Intelligence Review*, 33(3):261–274, 2010.

[86] Cigdem Demir and Bülent Yener. Automated cancer diagnosis based on histopathological images: a systematic survey.

[87] Lei He, L Rodney Long, Sameer Antani, and G Thoma. Computer assisted diagnosis in histopathology. *Sequence and genome analysis: methods and applications*, 3:271–287, 2010.

[88] Mitko Veta, Josien PW Pluim, Paul J Van Diest, and Max A Viergever. Breast cancer histopathology image analysis: A review. *IEEE Transactions on Biomedical Engineering*, 61(5):1400–1411, 2014.

[89] Thomas J Fuchs and Joachim M Buhmann. Computational pathology: Challenges and promises for tissue analysis. *Computerized Medical Imaging and Graphics*, 35(7-8):515–530, 2011.

[90] Sonal Kothari, John H Phan, Todd H Stokes, and May D Wang. Pathology imaging informatics for quantitative analysis of whole-slide images. *Journal of the American Medical Informatics Association*, 20(6):1099–1108, 2013.

[91] Mohammadreza Mostajabi, Payman Yadollahpour, and Gregory Shakhnarovich. Feedforward semantic segmentation with zoom-out features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3376–3385, 2015.

[92] R Desikan, F Ségonne, et al. A computer generated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest. *Human Brain Mapping*, 2005.

[93] Brian B Avants, Charles L Epstein, Murray Grossman, and James C Gee. Symmetric diffeomorphic image registration with cross-correlation: evaluating automated labeling of elderly and neurodegenerative brain. *Medical image analysis*, 12(1):26–41, 2008.

[94] F Segonne and B Fischl. Integration of topological constraints in medical image segmentation. In *Handbook of Biomedical Imaging*, pages 245–262. Springer, 2015.

[95] Vicent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic active contours. *International journal of computer vision*, 22(1):61–79, 1997.

[96] Christos Davatzikos, Xiaodong Tao, and Dinggang Shen. Hierarchical active shape models, using the wavelet transform. *IEEE transactions on medical imaging*, 22(3):414–423, 2003.

[97] Evgeny Strekalovskiy, Claudia Nieuwenhuis, and Daniel Cremers. Nonmetric priors for continuous multilabel optimization. In *European Conference on Computer Vision*, pages 208–221. Springer, 2012.

[98] Daniel Freedman and Tao Zhang. Interactive graph cut based segmentation with shape priors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 755–762. IEEE, 2005.

[99] Pierre-Louis Bazin and Dzung L. Pham. Topology preserving tissue classification with fast marching and topology templates. In *Biennial International Conference on Information Processing in Medical Imaging*, pages 234–245. Springer, 2005.

[100] Greg Slabaugh and Gozde Unal. Graph cuts segmentation using an elliptical shape prior. In *Image Processing, 2005. ICIP 2005. IEEE International Conference On*, volume 2, pages II–1222. IEEE, 2005.

[101] Olga Veksler. Star Shape Prior for Graph-Cut Image Segmentation. In *Computer Vision – ECCV 2008*, Lecture Notes in Computer Science, pages 454–467. Springer, Berlin, Heidelberg, October 2008.

[102] Xiaoqing Liu, Olga Veksler, and Jagath Samarabandu. Graph cut with ordering constraints on labels and its applications. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008.

[103] Pedro F. Felzenszwalb and Olga Veksler. Tiered scene labeling with dynamic programming. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3097–3104. IEEE, 2010.

[104] K. Li, X. Wu, D. Z. Chen, and M. Sonka. Optimal surface segmentation in volumetric images–a graph-theoretic approach., Optimal Surface Segmentation in Volumetric Images—A Graph-Theoretic Approach. *IEEE transactions on pattern analysis and*

machine intelligence, IEEE transactions on pattern analysis and machine intelligence, 28, 28(1, 1):119, 119–134, January 2006.

[105] Y. Yin, X. Zhang, R. Williams, X. Wu, D. D. Anderson, and M. Sonka. LOGISMOS-Layered Optimal Graph Image Segmentation of Multiple Objects and Surfaces: Cartilage Segmentation in the Knee Joint. *IEEE Transactions on Medical Imaging*, 29(12):2023–2037, December 2010.

[106] Aurlien Lucchi, Carlos Becker, Pablo Mrquez Neila, and Pascal Fua. Exploiting enclosing membranes and contextual cues for mitochondria segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 65–72. Springer, 2014.

[107] Chanho Jung, Changick Kim, Seoung Wan Chae, and Sukjoong Oh. Unsupervised segmentation of overlapped nuclei using Bayesian classification. *IEEE Transactions on Biomedical Engineering*, 57(12):2825–2832, 2010.

[108] S. Farhand, R. B. Montero, X. Vial, D. T. Nguyen, M. Reardon, S. M. Pham, F. M. Andreopoulos, and G. Tsechpenakis. Probabilistic multi-compartmenty geometric model: Application to cell segmentation. In *IEEE 9th International Symposium on Biomedical Imaging (ISBI)*, pages 174–177, May 2012.

[109] Kishore Mosaliganti, Arnaud Gelas, Alexandre Gouaillard, Ramil Noche, Nikolaus Obholzer, and Sean Megason. Detection of spatially correlated objects in 3d images using appearance models and coupled active contours. In *Medical Image Computing and Computer-Assisted InterventionMICCAI 2009*, pages 641–648. Springer, 2009.

[110] Simon Li, James Wakefield, and J. Alison Noble. Automated segmentation and alignment of mitotic nuclei for kymograph visualisation. In *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*, pages 622–625. IEEE, 2011.

[111] Ali Gooya, Kilian M. Pohl, Michel Bilello, Luigi Cirillo, George Biros, Elias R. Melhem, and Christos Davatzikos. GLISTR: Glioma Image Segmentation and Registration. *IEEE transactions on medical imaging*, 31(10):1941–1954, October 2012.

[112] Frederik O. Kaster, Bjoern H. Menze, Marc-Andr Weber, and Fred A. Hamprecht. Comparative validation of graphical models for learning tumor segmentations from noisy manual annotations. In *International MICCAI Workshop on Medical Computer Vision*, pages 74–85. Springer, 2010.

[113] Mohamed L. Seghier, Anil Ramlackhansingh, Jenny Crinion, Alexander P. Leff, and Cathy J. Price. Lesion identification using unified segmentation-normalisation models and fuzzy clustering. *NeuroImage*, 41(4):1253–1266, July 2008.

[114] S. M. Ali Eslami, Nicolas Heess, Christopher K. I. Williams, and John Winn. The Shape Boltzmann Machine: A Strong Model of Object Shape. *International Journal of Computer Vision*, 107(2):155–176, April 2014.

[115] Fei Chen, Huimin Yu, Roland Hu, and Xunxun Zeng. Deep learning shape priors for object segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1870–1877. IEEE, 2013.

[116] Aïcha BenTaieb and Ghassan Hamarneh. Topology aware fully convolutional networks for histology gland segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 460–468. Springer, 2016.

[117] Hao Chen, Xiaojuan Qi, Lequan Yu, Qi Dou, Jing Qin, and Pheng-Ann Heng. DCAN: Deep contour-aware networks for object instance segmentation from histology images. *Medical Image Analysis*, 36:135–146, February 2017.

[118] H. Ravishankar, R. Venkataramani, S. Thiruvenkadam, P. Sudhakar, and V. Vaidya. Learning and incorporating shape models for semantic segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 203–211. Springer, 2017.

[119] Ozan Oktay, Enzo Ferrante, Konstantinos Kamnitsas, Mattias Heinrich, Wenjia Bai, Jose Caballero, Stuart Cook, Antonio de Marvao, Timothy Dawes, Declan O'Regan, Bernhard Kainz, Ben Glocker, and Daniel Rueckert. Anatomically Constrained Neural Networks (ACNN): Application to Cardiac Image Enhancement and Segmentation. *IEEE Transactions on Medical Imaging*, pages 1–1, 2017.

[120] David W. Shattuck and Richard M. Leahy. Automated graph-based analysis and correction of cortical volume topology. *IEEE transactions on medical imaging*, 20(11):1167–1177, 2001.

[121] Xiao Han, Chenyang Xu, Ulisses Braga-Neto, and Jerry L. Prince. Topology correction in brain cortex segmentation using a multiscale, graph-based algorithm. *IEEE Transactions on Medical Imaging*, 21(2):109–121, 2002.

[122] Nir BenZadok, Tammy Riklin-Raviv, and Nahum Kiryati. Interactive level set segmentation for image-guided therapy. In *IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI)*, pages 1079–1082. IEEE, 2009.

[123] Victor Lempitsky, Pushmeet Kohli, Carsten Rother, and Toby Sharp. Image segmentation with a bounding box prior. In *Computer Vision, 2009 IEEE 12th International Conference On*, pages 277–284. IEEE, 2009.

[124] Ke Li, Bharath Hariharan, and Jitendra Malik. Iterative instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3659–3667, 2016.

[125] Zoltan Kato and Josiane Zerubia. Markov Random Fields in Image Segmentation. *Foundations and Trends in Signal Processing*, 5(12):1–155, October 2012.

[126] Lei Zhang and Qiang Ji. Image Segmentation with a Unified Graphical Model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1406–1425, August 2010.

[127] Martin J. Wainwright, Michael I. Jordan, and others. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(12):1–305, 2008.

[128] Shann-Ching Chen, Ting Zhao, Geoffrey J. Gordon, and Robert F. Murphy. A novel graphical model approach to segmenting cell images. In *2006 IEEE Symposium on Computational Intelligence and Bioinformatics and Computational Biology*, pages 1–8. IEEE, 2006.

[129] Mustafa Gokhan Uzunbas, Chao Chen, and Dimitris Metaxas. An efficient conditional random field approach for automatic and interactive neuron segmentation. *Medical Image Analysis*, 27:31–44, January 2016.

[130] Fabian Rathke, Stefan Schmidt, and Christoph Schnrr. Probabilistic Intra-Retinal Layer Segmentation in 3-D OCT Images Using Global Shape Regularization. *Medical Image Analysis*, 18(5):781–794, 2014.

[131] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *ACM SIGCOMM computer communication review*, volume 29, pages 251–262. ACM, 1999.

[132] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the web. *Computer networks*, 33(1-6):309–320, 2000.

[133] Duncan J Watts and Steven H Strogatz. Collective dynamics of small-worldnetworks. *nature*, 393(6684):6350–6376, 1998.

[134] Stefan Wuchty, Erszébet Ravasz, and Albert-László Barabási. The architecture of biological networks. In *Complex systems science in biomedicine*, pages 165–181. Springer, 2006.

[135] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT Press, 2009.

[136] Ross P Kindermann and J Laurie Snell. On the relation between Markov random fields and social networks. *Journal of Mathematical Sociology*, 7(1):1–13, 1980.

[137] Zhenyu Wu and Richard Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 15(11):1101–1113, 1993.

[138] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[139] Umberto Bertele and Francesco Brioschi. *Nonserial dynamic programming*. Academic Press, 1972.

[140] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Reasoning*. Morgan Kaufmann Publishers, Los Altos, 1988.

[141] Brendan J Frey, J Frey Brendan, and Brendan J Frey. *Graphical models for machine learning and digital communication*. MIT Press, 1998.

[142] Frank R Kschischang, Brendan J Frey, and H-A Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on information theory*, 47(2):498–519, 2001.

[143] Jonathan S Yedidia, William T Freeman, and Yair Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on information theory*, 51(7):2282–2312, 2005.

[144] Vladimir Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE transactions on pattern analysis and machine intelligence*, 28(10):1568–1583, 2006.

[145] Steffen L Lauritzen and David J Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 157–224, 1988.

[146] Greg CG Wei and Martin A Tanner. A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85(411):699–704, 1990.

[147] George Casella and Edward I George. Explaining the Gibbs sampler. *The American Statistician*, 46(3):167–174, 1992.

[148] Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[149] Dorothy M Greig, Bruce T Porteous, and Allan H Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 271–279, 1989.

[150] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23(11):1222–1239, 2001.

[151] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized-via graph cuts? *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 26(2):147–159, 2004.

[152] Yousef Al-Kofahi, Wiem Lassoued, William Lee, and Badrinath Roysam. Improved automatic detection and segmentation of cell nuclei in histopathology images. *IEEE Transactions on Biomedical Engineering*, 57(4):841–852, 2010.

[153] Hang Chang, Leandro A Loss, and Bahram Parvin. Nuclear segmentation in H&E sections via multi-reference graph cut (MRGC). In *IEEE 9th International Symposium in Biomedical Imaging (ISBI)*, pages 614–617. IEEE, 2012.

[154] Youyi Song, Ling Zhang, Siping Chen, Dong Ni, Baiying Lei, and Tianfu Wang. Accurate segmentation of cervical cytoplasm and nuclei based on multiscale convolutional network and graph partitioning. *IEEE Transactions on Biomedical Engineering*, 62(10):2421–2433, 2015.

[155] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, Sabine Süsstrunk, et al. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012.

[156] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE transactions on pattern analysis and machine intelligence*, 26(9):1124–1137, 2004.

[157] Huiguang Yang and Narendra Ahuja. Automatic segmentation of granular objects in images: Combining local density clustering and gradient-barrier watershed. *Pattern Recognition*, 47(6):2266–2279, 2014.

[158] Ning Xu, Narendra Ahuja, and Ravi Bansal. Object segmentation using graph cuts based active contours. *Computer Vision and Image Understanding*, 107(3):210–224, 2007.

[159] Vinh Thong Ta, Olivier Lezoray, and Abderrahim Elmoataz. Graph based semi and unsupervised classification and segmentation of microscopic images. In *IEEE International Symposium on Signal Processing and Information Technology*, pages 1160–1165. IEEE, 2007.

[160] Vinh-Thong Ta, Olivier Lézoray, Abderrahim Elmoataz, and Sophie Schüpp. Graph-based tools for microscopic cellular image segmentation. *Pattern Recognition*, 42(6):1113–1125, 2009.

[161] Gang Lin, Monica K Chawla, Kathy Olson, John F Guzowski, Carol A Barnes, and Badrinath Roysam. Hierarchical, model-based merging of multiple fragments for improved three-dimensional segmentation of nuclei. *Cytometry Part A: The Journal of the International Society for Analytical Cytology*, 63(1):20–33, 2005.

[162] Alberto Santamaria-Pang, Yuchi Huangy, and Jens Rittscher. Cell segmentation and classification via unsupervised shape ranking. In *IEEE 10th International Symposium on Biomedical Imaging (ISBI)*, pages 406–409. IEEE, 2013.

[163] Asli Kale and Selim Aksoy. Segmentation of cervical cell images. In *Proceedings of the 2010 20th International Conference on Pattern Recognition*, pages 2399–2402. IEEE Computer Society, 2010.

[164] H Gökhan Akçay and Selim Aksoy. Automatic detection of geospatial objects using multiple hierarchical segmentations. *IEEE Transactions on Geoscience and Remote Sensing*, 46(7):2097–2111, 2008.

[165] Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.

[166] Finn V. Jensen. *An Introduction to Bayesian Networks*, volume 210. UCL Press London, 1996.

[167] Jean-Marc Laferté, Patrick Pérez, and Fabrice Heitz. Discrete Markov image modeling and inference on the quadtree. *IEEE Transactions on Image Processing*, 9(3):390–404, 2000.

[168] Farid Flitti, Christophe Collet, and Annabelle Joannic-Chardin. Unsupervised multiband image segmentation using hidden Markov quadtree and copulas. In *Image Processing, 2005. ICIP 2005. IEEE International Conference On*, volume 2, pages II–634. IEEE, 2005.

[169] Bruno Oliveira, Sandro Queirós, Pedro Morais, Helena R. Torres, João Gomes-Fonseca, Jaime C. Fonseca, and João L. Vilaça. A novel multi-atlas strategy with dense deformation field reconstruction for abdominal and thoracic multi-organ segmentation from computed tomography. *Medical Image Analysis*, 45:108–120, April 2018.

[170] Benjamin Perret and Ch Collet. Connected image processing with multivariate attributes: An unsupervised Markovian classification approach. *Computer Vision and Image Understanding*, 133:1–14, 2015.

[171] Xiaojuan Feng, C.K.I. Williams, and S.N. Felderhof. Combining belief networks and neural networks for scene segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):467–483, April 2002.

[172] C.A. Bouman and M. Shapiro. A multiscale random field model for Bayesian image segmentation. *IEEE Transactions on Image Processing*, 3(2):162–177, March 1994.

[173] W. W. Irving, P. W. Fieguth, and A. S. Willsky. An overlapping tree approach to multiscale stochastic modeling and estimation. *IEEE Transactions on Image Processing*, 6(11):1517–1529, November 1997.

[174] Aaron W Dennis. Algorithms for learning the structure of monotone and nonmonotone sum-product networks. *Algorithms*, 2016:12–22, 2016.

[175] Sinisa Todorovic and Michael C Nechyba. Interpretation of complex scenes using dynamic tree-structure bayesian networks. *Computer Vision and Image Understanding*, 106(1):71–84, 2007.

[176] Jia Li, R. M. Gray, and R. A. Olshen. Multiresolution image classification by hierarchical modeling with two-dimensional hidden Markov models. *IEEE Transactions on Information Theory*, 46(5):1826–1841, August 2000.

[177] Christopher KI Williams and Nicholas J Adams. DTs: dynamic trees. In *Advances in neural information processing systems*, pages 634–640, 1999.

[178] A. J. Slorkey and C. K. L. Williams. Image modeling with position-encoding dynamic trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):859–871, July 2003.

[179] Sinisa Todorovic and Michael C. Nechyba. Dynamic trees for unsupervised segmentation and matching of image regions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(11):1762–1777, 2005.

[180] Brendan J Frey and Nebojsa Jojic. Advances in algorithms for inference and learning in complex probability models.

[181] Pei Zhang, Guorong Wu, Yaozong Gao, Pew-Thian Yap, and Dinggang Shen. A dynamic tree-based registration could handle possible large deformations among MR brain images. *Computerized Medical Imaging and Graphics*, 52(Supplement C):1–7, September 2016.

[182] A. Delong and Y. Boykov. Globally optimal segmentation of multi-region objects. In *2009 IEEE 12th International Conference on Computer Vision*, pages 285–292, September 2009.

[183] Y. Yin, X. Zhang, R. Williams, X. Wu, D. D. Anderson, and M. Sonka. LOGISMOS-Layered Optimal Graph Image Segmentation of Multiple Objects and Surfaces: Cartilage Segmentation in the Knee Joint. *IEEE Transactions on Medical Imaging*, 29(12):2023–2037, December 2010.

[184] Hossam Isack, Olga Veksler, Ipek Oguz, Milan Sonka, and Yuri Boykov. Efficient optimization for Hierarchically-structured Interacting Segments (HINTS). *arXiv:1703.10530 [cs]*, March 2017.

[185] Fujun Liu, Fuyong Xing, and Lin Yang. Robust muscle cell segmentation using region selection with dynamic programming. In *IEEE 11th International Symposium on Biomedical Imaging (ISBI)*, pages 521–524. IEEE, 2014.

[186] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2011.

[187] Michal Aharon, Michael Elad, Alfred Bruckstein, et al. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311, 2006.

[188] Josie F. Gibson and Simon A. Johnston. Immunity to Cryptococcus neoformans and C. gattii during cryptococcosis. *Fungal Genetics and Biology*, 78:76–86, May 2015.

[189] K. Kampa, D. Putthividhya, and J. C. Principe. Irregular Tree-Structured Bayesian Network for image segmentation. In *2011 IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6, September 2011.

[190] Elchanan Mossel, Sbastien Roch, and Allan Sly. Robust estimation of latent tree graphical models: Inferring hidden states with inexact parameters. *IEEE transactions on information theory*, 59(7):4357–4373, 2013.

[191] Le Song, Han Liu, Ankur Parikh, and Eric Xing. Nonparametric Latent Tree Graphical Models: Inference, Estimation, and Structure Learning. *arXiv:1401.3940 [stat]*, January 2014. arXiv: 1401.3940.

[192] Michael Van den Bergh, Xavier Boix, Gemma Roig, Benjamin de Capitani, and Luc Van Gool. SEEDS: Superpixels Extracted via Energy-Driven Sampling. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision ECCV 2012*, number 7578 in Lecture Notes in Computer Science, pages 13–26. Springer Berlin Heidelberg, 2012.

[193] H. Fehri, A. Gooya, S. A. Johnston, and A. F. Frangi. Multi-class Image Segmentation in Fluorescence Microscopy Using Polytrees. In *Information Processing in Medical Imaging*, volume 10265, pages 517–528. Springer, Cham, 2017.

[194] Liang Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018.

[195] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2890. IEEE, 2017.

[196] Michael Van den Bergh, Xavier Boix, Gemma Roig, and Luc Van Gool. SEEDS: Superpixels Extracted via Energy-Driven Sampling. *arXiv:1309.3848 [cs]*, September 2013.

[197] Jan Funke, Chong Zhang, Tobias Pietzsch, and Stephan Saalfeld. The Candidate Multi-Cut for Cell Segmentation. *ArXiv e-prints*, 1707:arXiv:1707.00907, July 2017.

[198] Tony Lindeberg. Scale-space theory: A basic tool for analyzing structures at different scales. *Journal of applied statistics*, 21(1-2):225–270, 1994.

[199] Lee R. Dice. Measures of the Amount of Ecologic Association Between Species. *Ecology*, 26(3):297–302, July 1945.

[200] Patrice Y. Simard, David Steinkraus, and John C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *ICDAR*, volume 3, pages 958–962, 2003.

[201] OpenCV library. https://opencv.org/.

[202] ITK - Segmentation & Registration Toolkit. https://itk.org/.

[203] Boost C++ Libraries. https://www.boost.org/.

[204] FSL - FslWiki. https://fsl.fmrib.ox.ac.uk/fsl/fslwiki.

[205] Gnu.org. https://www.gnu.org/software/gsl/.

[206] M. S. Zaveri and D. Hammerstrom. CMOL/CMOS Implementations of Bayesian Polytree Inference: Digital and Mixed-Signal Architectures and Performance/Price. *IEEE Transactions on Nanotechnology*, 9(2):194–211, March 2010.

[207] Yi Wang, Nevin L. Zhang, Tao Chen, and Leonard KM Poon. LTC: A latent tree approach to classification. *International journal of approximate reasoning*, 54(4):560–572, 2013.

[208] K. Kampa, J.C. Principe, D. Putthividhya, and A. Rangarajan. Data-driven tree-structured Bayesian network for image segmentation. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2213–2216, March 2012.

[209] Christian Wolf and Gérald Gavin. Inference and parameter estimation on hierarchical belief networks for image segmentation. *Neurocomputing*, 73(4):563–569, 2010.

[210] F. Lehmann. Turbo Segmentation of Textured Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):16–29, January 2011.

[211] Chanho Jung and Changick Kim. Segmenting clustered nuclei using h-minima transform-based marker extraction and contour parameterization. *IEEE transactions on biomedical engineering*, 57(10):2600–2604, 2010.

[212] Cecilia Zanella, Matteo Campana, Barbara Rizzi, Camilo Melani, Gonzalo Sanguinetti, Paul Bourgine, Karol Mikula, Nadine Peyriéras, and Alessandro Sarti. Cells segmentation from 3-d confocal images of early zebrafish embryogenesis. *IEEE transactions on Image Processing*, 19(3):770–781, 2010.

[213] Nathan D Lawson and Brant M Weinstein. In vivo imaging of embryonic vascular development using transgenic zebrafish. *Developmental biology*, 248(2):307–318, 2002.

[214] Youyi Song, Ee-Leng Tan, Xudong Jiang, Jie-Zhi Cheng, Dong Ni, Siping Chen, Baiying Lei, and Tianfu Wang. Accurate cervical cell segmentation from overlapping clumps in pap smear images. *IEEE transactions on medical imaging*, 36(1):288–300, 2017.

[215] Christian Schmitt-Engel, Dorothea Schultheis, Jonas Schwirz, Nadi Ströhlein, Nicole Troelenberg, Upalparna Majumdar, Daniela Grossmann, Tobias Richter, Maike Tech, Jürgen Dönitz, et al. The ibeetle large-scale rnai screen reveals gene functions for insect development and physiology. *Nature communications*, 6:7822, 2015.

[216] Rym Mehri, Catherine Mavriplis, and Marianne Fenech. Red blood cell aggregates and their effect on non-newtonian blood viscosity at low hematocrit in a two-fluid low shear rate microfluidic system. *PloS one*, 13(7):e0199911, 2018.

[217] Xin Li, Anil K Gorle, Tracy D Ainsworth, Kirsten Heimann, Clifford E Woodward, J Grant Collins, and F Richard Keene. Rna and dna binding of inert oligonuclear ruthenium (ii) complexes in live eukaryotic cells. *Dalton Transactions*, 44(8):3594–3603, 2015.

[218] Shuya Yano, Kiyoto Takehara, Shinji Miwa, Hiroyuki Kishimoto, Yukihiko Hiroshima, Takashi Murakami, Yasuo Urata, Shunsuke Kagawa, Michael Bouvet, Toshiyoshi Fujiwara, et al. Improved resection and outcome of colon-cancer liver metastasis with fluorescence-guided surgery using in situ gfp labeling with a telomerase-dependent adenovirus in an orthotopic mouse model. *PLoS One*, 11(2):e0148760, 2016.

[219] Ha-Rim Seo, Hyo Eun Jeong, Hyung Joon Joo, Seung-Cheol Choi, Chi-Yeon Park, Jong-Ho Kim, Ji-Hyun Choi, Long-Hui Cui, Soon Jun Hong, Seok Chung, et al. Intrinsic fgf2 and fgf5 promotes angiogenesis of human aortic endothelial cells in 3d microfluidic angiogenesis system. *Scientific reports*, 6:28832, 2016.

[220] Cristian Versari, Szymon Stoma, Kirill Batmanov, Artémis Llamosi, Filip Mroz, Adam Kaczmarek, Matt Deyell, Cédric Lhoussaine, Pascal Hersen, and Gregory Batt. Long-term tracking of budding yeast cells in brightfield microscopy: Cellstar and the evaluation platform. *Journal of The Royal Society Interface*, 14(127):20160705, 2017.

[221] Chloe PVJ Martens. Beyond the structure: Deciphering the molecular mechanisms of secondary transport with hydrogen-deuterium exchange mass spectrometry. *Biophysical Journal*, 114(3):334a, 2018.

[222] Rhys A Watkins, Alexandre Andrews, Charlotte Wynn, Caroline Barisch, Jason S King, and Simon A Johnston. Cryptococcus neoformans escape from dictyostelium amoeba by both wash-mediated constitutive exocytosis and vomocytosis. *Frontiers in cellular and infection microbiology*, 8:108, 2018.