Medical Image Segmentation: Thresholding and Minimum Spanning Trees

Simeon Sahani Mayala

Thesis for the degree of Philosophiae Doctor (PhD)
University of Bergen, Norway
2023

UNIVERSITY OF BERGEN

Medical Image Segmentation: Thresholding and Minimum Spanning Trees

Simeon Sahani Mayala

Thesis for the degree of Philosophiae Doctor (PhD)
at the University of Bergen

Date of defense: 18.08.2023

© Copyright Simeon Sahani Mayala

The material in this publication is covered by the provisions of the Copyright Act.

Year: 2023

Title: Medical Image Segmentation: Thresholding and Minimum Spanning Trees

Name: Simeon Sahani Mayala

Print: Skipnes Kommunikasjon / University of Bergen

Preface

This dissertation is submitted as a partial fulfillment of the requirements for the degree of Doctor of Philosophy (PhD) in Mathematics at the Department of Mathematics, University of Bergen. The scientific works in the thesis have been conducted under the supervision of Assoc. Prof. Morten Brun (Department of Mathematics) and Senior Researcher Dr. Sonia Gavasso (Neuro-SysMed, Department of Neurology, Haukeland University Hospital and Department of Clinical Medicine) at the University of Bergen, Norway.

Acknowledgment

I would like to give my heartfelt thanks to the faculty of Natural and Mathematical Sciences at the University of Bergen for offering me the opportunity to be a research fellow at the Department of Mathematics. I recognize their exceptional support for the time of my studies. Also, I would like to thank my supervisors Assoc. Prof. Morten Brun and Senior researcher Dr. Sonia Gavasso for their advisory supervision in accomplishing this work. Thank you very much.

I would like to thank all the co-authors from the research group at the center of excellence Neuro-SysMed, Haukeland University Hospital for their collaborations. Thank you Assoc. Prof. Nello Blaiser, Dr. Ida Herdlevær, Dr. Shamundeeswari Anandan, and Jonas Bull Haugsøen for everything we shared during our Friday meetings.

I would like to extend my sincere thanks to the Department mathematics community, everyone in the administration, the cleaners of our office, Professors and students in the topology group, my fellows PhD students, and all my friends including the master and Bachelor students. Everyone contributed to the accomplishment of this work. May God reward you a hundredfold.

Also, I would like to express my thanks to my friends outside the university in Norway and my family and friends back home in Tanzania. Your support and encouragement are highly appreciated. Special thanks to my beloved mother for your kindness and motherly love. Thank you very much. Also, I pray for the soul of my departed father; may his soul rest in peace.

Abstract

In image segmentation, an image is divided into separate objects or regions. It is an essential step in image processing to define areas of interest for further processing or analysis.

The segmentation process reduces the complexity of an image to simplify the analysis of the attributes obtained after segmentation. It changes the representation of the information in the original image and presents the pixels in a way that is more meaningful and easier to understand.

Image segmentation has various applications. For medical images, the segmentation process aims to extract the image data set to identify areas of the anatomy relevant to a particular study or diagnosis of the patient. For example, one can locate affected or abnormal parts of the body. Segmentation of follow-up data and baseline lesion segmentation is also very important to assess the treatment response.

There are different methods used for image segmentation. They can be classified based on how they are formulated and how the segmentation process is performed. The methods include those based on threshold values, edge-based, cluster-based, model-based and hybrid methods, and methods based on machine learning and deep learning. Other methods are based on growing, splitting and merging regions, finding discontinuities in the edge, watershed segmentation, active contours and graph-based methods.

In this thesis, we have developed methods for segmenting different types of medical images. We tested the methods on datasets for white blood cells (WBCs) and magnetic resonance images (MRI). The developed methods and the analysis performed on the image data set are presented in three articles.

In Paper A we proposed a method for segmenting nuclei and cytoplasm from white blood cells. The method estimates the threshold for segmentation of nuclei automatically based on local minima. The method segments the WBCs before segmenting the cytoplasm depending on the complexity of the objects in the image. For images where the WBCs are well separated from red blood cells (RBCs), the WBCs are segmented by taking the average of n images that were already filtered with a threshold value. For images where RBCs overlap the WBCs, the entire WBCs are segmented using simple linear iterative clustering (SLIC) and watershed methods. The cytoplasm is obtained by subtracting the segmented nucleus from the segmented WBC. The method is tested on two different publicly available datasets, and the re-

sults are compared with state of the art methods.

In Paper B, we proposed a method for segmenting brain tumors based on minimum spanning tree (MST) concepts. The method performs interactive segmentation based on the MST. In this paper, the image is loaded in an interactive window for segmenting the tumor. The region of interest and the background are selected by clicking to split the MST into two trees. One of these trees represents the region of interest and the other represents the background. The proposed method was tested by segmenting two different 2D brain T1-weighted magnetic resonance image data sets. The method is simple to implement and the results indicate that it is accurate and efficient.

In Paper C, we propose a method that processes a 3D MRI volume and partitions it into brain, non-brain tissues, and background segments. It is a graph-based method that uses MST to separate the 3D MRI into the brain, non-brain, and background regions. The graph is made from a preprocessed 3D MRI volume followed by constructing the MST. The segmentation process produces three labeled connected components which are reshaped back to the shape of the 3D MRI. The labels are used to segment the brain, non-brain tissues, and the background. The method was tested on three different publicly available data sets and the results were compared to different state of the art methods.

Abstrakt

I bildesegmentering deles et bilde i separate objekter eller regioner. Det er et essensielt skritt i bildebehandling for å definere interesseområder for videre behandling eller analyse.

Oppdelingsprosessen reduserer kompleksiteten til et bilde for å forenkle analysen av attributtene oppnådd etter segmentering. Det forandrer representasjonen av informasjonen i det opprinnelige bildet og presenterer pikslene på en mte som er mer meningsfull og lettere å forstå.

Bildesegmentering har forskjellige anvendelser. For medisinske bilder tar segmenteringsprosessen sikte på å trekke ut bildedatasettet for å identifisere områder av anatomien som er relevante for en bestemt studie eller diagnose av pasienten. For eksempel kan man lokalisere berørte eller anormale deler av kroppen. Segmentering av oppfølgingsdata og baseline lesjonssegmentering er også svrt viktig for å vurdere behandlingsresponsen.

Det er forskjellige metoder som blir brukt for bildesegmentering. De kan klassifiseres basert på hvordan de er formulert og hvordan segmenteringsprosessen utføres. Metodene inkluderer de som er baserte på terskelverdier, graf-baserte, kant-baserte, klynge-baserte, modell-baserte og hybride metoder, og metoder basert p maskinlæring og dyp læring. Andre metoder er baserte på å utvide, splitte og legge sammen regioner, å finne diskontinuiteter i randen, vannskille segmentering, aktive kontuter og graf-baserte metoder.

I denne avhandlingen har vi utviklet metoder for å segmentere forskjellige typer medisinske bilder. Vi testet metodene på datasett for hvite blodceller (WBCs) og magnetiske resonansbilder (MRI). De utviklede metodene og analysen som er utført på bildedatasettet er presentert i tre artikler.

I artikkel A (Paper A) foreslo vi en metode for segmentering av nukleuser og cytoplasma fra hvite blodceller. Metodene estimerer terskelen for segmentering av nukleuser automatisk basert på lokale minima. Metoden segmenterer WBC-ene før segmentering av cytoplasma avhengig av kompleksiteten til objektene i bildet. For bilder der WBC-ene er godt skilt fra røde blodlegemer (RBC), er WBC-ene segmentert ved å ta gjennomsnittet av n bilder som allerede var filtrert med en terskelverdi. For bilder der RBC-er overlapper WBC-ene, er hele WBC-ene segmentert ved hjelp av enkle lineære iterative klynger (SLIC) og vannskillemetoder. Cytoplasmaet oppnås ved å trekke den segmenterte nukleusen fra den segmenterte WBC-en. Metoden

testes på to forskjellige offentlig tilgjengelige datasett, og resultatene sammenlignes med toppmoderne metoder.

I artikkel B (Paper B) foreslo vi en metode for segmentering av hjernesvulster basert på minste dekkende tre-konsepter (minimum spanning tree, MST). Metoden utfører interaktiv segmentering basert på MST. I denne artikkelen er bildet lastet inn i et interaktivt vindu for segmentering av svulsten. Fokusregion og bakgrunn skilles ved å klikke for å dele MST i to trær. Ett av disse trærne representerer fokusregionen og det andre representerer bakgrunnen. Den foreslåtte metoden ble testet ved å segmentere to forskjellige 2D-hjerne T1 vektede magnetisk resonans bildedatasett. Metoden er enkel å implementere og resultatene indikerer at den er nøyaktig og effektiv.

I artikkel C (Paper C) foreslår vi en metode som behandler et 3D MRI-volum og deler det i hjernen, ikke-hjernevev og bakgrunnsegmenter. Det er en grafbasert metode som bruker MST til å skille 3D MRI inn i de tre regiontypene. Grafen lages av et forhåndsbehandlet 3D MRI-volum etterfulgt av konstrueringen av MST-en. Segmenteringsprosessen gir tre merkede, sammenkoblende komponenter som omformes tilbake til 3D MRI-form. Etikettene brukes til å segmentere hjernen, ikke-hjernevev og bakgrunn. Metoden ble testet på tre forskjellige offentlig tilgjengelige datasett og resultatene ble sammenlignet med ulike toppmoderne metoder.

Outline

This thesis is organized into two parts. Part I contains general background and part II presents the papers contributing to this thesis.

Part I presents the introductory chapters in which the reader is guided through different concepts covering medical images and segmentation methods. These concepts contribute to the prior understanding of the methods developed for solving the problem of segmenting medical images. The thesis presents an application of mathematical concepts for solving problems arising from the medical field.

Chapter 1 contains motivational ideas and a general introduction to the broad audience. It presents image segmentation concepts, research questions, contributions, and objectives of image segmentation. Also, it briefly highlights the types of image segmentation, the popular segmentation methods, and the application of image segmentation. It ends by explaining the medical images especially those analyzed in this study.

Chapter 2 defines digital images and their mathematical representation for both 2D and 3D. Experts in image analysis will be familiar with the content in this chapter. The chapter is intended for mathematicians who are not familiar with image analysis. Different operations commonly applied to digital images are presented with examples for clarity. The mathematical operations presented in this chapter include point, local and global operations. Also, it defines erosion and dilation, closing and opening, and ends with image filtering.

Chapter 3 is for the domain experts in addition to the intended audience for chapter 2. The methods developed in the papers are also briefly presented in section 3.2.2.2, 3.8.5.3, and 3.8.5.4. It categorizes different image segmentation methods by explaining and providing examples to show their differences.

Chapter 4 is also intended for the experts in image analysis in addition to the audience for chapter 2. It describes techniques for evaluating newly developed segmentation methods. It also presents the state of the art methods compared to the classical methods.

Chapter 5 is intended for the experts in image analysis because that is where we give our contribution. It provides an overview and summary of the included papers. It presents a discussion that connects the research questions presented in Chapter 2 by comparing them to the conclusions of the work in Part II.

Part II contains the scientific research work and the contribution consists of the following listed scientific papers:

List of papers

Paper A: **S. Mayala** and JB. Haugsøen, *Threshold estimation based on local minima for nucleus and cytoplasm segmentation.* BMC Medical Imaging **29** (2022), 22:77.

Paper B: **S. Mayala**, I. Herdlevær, JB. Haugsøen, S. Anandan, S. Gavasso, and M. Brun, *Brain Tumor Segmentation Based on Minimum Spanning Tree.* Frontiers in Signal Processing **2** (2022), 816186.

Paper C: **S. Mayala,** I. Herdlevær, JB. Haugsøen, S. Anandan, N. Blaiser, S. Gavasso, and M. Brun, *GUBS: Graph-based Unsupervised Brain Segmentation in MRI images.* MDPI Journal of imaging **8** (2022), 262.

Contents

Part I

Background

Chapter 1

Introduction

The main theme of this thesis is medical image segmentation of microscopic images of white blood cells (WBCs) and magnetic resonance images (MRI). In this chapter we give motivation and preliminary concepts to understand the segmented image data sets from two different technologies.

1.1 Motivation and Preliminary Concepts

1.1.1 Motivation

Computing platforms for biological data from different imaging modalities have generated a massive amount of information stored in the form of images which are useful for research and discoveries of unknown mechanisms in human body systems. The generated images need to be analyzed to attain certain intended goals. From these images, objects of interest such as tumors, normal and cancerous cells are segmented and analyzed to identify different problems in the human body. There are different techniques applied for segmenting the images.

Image segmentation is a process that makes transitions by outputting attributes that are extracted from the images. The attributes extracted in the segmentation are further analyzed in high-level processing to understand the hidden information in the image. It is an essential step in image processing because it aims at segmenting the regions of interest (ROI) for further processing. The segmentation process represents the image in a simple form that is easier to understand and simple to analyze.

The use of segmentation has shown a great need in different fields of application. Areas of application include object detection and recognition, identification of plate numbers, robotics, image-based search, and autonomous vehicles. For medical images, the examples of application include the quantification of brain volume and localization of different pathology. Also, studying the anatomical structure of different parts of the body, and computer-aided diagnosis and treatment. The application can vary based on the problem and the form of image data to be segmented.

It is important to have accurate segmentation in medical image analysis because of its use especially in computer-aided diagnosis, surgery planning for treatment, and other medical research such as segmenting normal and diseased cells. So, the attributes segmented accurately will foster a higher-level analysis and give a more accurate understanding of the information presented in the images. Inaccurate segmentation for medical images is dangerous because it can provide misleading information. Also, the quality of the data presented from different imaging modalities can compromise the results. So, the quality of the data sets needs to be good enough to produce nice results.

Accuracy for any segmentation method in medical imaging is necessary for it to be reliable. Manual segmentation performed by human beings is reliable. However, manual segmentation is time-consuming and tedious. There is a need for automated, fast, and reliable segmentation algorithms to avoid tedious work. Different methods for segmenting images need to be tested and justified if their results are reliable. It is common to evaluate newly developed segmentation methods by different strategies.

Also, the results of the newly developed methods need to be compared to the results obtained using the state of the art methods. But most of the state of the art methods depend on ground truth which are mostly obtained manually. Due to the insufficiency of ground truth information many newly developed and automatic methods are tested on limited information. So, some methods are likely to work on certain kind of data. This is a problem of many methods failing to generalize on data set produced from different image modalities. The automated methods need to produce accurate results at the level of manual segmentation in terms of performance. Also, for automated methods to be fully integrated in the application of clinical settings they need to perform well independent from the manual segmentation.

Furthermore, the images generated from the imaging modalities are presented differently because of some factors. The factors could be the problem to be addressed and the condition of the patients being scanned. But also, the complexity of tissue structures to be analyzed from the acquired data sets is a problem. For example, segmenting brain tissues to isolate them from non-brain tissues MRI might be more challenging than segmenting nuclei from microscopic images of normal WBCs.

Also, machines have different parameters to be set during data acquisition. This can impact the quality of data produced from different machines. Some images have artifacts possibly introduced in the signals during the acquisition. Furthermore, the variability of structures or the shapes for the object of interest to be segmented is challenging because the information are non-homogeneous. Also, the low contrast of the captured signal between different anatomical structure is a challenge. Due to these differences, a method is likely to perform well on one data set and fail on another data set. It is not guaranteed that two methods working well on one data set will perform with the same accuracy when tested on another different data set. The general situation of many existing methods is most likely streamlined to the problem at hand and the nature of the data.

With regards to the listed challenges above, segmenting medical images is neither

an easy nor a straightforward task. On the other hand medical image segmentation has proved multiple listed applications in the medical field and the general goal of these application is to improve the patient outcomes. Because of this reason we are motivated to develop methods for segmenting the medical images. The developed methods are expected to be easier to explain but whose performance is comparable to the performance of the state of the art methods. We are also motivated to develop methods which are simple and flexible to be adapted for segmenting images from different imaging modalities. Simple methods which give good performance are easy to be adapted in clinical settings.

In this work, we contribute to the problem of image segmentation in two different ways. We proposed an approach that estimates the threshold automatically and uses it for segmenting the nucleus and cytoplasm of the microscopic image of normal WBCs. We also propose a graph-based approach and use its minimum spanning tree to solve two problems. First, we apply the approach to perform an interactive segmentation of brain tumors from 2D MRI. Secondly, we also apply the approach to segment brain volume from 3D MRI.

1.1.2 Research Questions

The purpose of this thesis is to research the problem of medical image segmentation. So, the main research question is presented as follows: Can we develop methods based on classical techniques for segmenting medical images and attain the accuracy of the state of the art methods?

The main research question may be responded to by implementing different specific questions. So, these questions following from the main research question are answered in the research to contribute to the problem of medical image segmentation which is the main focus of this thesis.

The first specific question is stated as follows: What are the existing methods used for segmenting medical images? This question helps to review and understand if there is any chance of either improving the existing or developing new methods.

The second specific question is stated as follows: What are the ways of representing information captured from different imaging models? It is important to understand the theory of image representation before segmentation.

The third specific question states as follows: Can we develop simple explainable and competitive methods and apply them for segmenting regions of interest from medical images? The expected outcome of this question is the description of the developed research methods, analysis, and application of these methods for segmenting medical images. It is important to check the performance and the suitability of the developed methods on different data sets.

1.1.3 Contributions

In this thesis, various methods have been developed for segmenting medical images. These include an automatic thresholding method for segmenting the nucleus and

cytoplasm, a graph-based method utilizing a minimum spanning tree for segmenting brain tumors from MRI, and a Graph-Based Unsupervised Brain Segmentation (GUBS) for segmenting 3D brain volume. The obtained results are compared to results obtained by using deep learning methods.

Even though deep learning methods have revolutionized the classical methods in terms of accuracy, this fact does not make the classical image segmentation methods obsolete. Some segmentation problems do not necessarily need deep learning methods because classical techniques can give a better solution. The results obtained in Paper A by the thresholding method show that the nucleus and cytoplasm are segmented with high accuracy. Also, deep learning methods present high accuracy for this kind of image. So, comparing the classical methods and deep learning, the former is simple.

For all the image data sets segmented in the three papers the accuracy is relatively higher for deep learning but not extremely far from the performance of classical methods. Different from the classical methods, deep learning methods will obtain the results at the cost of computational resources and a long time for training the model. The methods based on classical ideas especially those developed in this thesis do not need a huge amount of data for training models, also do not spend long time for training the model compared to deep learning methods. For example, the results obtained using deep learning for comparing the classical method results presented in papers A, B, and C have been obtained after augmenting to increase the data for training the models.

The segmentation methods developed in this thesis are based on thresholding and minimum spanning trees. These algorithms are based on simple concepts and can easily be adapted to other problems. For these methods, it is easy to tweak any parameters to improve the algorithm for better performance compared to deep learning methods. The deep learning methods consist of parameters in the order of millions inside it and each parameter is connected with complex interrelationships. The classical methods developed in this thesis offer complete transparency in their processes compared to deep learning methods which have poor interpretability. This fact limits the application of deep learning for clinical decisions because any use case in health care needs to be auditable and explainable to pass any test.

The advantage of the methods developed in this thesis do not require labels for training the model. The ground truth is needed for evaluating the performance of the developed method. It compares the obtained results to the ground truth to check the level of accuracy. On the contrary, deep learning methods require annotated data for training the model. For huge data sets, annotation of the data is time-consuming and tedious task. So, this is a drawback that makes a difficult adaptation of deep learning methods in clinical settings.

This subsection discusses the classical methods developed in this thesis compared to the deep learning methods. The results presented in each paper show that the classical image segmentation methods are not obsolete but still relevant in this era of deep learning application. Both groups of methods have their strength and

weaknesses. The choice of the most suitable approach for a given medical image segmentation problem need to consider the practical feasibility.

1.1.4 Image segmentation

Image segmentation is a process that partitions an image into non-overlapping segments based on its features and other properties. The process focuses on separating the region of interest from the other regions considered to be the background [32, 33]. The process is done by dividing the pixels in an image into different regions or categories, which correspond to different objects in the image [61]. In computer vision, it is the prime research area that aims at partitioning an image into its constituent region of interests [75].

The process is achieved by assigning same labels to pixels which share or have similar properties in the image. The features with similar properties in the same area are expected to be consistent when the labels are assigned to the pixels. Likewise, areas with different properties show clear separation. Generally, any method developed for segmenting images is expected to account for challenges such as complexity of region considered to be the background, variation within the same class, unclear boundaries between the ROI and the background as well as illumination variation within the image [75].

The good segmentation is achieved when pixels in the same category or region have similar gray level values and neighboring pixels which are in different categories have dissimilar values. This idea can be summarized symbolically by

$$\bigcup_{i=1}^{N} R_i = I \text{ and } R_i \bigcap R_{i+1} = \emptyset, \text{ for } i \in [1, N]. \tag{1.1}$$

where R_i represents a region in the image I. The union of different regions to form the image indicates that the segmented regions must cover the whole image and the intersection part show that the segmented regions should be non-overlapping.

Figure 1.1 shows different examples of regions of interest (second row) segmented from their corresponding input images (first row).

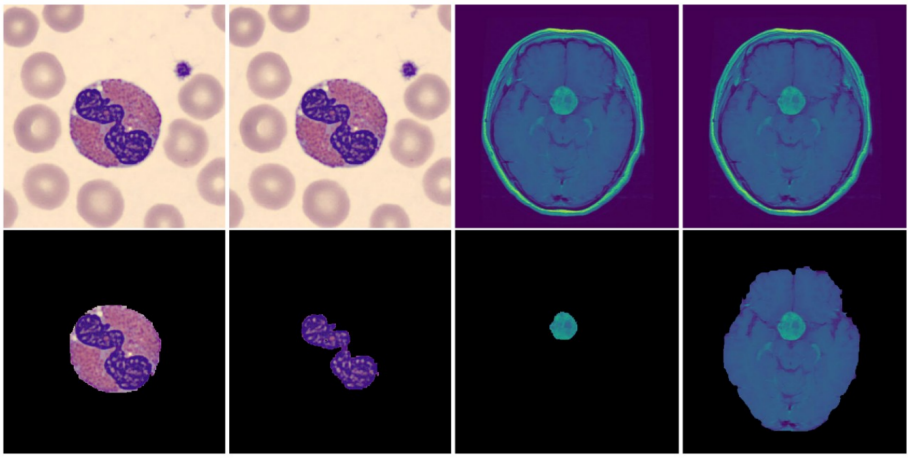


Figure 1.1: **Examples of segmented images in second row.** *(a) Segmented white blood cell (eosinophil cell), (b) Segmented nucleus of eosinophil cell, (c) segmented tumor from the representative axial MRI section, (d) segmented brain from the representative axial MRI section. Original images (row one) from [1] for WBC and [12] for MRI image.*

Generally, the task of segmenting an image is achieved after performing two related tasks which are recognition and delineation [119]. Recognition is a qualitative task of determining important information from the object of interest in the image. Delineation is a quantitative task of specifying the precise location and extent of the objects boundary in the image [119]. The task of recognition can highly depend on a knowledgeable person but computer algorithms can perform well the delineation task.

The extraction of quantitative information from an image to make the attributes might be challenging if the task is carried out on the whole image and not working with the objects which compose a group of pixels in the image. A quality image segmentation simplifies the next step in image analysis. So, the segmentation step is very important in image analysis [32, 106] because it makes a transition from image processing methods that output images. Instead, it focuses on methods that output attributes that are extracted from the images [33].

1.1.5 Objectives of image segmentation

The process of image segmentation from the input image outputs attributes which are considered as input in high level processing. There could be different objectives of image segmentation to be achieved but in the book [106] they are summarized into two main objectives:-

- Image segmentation decomposes the image into parts for further analysis. Most literature highlight this fact as the primary objective of image segmentation. During this step, a digital image is broken down into several segments to reduce the complexity of the image and simplify the analysis of the attributes obtained from segmentation.

- Image segmentation changes the representation of the information in the input image. The pixels in input image are organized or presented in a way that is more meaningful and easier to understand. But also organizing the information in a more efficient way to ease the possible analysis that are carried on the high-level processing [106].

In the process of image segmentation the two objectives are seemingly inseparable in the implementation. The process shows the narrow possibility of achieving the first objective without implementing the second objective. Primarily, image segmentation remains a prerequisite for more advanced analysis of image data [115].

In this thesis, the main objective is to develop methods based on classical ideas that can efficiently segment targeted regions from medical images with accuracy comparable to the state of the art methods. The specific objectives which are expected to be covered include:

- To review different methods which are applied for segmenting medical images.

- To understand the representation and complexity of different anatomical structures of images captured from different imaging modals.

- To develop methods that can be used for segmenting regions of interest from medical images and compare the performance of the developed methods to the state of the art methods.

The main objective together with its specific objective will be achieved by implementing and presenting in a paper-wise style.

1.1.6 Types of image segmentation

Image segmentation is divided into semantic, instance, and panoptic segmentation. The division of image segmentation into different groups is done by considering the different coarse and fine granularity of segmentation [61].

1.1.6.1 Semantic segmentation

Semantic segmentation is the process of identifying classes of objects found in the image. The objects found in the image are classified into semantic categories such as cells, people, trees, and so on. It classifies each pixel in the image to differentiate the classes in that image. To differentiate it from instance segmentation, semantic segmentation assigns the same labels to different instances. This is a type of segmentation in which multiple instances of the same class are not differentiated. It obtains

pixel-level of the whole image classification without differentiating different objects belonging to the same class [139].

1.1.6.2 Instance segmentation

This is a type of segmentation that identifies and classifies each pixel into its belonging instances. It combines object detection and semantic segmentation and discriminates instances of the same class [132]. It assigns pixel-level segmentation masks to each object [139]. To differentiate it from semantic segmentation, instance segmentation assigns unique labels to instances.

1.1.6.3 Panoptic segmentation

This is a type of segmentation that combines semantic and instance segmentation [51]. It is related to semantic segmentation in the sense that both identify the belonging class of every pixel. But differs from semantic segmentation because it distinguishes different instances of the same class [51]. Panoptic segmentation produces semantic labels which are known as classes and instance ids [139].

Figure 1.2 presents segmented images comparing semantic, instance and panoptic segmentation.

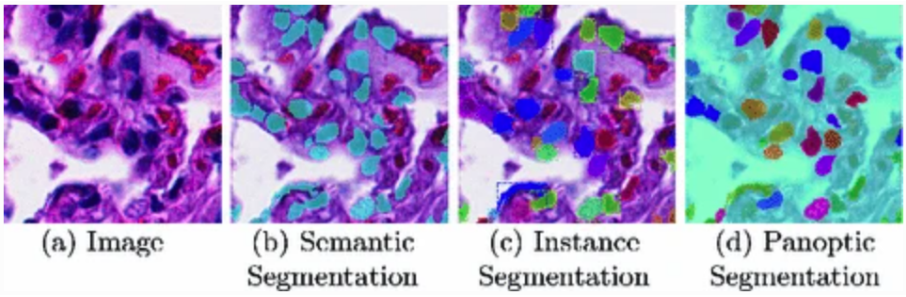


Figure 1.2: **Examples of different type of image segmentation.** *Illustration to differentiate between semantic, instance and panoptic segmentation. Figure obtained from [139].*

1.1.7 Popular Method in image segmentation

Different popular methods are used for segmenting images. Some of the popular methods for image segmentation include thresholding, graph-based, Superpixels-based, and deep learning-based methods. These methods are briefly defined here but they are detailed in chapter 3 together with other methods.

Thresholding is a widely used method for segmentation because they are simple to be applied. These methods are based on a value(s) obtained either manually or automatically to categorize intensity values in the images based on the range of

values. Intensity values in the region of interest are assigned the same label and those in the background are assigned the same label [60].

The graph-based methods for image segmentation represent the problem in terms of a graph. In this method, each pixel in the image is mapped to a node or vertex in a graph and each edge in the graph connects a pair of neighboring pixels [27]. The edge weights between two vertices are the measure of similarity between two neighboring pixels. Based on the application there are different variations.

The methods based on superpixel work by considering similarity measures defined using perceptual features. This type of method creates groups of pixels that are close or relatively similar. One of the popular methods based on superpixels is simple linear iterative clustering (SLIC). The methods show good performance and each superpixel should adhere to the boundaries [2]. Superpixels are created by minimizing a cost function defined over the graph.

Deep learning methods are machine learning techniques that teach a computer to perform a specific task. These methods have highly influenced the analysis of medical images showing state of the art accuracy [85]. For image segmentation, Convolutional neural networks (CNNs) are among the deep learning methods that are used for processing image data.

1.1.8 Application of image segmentation

Image segmentation has been applied in a number of fields. It is one of the most essential step in image processing because it separates the region of interest from the non interesting objects [36]. We briefly present application of image segmentation on medical images and other applications.

1.1.8.1 Segmentation of medical images

Medical images are different from other normal pictures [53]. They show body attributes that can not be accessed in simple ways. Any kind of analysis performed on such images is guided by prerequisites that give rise to acquiring the images in the first place. Medical image segmentation focuses on extracting the regions of interest from 3D image data (usually visualized as 2D). The image data comes from different scanning technology.

In medical images analysis, the segmentation step appears to be useful in a range of applications which include locating affected or abnormal parts of the body such as tumor detection and segmentation [72, 126], brain segmentation [30] and fracture identification [105] in case of accidents. Also, segmentation is applied on normal and abnormal cells for research [71] and diagnosis [15]. It is also applied on measuring tissue volumes [130]. Also, for medical images, the segmentation process is useful for extracting the image data set for particular study, or diagnosis of the patient, and it helps during decision making of treatment plan [107].

For example, the segmentation of follow-up data and the baseline lesion segmentation is very important for exploring and assessing the potential for treatment

response [76]. Also, segmentation is very useful in the process of detecting and classifying cancer from microscopic biopsy images. It is a useful step before features extraction and classification [56].

The application can vary based on the problem and the form of image data to be segmented. In this thesis, we mainly focus on segmenting medical images in which we first segment normal white blood cells (WBCs) from 2D microscopic images. We also segment brain tumors from 2D T1W MRI, and lastly, segment brain from 3D T1W MRI.

1.1.8.2 Other application of image segmentation

Also, there are other applications of image segmentation. These include object detection and recognition, identification of number plates, image-based search, robotics, autonomous vehicles and intelligent video [3, 19, 33, 62].

For object detection, segmentation is applied to face detection, pedestrian detection, locating objects in satellite images such as roads, forests, crops, and so on [19]. It is also used for recognition tasks including face recognition, fingerprint recognition, and Iris recognition. Image segmentation is also useful for video object co-segmentation and action localization [62].

1.2 Medical imaging

The imaging concept can be defined as any technique that can be utilized to obtain biological information within two (x, y), three (x, y, z) morphological dimensions [134]. Not limited to that, the longitudinal studies include the time which creates a fourth dimension.

The definition of medical imaging in many studies considers the technology and processes which are intended for imaging the human body for clinical analysis especially, monitoring and treatment of medical conditions. Even though the idea is mostly influenced by the dominance of imaging in electromagnetic spectrum but there are other imaging modalities [33].

Different famous techniques which are developed based on electromagnetic techniques include X-rays, computerized tomography (CT) scan, positron emission tomography (PET) scan, and Magnetic resonance imaging (MRI). Other imaging modalities include ultrasound, and those based on electron microscopy, light and fluorescent microscopy techniques [33, 100].

Generally, the choice of imaging technique highly depends on the resolution needed for the structures of interest to be observed and their categorization may vary based on their application or the technology used. Since the categorization of imaging technique is not the focus of this thesis we limit ourselves to only two types of images. The next section presents a short understanding of two different types of the medical images which have been analyzed in the papers included in this thesis.

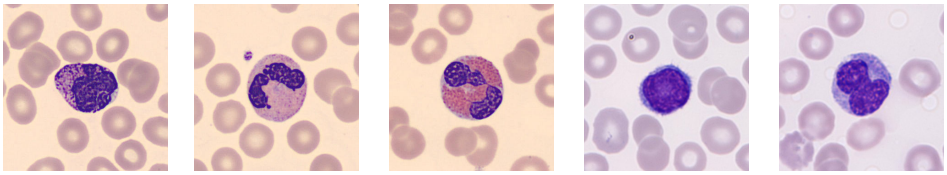
The first data sets are microscopic images of normal WBCs and the second data sets are MRI images.

1.2.1 Microscopic images

A digital microscopic image is an image taken through a microscope device to show a magnified image of an object. They are used for pathological diagnosis in tissue specimen on the cell level [53]. They are usually colored image showing a good contrast of the objects on the image. The image files come in various formats including Tagged Image File Format (TIFF), Portable Network Graphic (PNG), Joint Graphics Expert Group (JPEG). The sizes of these images vary greatly depending on the type of microscopes used for their acquiring and the acquisition technique.

Because of their high quality and resolution, the microscopic images are useful in different field of analysis such as cell counting, analyzing shape and structures of cells. Also, the microscopic images are used for analyzing the distribution of cells [53]. Microscopic imaging plays an important role in predicting and detecting diseases within the body.

Figure 1.3 presents examples of different microscopic images of the normal white blood cells.



(a) Basophil (b) Neutrophil (c) Eosinophil (d) Lymphocyte (e) Monocyte

Figure 1.3: **Examples of normal white blood cells.** *The cells seen in the background are red blood cells identified by their red color, thin (often almost transparent) center, and lack of nuclei. Images obtained from [1].*

1.2.2 Magnetic Resonance Images

Magnetic resonance imaging (MRI) is a medical imaging procedure that employs magnetic field and radio frequency signals to create images showing anatomical structures (diseased or without disease), and of different functions in the human body [113]. It is a non-invasive technique for viewing sections through living tissues [32]. The images produced by MRI highlight the anatomical structures within the body and they are commonly referred to as structural MRI and are represented as 3D data sets. Also, the same technology acquires metabolic function information stored in 4D by using functional magnetic resonance imaging (fMRI).

Different information for MRI data are acquired based on the suggested protocols, parameters and planning. However, the specifics vary depending on the factors

such as the MRI hardware and software used, radiologist's and referrer's preference, institutional protocols, patient factors (example allergy) and time constraints [5]. Mainly, the processes including signal acquisition followed by image reconstruction are required to produce an MR images [113]. We define MRI sequences, slice thickness, and slice interval.

- **MRI sequences:** MRI images are acquired using different sequences which include T1-weighted and T2-weighted scans. The sequences have different properties of the tissues which determine the contrast and brightness of the image. The Cerebral Spinal Fluid (CSF) is useful for differentiating T1 and T2 weighted MRI. The CSF appears to be dark on T1-weighted imaging and bright on T2-weighted imaging [68]. Also, there are other sequences called Fluid Attenuated Inversion Recovery (Flair) [68], diffusion-weighted imaging (DWI) [5].

- **Slice thickness:** Another important information about the MRI image is the slice thickness. Slice thickness (often axial) determines the trade-off in image quality between spatial resolution (how one can differentiate small changes in the image) and image noise (the standard deviation of the image) [112, 124]. The increase in Slice thickness decreases the spatial resolution and image noise.

  The choice of the slice thickness depends on what one wants to show. If the interest is to show large soft-tissue objects then less image noise is preferred. Whereas if the interest is to visualize tiny structures then a higher spatial resolution is needed [112].

- **Slice Interval** is the distance between the center of two adjacent slices, and it ultimately determines the number of images in a series [112, 124].

  - **Contiguous:** Where one slice ends the next slice starts. That is, the slice interval equal to the thickness.
  - **Non-Contiguous:** Some areas of anatomy between two adjacent slices are missed. That is slice interval is greater than slice thickness. This creates fewer images in the series.
  - **Overlapped:** Some areas of anatomy appear in two adjacent slices. That is slice interval is less than slice thickness.This creates more images in the series.

In image acquisition, voxel size is one of the major considerations. The voxel is obtained by subdividing each slice of tissue into rows and columns of individual volume elements (voxels) and the quality of the image is highly affected by the voxel size [113]. To obtain image pixels, the image is also divided into rows and columns of picture elements (pixels) which in turn represents a corresponding voxel within the slice [113].

Figure 1.4 shows MRI slices of a representative subject from the cancer imaging archive (TCIA) database [13, 54]. The Slice thickness is 5mm and the interval is 6.5mm. This is non-contiguous because slice interval is greater than slice thickness.

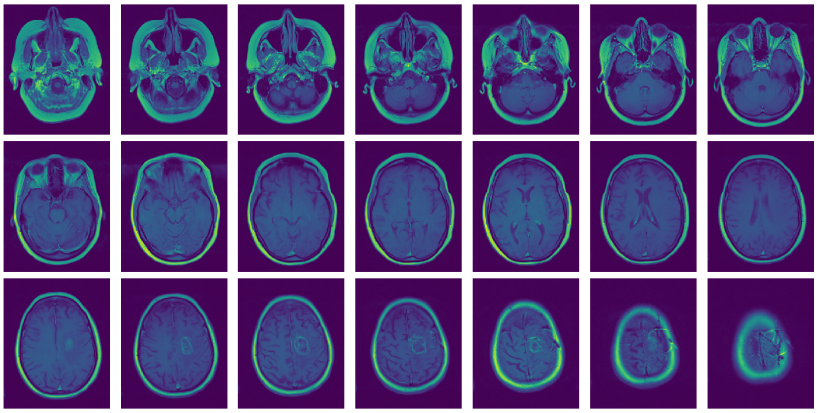


Figure 1.4: **Examples of MRI slices.** *MRI sections of 2D series of 22 slices (last slice not visualized). Images obtained from [13, 54].*

This chapter has presented the motivation and preliminary concepts which are used in image segmentation. It presents the segmentation idea and the objectives of image segmentation. It gives different types of image segmentation as well as outlining different applications of image segmentation. It focuses on medical images by explaining microscopic and magnetic resonance images from two different imaging technologies. These are the two types of images which have been segmented in the papers contributing this thesis. The next chapter presents the concepts of digital image processing in which image segmentation is a step.

Chapter 2

Digital image processing

Digital image processing is the computerized processing of images [39] with a target of extracting specific information. It is the process of manipulating digital images by using digital computers. During the processing of the image, noise information is removed because it would affect the analysis.

Image analysis and computer vision are the related areas that are considered to follow after image processing but there is no clear-cut or general agreement regarding where image processing stops and others start [33]. Different from image analysis, image processing outputs images whereas image analysis outputs attributes of the input image. Image processing has many steps but in this section, we focus on defining a digital image and digital representation of the signal before restricting the discussion of the thesis to image segmentation. Figure 2.1 presents summarized different steps covered in image processing.

Figure 2.1: **Image processing steps.** *The steps in the figure are summarized based on the useful description for computerized processes given in [33].*

2.1 Digital images

Let A represent a digital image storing information digitized from a real-world problem. We define, a digital image A as an array of values such that a value at any location represents a mapping of continuous signal by a function into digitized signals [21, 33, 135]. It is a 2D say $M \times N$ image defined as

$$A = \left[A_{r,c}\right]_{M \times N} \text{ for } r = 0, 1, \ldots M-1; c = 0, 1, \ldots N-1, \tag{2.1}$$

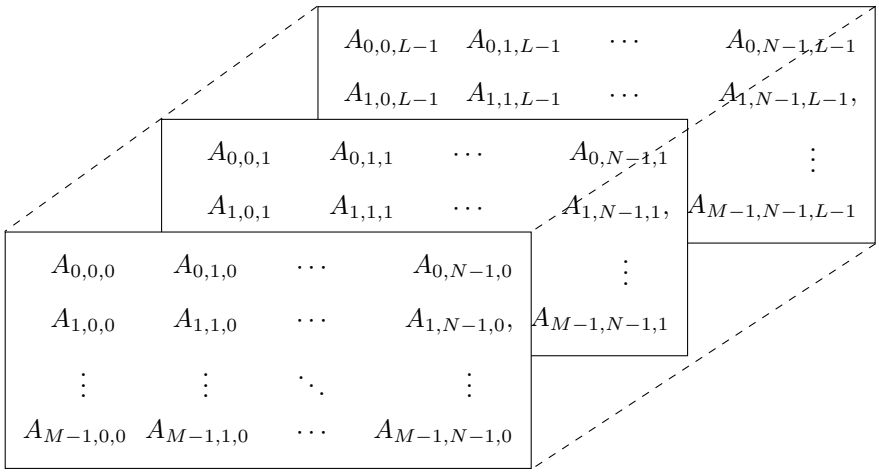
whereby r = row, c = column and each intensity value of an array A is accessed by its row and column numbers. In the expanded form it can be represented as

$$A = \begin{bmatrix} A_{0,0} & A_{0,1} & \cdots & A_{0,N-1} \\ A_{1,0} & A_{1,1} & \cdots & A_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{M-1,0} & A_{M-1,1} & \cdots & A_{M-1,N-1} \end{bmatrix}, \tag{2.2}$$

We also define a 3D representation of the digitized data say $M \times N \times L$ a sequence of stacked 2D image represented as

$$A = [A_{(r,c,h)}]_{M \times N \times L}. \tag{2.3}$$

such that $r = 0, 1, \ldots, M-1$; $c = 0, 1, \ldots, N-1$ and $h = 0, 1, \ldots, L-1$ and each point has a change in three directions r, c and h with respect to the position of the adjacent voxels. It is a union of voxels, implying the upright unit cubes whose vertices have integer coordinates [94]. In the expanded form the 3D can be represented as



The digital image comes from the continuous sensed data which is converted into digital form by digitizing the coordinate values (sampling) and then digitize the amplitude values (quantization) [33]. Generally, two important things for digital image are: (a) it is composed of digitized quantities (b) it is about the arrangement of these digitized quantities on a rectangular grid of pixels. The 2D may be stacked to form a 3D.

2.2 Operations on digital images

There are different operations that can be performed on a digital image which range from basic to complex based on the goal to be achieved. In this subsection we use definitions and symbols presented in [26].

Let $\mathcal{O}\{\}$ represents the various image processing operations applied to the digital input image. Let $G[x, y]$ be the output of input image $A[x, y]$ after applying operations. We define the general form for the operators to be

$$\mathcal{O}\{A[x, y]\} = G[x', y']. \tag{2.4}$$

Different literature classify different three operations that can be performed on digital images [26, 123, 135] but other literature give more than three types of operations.

2.2.1 Point, Local and Global operations

We first present three types of operations and thereafter other operation will be mentioned. The three types of operations are classified as follows

1. **Point operations:** This is the kind of operation whose output value at a specific coordinate depends only on the input value at that same coordinate. Example of points operations that can be done on single image include contrast stretching and histogram equalization [26]. The point operator can be represents as

$$G[x, y] = \mathcal{O}\{A[x, y]\}. \tag{2.5}$$

   For point operations, each pixel value of the input image $A[x, y]$ affects only the corresponding pixel value in the resulting output image $G[x, y]$.

2. **Local operations:** In this type of operation the output value at a specific coordinate depends on the input values in the neighborhood of that same coordinate. Examples of local operation include edge detection, smoothing [123]. Then local operator can be defined as

$$G[x, y] = \mathcal{O}\{A[x \pm \triangle x, y \pm \triangle y]\}. \tag{2.6}$$

3. **Global operations:** This is the type of operations in which the output value at a specific coordinate depends on all the values in the input image. Examples of global operation include image coordinate transformations [26]. If a pixel in the output image $G[x, y]$ is a function of (almost) all of the pixels in $A[x, y]$ the $\mathcal{O}\{A[x, y]\}$ is a global operator.

There are also other operations including geometric operations, shape-based operations [26]. From the shape-based operations, we define morphological techniques which are useful for both pre and post processing, such as filtering, thinning, and pruning [33].

We define morphology operations for binary image which are also useful for other images such as gray scale images. They highly depend on the ordering of the pixels instead of their numerical values [114]. Morphological operations highly depend on

the structure element [1] to perform the intended operation. Figures 2.2, 2.3, 2.4 and 2.5 give examples showing the effect of erosion, dilation, opening and closing respectively.

2.2.2 Erosion and dilation

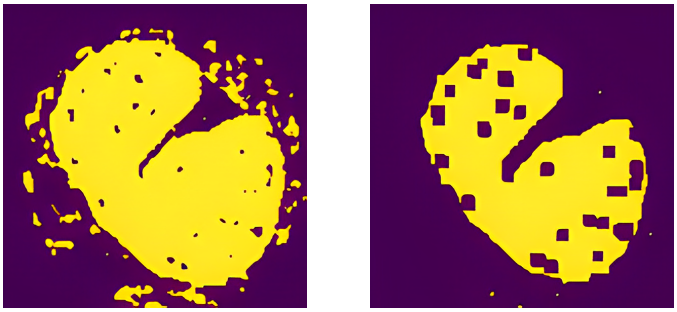
Erosion and dilation are fundamental operations for morphological processes. We defined A to represent both 2D and 3D images. Let Q be a binary image, and B be a structuring element.

Erosion

This is a morphological operation which uses a structuring element for probing and reducing the shapes contained in the input image. Then, erosion of a binary image Q by a structuring element B is mathematically defined as

$$Q \ominus B = \{z|(B)_z \subseteq Q\}. \tag{2.7}$$

where z's are foreground values (1's) and $\ominus$ is considered to be the Minkowski subtraction in [80, 103]. Equation (2.7) shows that erosion of Q by B is the set of all points z such that, B translated by z is contained in Q and the displacement is defined with respect to the origin of B [33].



(a) Input binary image (b) Eroded image

Figure 2.2: **Example showing effect of erosion.** *(a) input binary image of* 150×150 *size, (b) image eroded by a square structuring element of* 5×5 *size.*

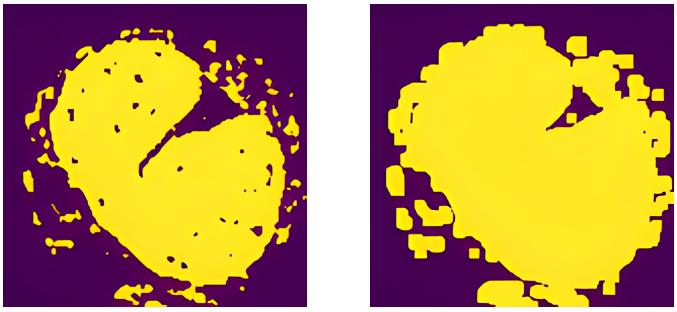
Dilation

This is also a morphological operation that uses a structuring element to expand shapes in the input image. The dilation of a binary image Q by a structuring element B is defined by

$$Q \oplus B = \{z|(\hat{B})_z \cap Q = \emptyset\}. \tag{2.8}$$

[1] Is a binary small probe with a predefined shape. It is applied to the input binary image to get the intended result.

where $\oplus$ is considered to be the Minkowski addition in [80, 103], $\hat{B}$ represents a reflection of a set of structuring element B about its origin [33]. Note that results for erosion and dilation presented in [103] are valid in both Euclidean space $\mathbb{R}$ and grid $\mathbb{Z}$, in n dimensions ($n = 1, 2, 3, ...,$) or more.



(a) Input binary image (b) Dilated image

Figure 2.3: **Example showing effect of dilation.** *(a) input binary image of* 150×150 *size, (b) image dilated by a square structuring element of* 5×5 *size.*

2.2.3 Opening and closing

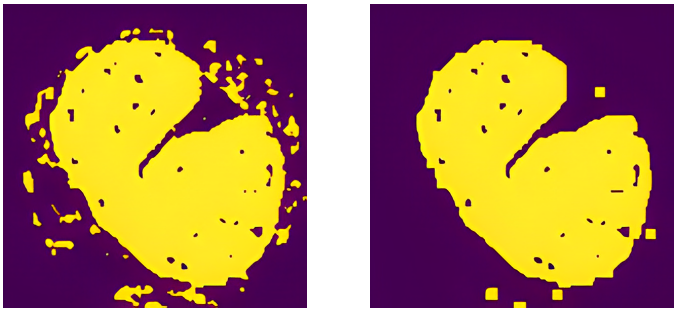
Opening and closing are operation defined based on the fundamental operations, that is erosion and dilation.

Opening

This is a combination of two processes which starts with erosion followed by dilation. That is, opening a hole of an object in a binary image Q by a structuring element B is mathematically defined by

$$Q \circ B = (Q \ominus B) \oplus B. \tag{2.9}$$

If the Q remains unchanged after opening with the structuring element B, then it is open with respect to B.



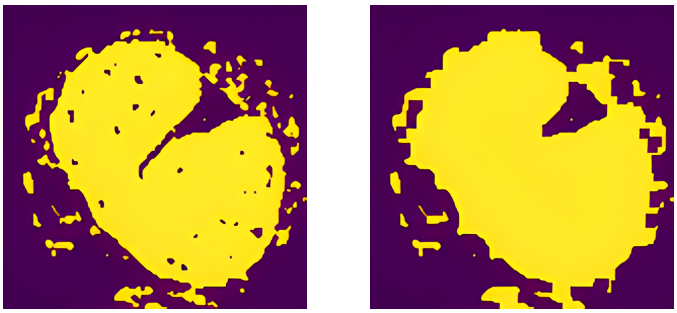
(a) Input binary image (b) Opening image

Figure 2.4: **Example showing the opening effect.** *(a) input binary image of* 150×150 *size, (b) image opened by a square structuring element of* 5×5 *size.*

Closing

This process is also a combination of two processes which starts with dilation followed by erosion. That is, closing an object in a binary image Q by a structuring element B is mathematically defined by

$$Q \bullet B = (Q \oplus B) \ominus B. \tag{2.10}$$



(a) Input binary image (b) Closing holes in image

Figure 2.5: **Example showing the closing effect.** *(a) input binary image of* 150×150 *size, (b) holes in image have been closed by a square structuring element of* 5×5 *size.*

If the Q remains unchanged after closing with the structuring element B, then it is closed with respect to B.

It should be noted that the purpose of erosion is to narrow and disconnect some parts in case we want to extract certain objects in the image and try to recover the loss of information by dilation. But the target is hardly achievable because all

of the morphological transformations are non-reversible, (except, in each particular case, for some sub-classes of sets such as the invariant ones) [103]. In this thesis, the morphological operations have been mostly used for post-processing, especially in Paper A and Paper C.

2.2.4 Image Filtering

Images acquired from different imaging systems are presented with noise and artifacts. The noises are random variations that occur in the intensity values of an image and they obscure the true signals. Examples of the popular noise found in the images are salt and pepper noise, impulse noise and Gaussian noise [33, 46].

Image filtering is a technique used for modifying and reducing the unwanted signals in the images [33]. These are operations in which algorithms are applied to the pixels of the input image to output enhanced images. We define convolution which is important in image processing. For simplicity, it is defined based on 2D images.

Suppose a 2D image I is to be convolved by a 2D filter g then the convolution operation between I and g is presented as

$$h[i,j] = I[i,j] * g[i,j] = \sum_{k=1}^{n}\sum_{l=1}^{m} I[i,j] \cdot g[i-k, j-l]. \tag{2.11}$$

Where i, j are spatial variables. The symbols $*$ and $\cdot$ are the convolution and pairwise product operation between I and g for obtaining the weighted sum of the pixel neighborhood about $[i,j]$.

2.2.4.1 Examples of Filters

Different techniques are used for filtering images. Some of the filters applied are mean filters, median filters, Gaussian filters, and derivative filters. We describe mean, median, and Gaussian filters.

Mean filter: It is a linear filter which works by local averaging operation and the value of each pixel is replaced by the average of all the values in the local neighborhood.

$$h[i,j] = \frac{1}{M}\sum_{k=i-1}^{i+1}\sum_{l=j-1}^{j+1} I[k,l]. \tag{2.12}$$

where M is the total number of pixels in the neighborhood. Comparing equation (2.11) and (2.12), if $g[i,j] = \frac{1}{M}$ for every $[i,j]$ in the convolutions mask, the convolution operation in equation (2.11) reduces to the local averaging operation [46]. Figure 2.6 shows the illustration of mean filter.

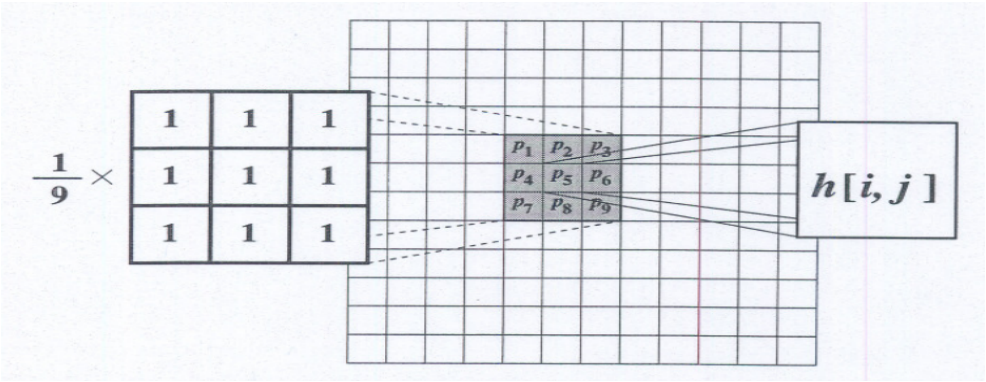


Figure 2.6: An example illustrating the mean filter using a 3×3 neighborhood. Figure from [46].

Median filter: This is a non-linear filter in which a pixel value is replaced by the median value of the gray-level neighboring pixels. It reduces noise in the image and the edges are maintained relatively sharper. They are effective at reducing the salt and pepper and impulse noise while retaining image details [26, 46].

The median filters work in successive image windows similar to linear filters. However, in a window, a median of the pixels in each window centered at $[i, j]$ is computed [46]. See illustration in Figure 2.7.

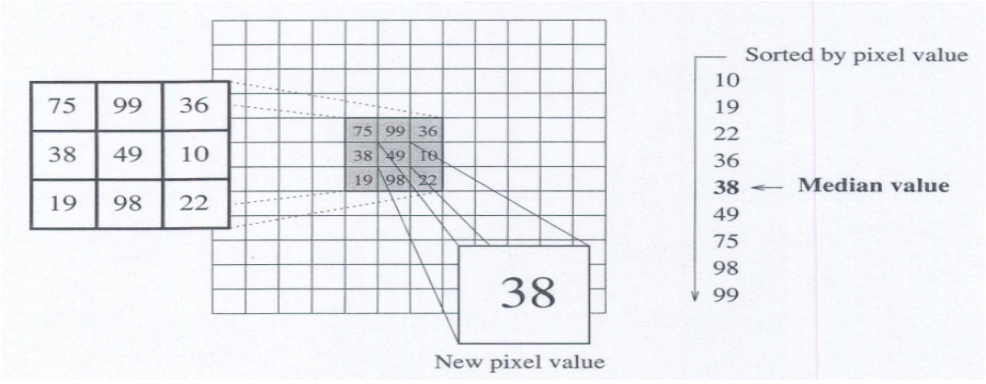


Figure 2.7: An example illustrating the median filter using a 3×3 neighborhood. Figure from [46].

Gaussian filters: This is a class of linear filters where weights are chosen according to the shape of a Gaussian function. The Gaussian filters are efficient for smoothing, noise reduction or blurring images [46].

For image processing, the two-dimensional zero-mean discrete Gaussian function

is given by,

$$G[i,j] = \frac{1}{2\pi\sigma^2} e^{\frac{-(i^2+j^2)}{2\sigma^2}}. \tag{2.13}$$

where σ is a standard deviation of the distribution and the distribution is assumed to have mean of 0 and σ =1. The 2D function of the Gaussian filter is illustrated in Figure 2.8.

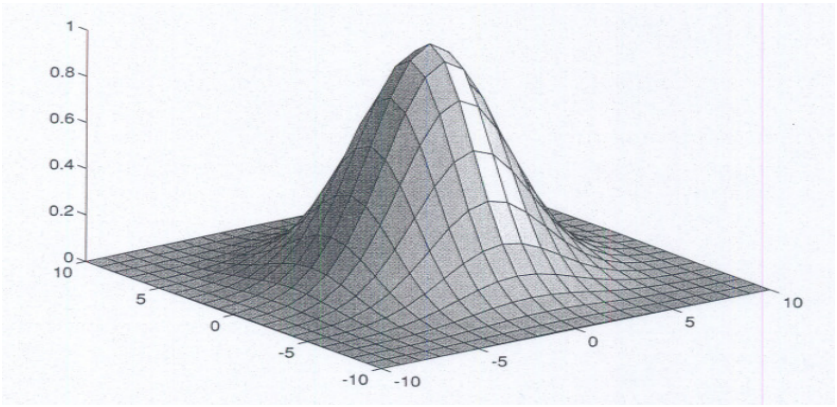


Figure 2.8: An example of a two-dimensional Gaussian function with zero mean. Figure from [46].

Practically, the Gaussian smoothing filters are effective because of their different properties. The Gaussian smoothing filters are unbiased for detecting edges in any direction because they are rotational symmetry. It is parameterized with σ which controls the degree of smoothing. The parameter is adjusted to get the desired smoothed image [33, 46].

The chapter has presented the concepts of digital image processing in which image segmentation is a step. It gives a mathematical representation of digital images for both 2D and 3D. It presents different operations that can be applied to digital images. It gives examples for each of the operations presented in this chapter. It also gives examples of the operations combined. The examples provided are based on binary images. The chapter ends by discussing image filtering. Examples of selected image filters are given describing how the convolving filters are applied to image filtering.

Chapter 3

Image segmentation Methods

Different approaches are used for image segmentation. They can be classified based on their formulation and how they perform the segmentation process. The use of image segmentation in different fields of application has led to an increase in newly developed methods for segmenting images. Different methods which are used for image segmentation include those based on thresholding, edge-based segmentation methods, boundary-based, clustering-based, model-based segmentation, hybrid methods, machine learning approaches, deep learning [41]. Other segmentation techniques include region growing, region splitting, region merging, detection of boundary discontinuities, watershed segmentation, active contours [33, 39] and graph-based image segmentation [27, 64, 72, 98, 109, 110]. Based on the formulation of different methods the region to be segmented in the image can be defined either by its interior or by the edges of the region, and any of the two representations can be used.

3.1 Threshold based methods

Thresholding is a segmentation method that assigns the same label to pixel values confined in a certain range. The pixel values considered to be the object of interest are assigned the same label and those considered to be the background are assigned a different label. It categorizes pixels based on the range of values in which a pixel lies. These methods are popular and widely used compared to other methods because they are simple to use. They are categorized into either global or local thresholding [57, 111]. Also, thresholding can be applied on an image in a multi-level style [60].

3.1.1 Global thresholding methods

The global thresholding methods are simple and suitable for binarizing grayscale images, especially for images whose foreground are well-separated from the background [28, 92]. It is an approach which uses a single threshold value to perform

segmentation for the entire image [33, 84]. If $A(x, y)$ is the image to be segmented using a global threshold, mathematically the segmented image A_{seg} is represented as

$$A_{seg} = \begin{cases} 1 \text{ if } A(x,y) \geq Threshold \\ 0 \text{ if } A(x,y) < Threshold \end{cases}$$

One of the famous and optimum global thresholding is Otsu technique [33]. It is based on discriminant analysis. It selects the threshold by maximizing the between-class variance [82].

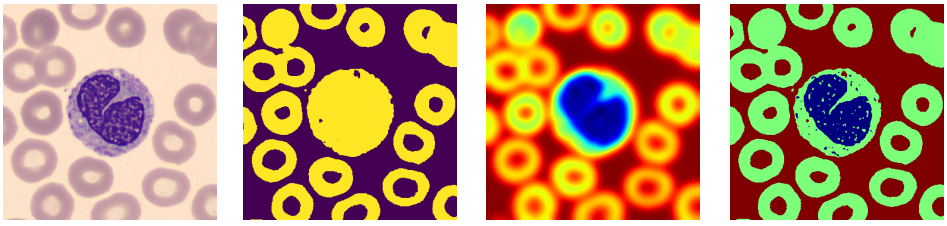
In practice, global thresholding depends on image histogram and the chances of finding a relatively good threshold is influenced by the presence of histogram peaks which are tall, narrow, symmetric, and separated by deep valleys [33]. These situations are practically hard to find in many images. It is challenging to use global thresholding if the histogram does not have clear separation of foreground and background features. It is also challenging if the image is noisy and the background intensity varies significantly across the image but the noisy image can be improved by smoothing prior to thresholding [33]. But also, thresholding does not consider spatial information, so it is difficult to guarantee that segmented regions are contiguous [120].

3.1.2 Local thresholding methods

Local or adaptive thresholding is used to overcome the complexity of binarizing an image whose illumination is not uniform. Different thresholds are estimated locally for each pixel based on the local feature in the image, which differs from its immediate neighbors [28]. It utilizes the local information of pixels in the image to estimate thresholds for different parts of the image because of the in-homogeneity of signals in the image. Examples of local thresholding methods are Niblack [78] and Sauvola [99] thresholding methods.

3.1.3 Multi-level thresholding methods

This is a process that segments a gray-level image into several distinct regions. This technique determines more than one threshold for the input image and then segments the image into several regions based on the classification of their intensity values [4, 60]. The method segments a background and several objects which are considered to be the foreground. An example of the popular method in this category is the extension of Otsu algorithm into multilevel thresholding [60]. Figure 3.1 presents examples of segmented image comparing global, local and multi-level thresholding.



(a) Input image (b) Global threshold (c) Local threshold (d) Multilevel threshold

Figure 3.1: **Comparison of global, local and multilevel thresholding.** *(a) Input image (b) segmented image using global thresholding (c) segmented image using local thresholding (d) segmented image using multi-level thresholding.*

3.2 Threshold estimation

Manual selection of thresholds to segment regions of interest is one way but an automatic estimation of thresholds is mostly preferred because the manual selection is time consuming. Depending on the complexity of the image and how the foreground is well separated from the background, both manual and automatic selection of the threshold are applied.

3.2.1 Manual selection

This approach of obtaining thresholds is usually based on trial and error. The user selects values manually and tests them until the user is satisfied with the results. It is based on the visual judgment of the user and then decides to stop or continue testing. The approach is mostly assisted by first constructing an image histogram to see the possible first guess and then start testing different values to get the optimal value.

3.2.2 Automatic estimation

This is an automatic method of estimating thresholds for segmenting images. The estimation techniques can differ based on the formulation of the algorithm. Some techniques require an initial value either selected manually or automatically to estimate the optimal threshold based on the initialized value. Different methods which estimate thresholds automatically include Huangs fuzzy thresholding method [43], Yens thresholding method [104, 131], triangle thresholding [136], percentile threshold [25], moments thresholding [118], minimum error thresholding [52], MaxEntropy thresholding [50], Li thresholding [58, 59]. Below we present different thresholding methods which estimate the threshold automatically.

3.2.2.1 Examples of automatic methods

Intermodes

The method is applied to images which are assumed to have a bimodal histogram. The histogram of an image is iteratively smoothed until there are only two local maxima x_j and x_k. The threshold is computed by

$$\text{Threshold} = \frac{x_j + x_k}{2}. \tag{3.1}$$

The method is not suitable for images with histograms having extremely unequal peaks or a broad and flat valley [86].

IsoData

The method divides the image into object and background by taking initial threshold. The average of the pixels values below the threshold is computed. Also, the average of the pixel above the threshold is compute. Then, the method computes the average of the two values. The threshold is incremented and the process is repeated until the threshold is larger than the composite average [91].

$$\text{Threshold} = \frac{\text{Average background} + \text{Average object}}{2}. \tag{3.2}$$

Mean threshold

This approach computes the mean of the gray level and uses the computed value as the threshold [31]. For method which require initialization values, the mean values has been mostly used as the initial guess threshold.

$$\text{Threshold} = \frac{\text{sum of gray level values}}{\text{number of gray level values}}. \tag{3.3}$$

Otsu threshold

The method proposed by N. Otsu [82] searches for the threshold that minimizes the within-class (intra-class) variance. The method computes a histogram of the image which expresses two peaks. The two peaks represent different groups or range of intensity values. The method aims at separating the computed image histogram into two segments with a threshold value. The threshold is defined as a result of minimizing the weighted variance of the classes [82].

$$\sigma_w^2(t) = w_0(t)\sigma_0^2(t) + w_1(t)\sigma_1^2(t), \tag{3.4}$$

$$w_0(t) = \sum_{i=0}^{t-1} p(i);\ \ w_1(t) = \sum_{i=t}^{L-1} p(i). \tag{3.5}$$

where w_0 and w_1 are the probabilities of the two classes separated by the threshold t. σ_0^2 and σ_1^2 are the variances of the two classes. L represents bins of the histogram. $\sigma_w^2(t)$ is the threshold which is defined as a result of minimization of the weighted variance of these classes.
Also, the method can be achieved by another option which requires to maximize the between-class (inta-class) variance using the expression

$$\sigma_b^2(t) = w_0(t)w_1(t)[\mu_0(t) - \mu_1(t)]^2, \tag{3.6}$$

where $\mu_i(t)$ is the mean of class i

$$\mu_0(t) = \sum_{i=0}^{t-1} ip(i);\ \mu_1(t) = \sum_{i=t}^{L-1} ip(i). \tag{3.7}$$

The probability of gray-level i is computed by

$$p(i) = \frac{N_i}{N}. \tag{3.8}$$

N is the number of pixels in the image and N_i is number of occurrence of gray-level i. The optimum threshold is the value t^* that maximizes $\sigma_b^2(t)$.

$$\sigma_b^2(t^*) = \max_{0<t<L-1} \sigma_b^2(t). \tag{3.9}$$

The algorithm finds the final threshold t^* value given by the maximum $\sigma_b^2(t)$ value to perform a global thresholding of an image.

For multiple K classes the problem has been extended and used to produce *multiple thresholds* [60]. Then, equation (3.9) can be rewritten as follows

$$\sigma_b^2(t_1^*, t_2^*, \ldots, t_K^*) = \max_{0<t_1<t_2<\cdots<t_K<L-1} \sigma_b^2(t_1, t_2, \ldots, t_K). \tag{3.10}$$

Asumming that the image to be segmented has intensity values that can be segmented into three classes then two thresholds t_1^* and t_2^* are needed.

$$\sigma_b^2(t_1^*, t_2^*) = \max_{0<t_1<t_2<L-1} \sigma_b^2(t_1, t_2). \tag{3.11}$$

The image is segmented using multiple thresholds and represented as

$$A_{seg} = \begin{cases} a \text{ if } A(x,y) \leq t_1^* \\ b \text{ if } t_1^* < A(x,y) \leq t_2^* \\ c \text{ if } A(x,y) > t_2^* \end{cases}$$

where a, b and c are three distinct intensity values and A_{seg} is the segmented image. But as the number of classes increases the method fails because multiple global thresholding is a viable approach when we are sure about the number of classes in the image.

3.2.2.2 Threshold estimation based on local minima

This is the group of methods that are considered to find threshold by searching for the local minima which separate objects which are identified as local maxima in the histogram of the image. In most cases, the histogram is assumed to be bimodal after undergoing some processing.

The method presented in the paper A [71] is also based on the histogram but ignores the possibility of bimodal existence in the histogram of the image. It searches for a threshold that segments a targeted group of information. It was developed based on local minima for segmenting the nucleus of the WBCs but it can also be used on related images. The steps in the method can be divided as follows:

1. Automatic initialization of a value computed from the input image. The initial value T_{nc0} is computed by

$$T_{nc0}(n_1, n_2) = \frac{Max(A)}{n_1} + \frac{Min(A)}{n_2}. \tag{3.12}$$

   Where n_1 and n_2 are integers greater than 0. A simple analysis on how to choose n_1 and n_2 is presented in the supplementary information for paper A.

2. Then, the 2D input image A is vectorized, and then a non-decreasing sequence of the intensity values is constructed and represented by $\tilde{A}_{seq}$

$$\tilde{A}_{seq} = \left[I_i \leq I_{i+1} \leq \cdots \leq I_{k-2} \leq I_{k-1}\right] = \left[I_i \ldots, T_{nc0}, \ldots, I_{k-1}\right]. \tag{3.13}$$

   where I_i represent the intensity values in the input image. T_{nc0} is not necessarily one of the values in $\tilde{A}_{seq}$ but it is within the range of $\tilde{A}_{seq}$.

3. Then, the threshold $\tilde{\varepsilon}_t$ is computed by checking the following conditions:

   (a) $x_c < T_{nc0}$.

   (b) If $|T_{nc0} - x_{c+1}| < Er$ then, $\tilde{\varepsilon}_t = \dfrac{T_{nc0} + x_{c+1}}{2}$.

   (c) If $|T_{nc0} - x_{c+1}| > Er$ and $T_{nc0} > x_{c+1}$ then, $\tilde{\varepsilon}_t = \dfrac{T_{nc0} + x_{c+1} + Er}{2}$.

   (d) If $|T_{nc0} - x_{c+1}| > Er$ and $T_{nc0} < x_{c+1}$ then, $\tilde{\varepsilon}_t = \dfrac{T_{nc0} + x_{c+1} - Er}{2}$.

   where x_c and x_{c+1} are values obtained from first local maximum $(x_c, g(x_c))$, and first local minimum $(x_{c+1}, g(x_{c+1}))$ of the function $g(x)$ respectively. $g(x)$ is the spline of points from the approximated histogram of the input image. Er is a parameter introduced to control T_{cn0} and the local minimum value during threshold estimation.

4. The threshold $\tilde{\varepsilon}_t$ is applied on the input image A to produce image A_{recons} and it is mathematically obtained by

$$A_{recons} = \begin{cases} Min(A) \text{ if } A(x,y) \leq \tilde{\varepsilon}_t \\ A(x,y) \text{ elsewhere} \end{cases}$$

The image A_{recons} is binarized to segment the nucleus of the WBCs. Different from other thresholding methods, the method uses the estimated threshold to reconstruct the input image to make it simple for segmentation. Then, post-processing operations are applied to the binarized image to obtain the final segmented nucleus. For more information see paper A [71].

Examples comparing different thresholding methods

In this subsection we provide examples comparing different segmentation methods based on thresholding. Different methods are tested on the same image to show their performance on segmenting the image.

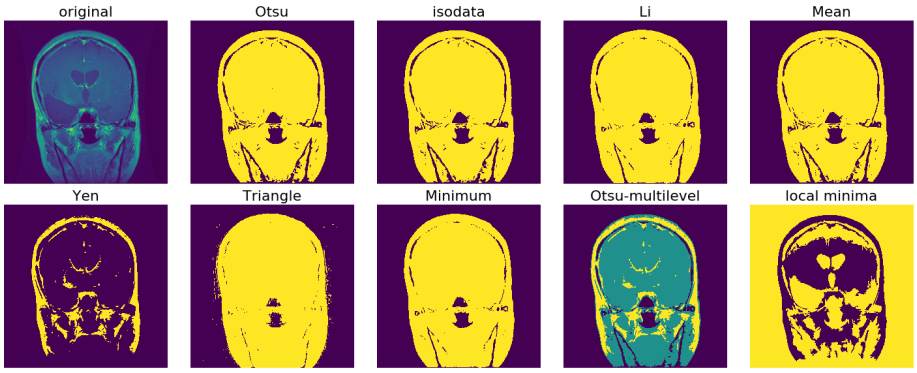


Figure 3.2: **Comparison of thresholding methods.** *Different methods applied on representative MRI coronal section with a tumor.*

Looking at all the results based on different methods in Figure 3.2, if the ROI is the tumor, then the results that can be post-processed to segment the tumor are the results obtained using the local minima thresholding method. See Figure 3.3.

Figure 3.3: **Threshold estimated using the local minima approach to segment the tumor.** *Flow showing the tumor segmentation steps based local minima approach in [71]: (a) tumor pointed by an arrow in the image, (b) binarized image using the threshold estimated automatically, (c) opening to detach the tumor from other tissues, (d) assigning different labels to the connected components, (e) final segmented tumor.*

Figure 3.4 gives another example comparing the results of segmented nucleus and other objects from image of WBC. The results that can be post-processed to give the nucleus are those obtained by isodata, Yen, minimum, Otsu multilevel, and local minima thresholding methods.

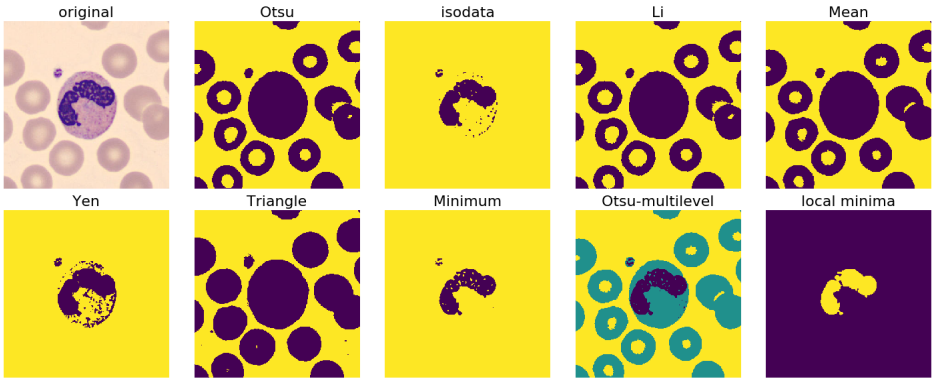


Figure 3.4: **Nucleus segmentation of WBC using different thresholding methods.** *Different methods applied on the image with RBCs and WBC to segment the nucleus by thresholding methods. For some of the methods the results can be post-processed to segment the nucleus.*

3.3 Segmentation method based on deep learning

3.3.1 Neural Network

Neural Networks (sometimes known as artificial neural networks) are artificial networks inspired by the behavior of biological neurons. They are computational models which mimic the neurons in human brains and are used for recognizing patterns in the datasets [77]. Figure 3.5 shows the inspiration for the artificial neuron from the biological neuron.

The neural networks have their functional form of the model known to be its architecture. Sometimes the model and architecture are synonyms. These models have weights which are called parameters and the results of the model are called predictions [40].

3.3.1.1 Components of Neural of Network

Neural Networks consist of the following components: an input layer, an output layer, and a hidden layer(s). The input layer is at the beginning of the network and brings the initial data into the system for further processing. A hidden layer is an intermediate between the input and output layers. It receives data from the input layer and processes it to produce output for the next layer. The output layer takes in the processed data as input to produce final results [35, 89].

3.3.1.2 Types of Neural Network

There are different types of Neural networks. They can be classified into different categories based on their: Structure, Layers and depth, data flow, neurons, and dense or non-dense. These include Perceptron, Feedforward neural networks, Multi-layer perceptrons (MLPs), Convolutional neural networks (CNNs), and Recurrent neural networks (RNNs) [35]. The content in this subsection will focus on CNN model because is mostly applied to image processing.

3.3.2 Deep learning

Deep learning is a class or subfield of machine learning (ML) algorithms that use multiple layers to progressively extract higher-level features from the input [20, 83]. Deep learning algorithms have more than one hidden layer.

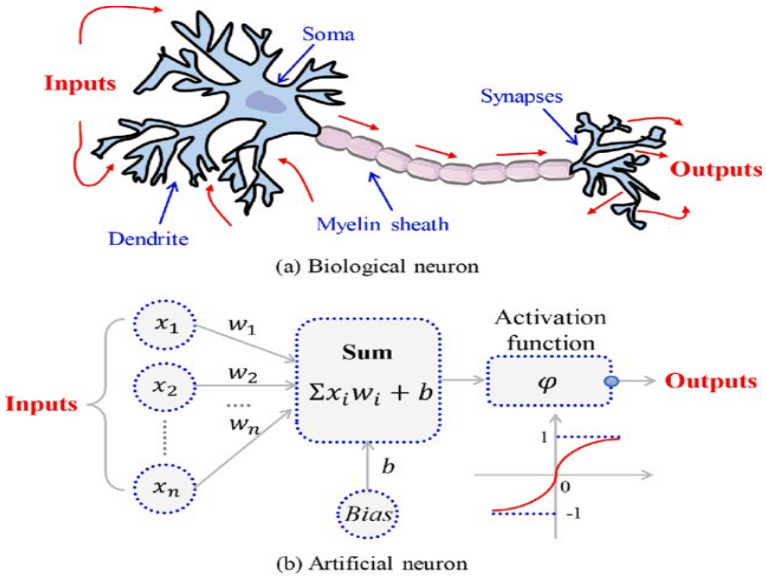


Figure 3.5: (a) Biological neuron and (b) Artificial neuron. A comparison showing the inspiration of the Artificial neuron from biological neuron. Figure from [121].

3.3.3 Convolutional Neural Network (CNN)

CNN are neural networks often applied to image processing problems. In the CNN, layers are not densely connected, and the weights are shared. They are primarily used for classification, segmentation, and object recognition [81].

These models utilize sets of filters for learning different feature maps from the input images and then forward to the subsequent layer in the model. Considering a

set of K filters, the feature map generated in the n^{th} layer can be represented as:

$$X^n = W_k^n * Y^{n-1} + b_k^n, k = \{1, 2, \dots K\}. \tag{3.14}$$

where K is the set of filters, W_k^n is the k^{th} filter in the n^{th} layer, b_k^n is the k^{th} bias in the n^{th} layer. Y^{n-1} is the feature map from $(n-1)^n$ layer and the symbols $*$ is the convolution operation between W_k^n and Y^{n-1}, refer equation (2.11).

Then, an activation function $f(\cdot)$ is applied on X^n to generate feature maps as follows:

$$Y^n = f(X^n). \tag{3.15}$$

Equations (3.14) and (3.15) are updated to produce inputs of the subsequent layers. The idea governing the operation in equation (3.14) is to learn the filter weights so as to extract feature maps and the filter's weights are spatially shared.

For CNN, the inputs are the image data sets. Depending on the dimension of the input images, the CNN model applies convolution to capture the features found in the image data set. Assuming that the dimension of the input Y^{n-1} for generating feature maps in n^{th} layer is $w \times h \times n_c$, where w, h and n_c are the width, height and number of channels of Y^{n-1} respectively. For example, a color image normally has red, green, and blue channels. This implies that number of channels n_c for the color image is three. It is the depth involved in the convolutions. Generally, inputs in CNN models are tensors with shape (number of input)$\times$ (input width)$\times$ (input height)$\times$ (input channels). The tensors are multi-dimensional arrays with a uniform data type.

3.3.4 CNN architecture for image segmentation

The CNN architectures can have slight variations based on the task to be performed. For classification, the CNN models have the convolution layer, pooling layer, and fully connected dense layer with some other modules which are used to improve the learning process [81].

For image segmentation, CNN uses convolution, pooling, and up-sampling layers. In the CNN model for segmentation, the fully connected layers are removed to retain the spatial relationship of pixels throughout the network. Figures 3.6 and 3.7 present examples of 2D and 3D CNN architectures respectively. As seen in Figures 3.6 and 3.7, the CNN models have encoding and decoding paths. In the encoding path, abstract representations of the input image are produced. The network maps the input image pixels to get a representation of a collection of feature vectors. In the decoding path, the abstract image representations are up-sampled to make their spatial dimensions equal to the input image. The decoding path takes these features to produce output maps. There are Skip connections which are introduced to help recover spatial information of input images.

Example of a 2D CNN architecture

2D CNN applies a 2D filter to the image and the filter moves 2-direction (x, y). Their output shape is a 2D space.

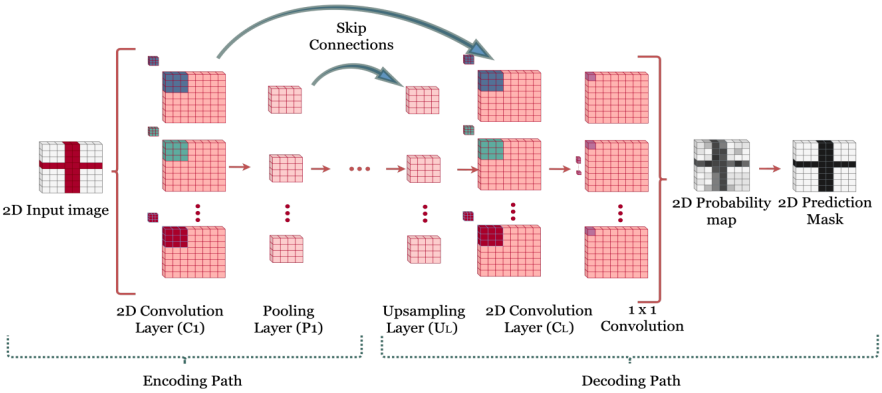


Figure 3.6: A 2D CNN architecture for image segmentation. Figure from [81].

Example of a 3D CNN architecture

3D convolutions applies a 3D filter to the dataset and the filter moves 3-direction (x, y, z). Their output shape is a 3D volume space.

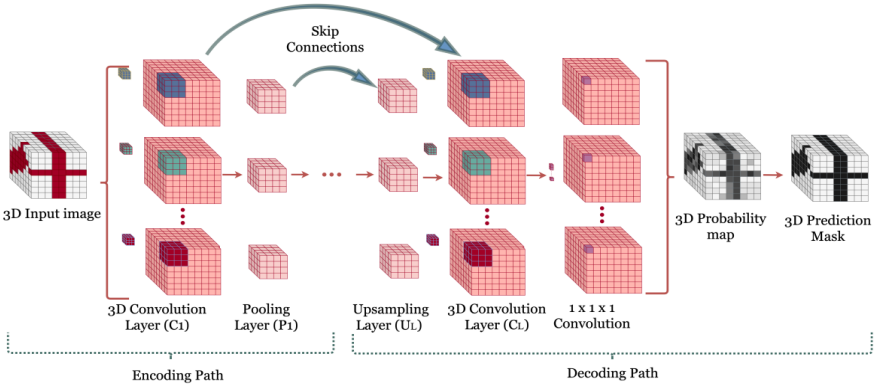


Figure 3.7: A 3D CNN architecture for image segmentation. Figure from [81].

3.3.4.1 Convolution layer

The convolutional layer is an essential building block of a CNN. It is in this layer where most of the computation occurs. In this layer, the feature detector (filter) convolves across the input images for the features present. The learnable filter slide over the input to detect patterns that are used as new feature maps from the previous

layer and used as input in the subsequent layer. These operations are performed to obtain the correlation between their weights. The dimension of the filters depends on the dimension of the input images.

3.3.4.2 Pooling layers

Pooling layers are used to reduce the dimensions of the feature maps. Thus, it reduces the number of parameters to learn and the amount of computation performed in the network. They are used to consolidate the features learned by convolution neural networks. There are mainly two types of pooling opearations which include Maximum and average pooling. For maximum pooling a largest value for each patch is calculated to generate feature maps whereas an average value is calculated in average pooling. Also, there is a global pooling which is a variant of maximum and average pooling. Figure 3.8 shows the maximum pooling and up-sampling.

3.3.4.3 Up-sampling layers

These are the layers used for increasing the dimension of the input from the pooling layer. Note that in the pooling layer, the input is down-sampled to reduce the computation. So, the up-sampling layers are added to compensate for the down-sampling which happened in the pooling operation [81]. It is performed to increase the number of rows and/or columns of the down-sampled images. For example, for 2D dimension images the up-sampling is performed in both the x and y directions to fill in gaps rows-wise and column-wise by a certain technique to get the required dimension. The step tends to recover some of the lost information in the pooling layer [133].

Different techniques are used for up-sampling images. These include methods that are based on interpolation such as Nearest Neighbour, Bilinear, Bicubic Spline, and Generalized Bicubic Interpolation. Also, some simple operations are applied whereby zeros are added in between pixels of the input map to get the required dimension. Figure 3.8 compares the maximum pooling and up-sampling.

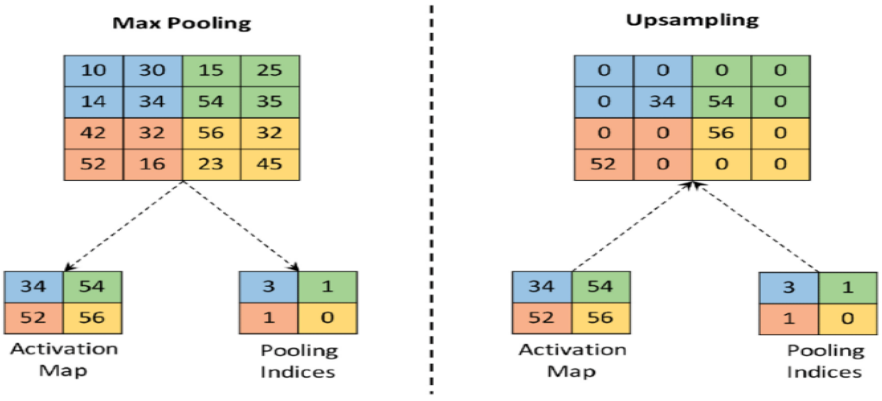


Figure 3.8: Figure showing maximum pooling and up-sampling. Figure from [24].

The up-sampling performed in the Figure 3.8 is a simple operation that fills in gaps by zeros but the pooled values are taken back to their original positions.

3.3.4.4 Loss functions

This is a measure of an error between the actual and the predicted values. These functions are used for evaluating how well the algorithm is modeling the data set. That is quantifying how good or bad the model is performing to know the accuracy of the model's predictions. In a neural network, the loss function (which is an objective function of the model) is minimized [45].

There are different types of Loss functions such as Regression Loss Functions and classification loss functions. Since image segmentation is the classification task on the pixel level the loss function for this task falls in the category of classification loss functions. Examples of loss functions for segmentation include Binary Cross-Entropy, Weighted Binary Cross-Entropy, and Balanced Cross-Entropy [45].
The Binary Cross-Entropy is defined as:

$$L(y, \hat{y}) = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})). \tag{3.16}$$

where y and $\hat{y}$ are the actual and predicted values of the data respectively.
The Weighted Binary Cross-Entropy which is a variant of Binary Cross-Entropy is defined as:

$$L(y, \hat{y}) = -(\beta * y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})). \tag{3.17}$$

Where β is the value that can be used for tuning false negatives and false positives. For example, when $\beta > 1$ reduces the number of false negatives and when $\beta < 1$ decreases the number of false positives [45].
The Balanced Cross-Entropy is similar to Weighted Cross Entropy and is defined as:

$$L(y, \hat{y}) = -(\beta * y \log(\hat{y}) + (1 - \beta) * (1 - y) \log(1 - \hat{y})). \tag{3.18}$$

For this case β is the number of negative samples divided by total number of samples. That is the fraction of the sample which is dominant in a data set.

3.3.4.5 Optimization functions

These are methods that are used for adjusting the model's parameter to minimize the loss function. They are optimization functions that improve the performance of the model. In each epoch, the weights are modified to minimize the loss function.

There are different optimizers such as Gradient Descent, Stochastic Gradient Descent (SGD), Root Mean Square Propagation (RMS Prop), and Adaptive Moment Estimation (Adam).

Gradient Descent (GD)

This is an optimization algorithm that iteratively computes the parameter for minimizing the loss function. The algorithm works well for differentiable and convex functions.

Assuming that $F(\theta)$ is the loss function with respect to the training data set and θ is the parameter vector. We want to solve a minimization problem of the loss function F. The general form of GD is written as

$$\theta_{t+1} = \theta_t - \alpha_t \nabla F(\theta_t). \tag{3.19}$$

where α_t is the learning rate (step size) which may remain fixed or vary with t and ∇F is the gradient of the loss function for the t^{th} iterate . The parameters are successively refined until when the stopping criteria is satisfied.

The gradient descent algorithm is guaranteed to find the global minimum whenever the function is convex. For non-convex functions, the GD algorithm struggles to find the global minimum because sometimes the GD gets stuck in local minima or saddle points.

Since the convergence speed of GD depends on the learning rate, then smaller learning rates will require more training epochs because only smaller changes will happen to the weights for each update. The larger learning rate requires fewer training epochs because of the rapid changes which happen during the weight updates [10, 117].

Note that a model trained using a learning rate that is too large causes a convergence that is too quick most likely to a solution that is sub optimal. Also, a model trained using a learning rate that is too small can lead the process to be stuck [10, 117].

Stochastic Gradient Descent (SGD)

This is a modification of gradient descent. In each iteration of the stochastic gradient descent, a gradient is calculated using a small part of the observations instead of all of them.

Given a data set of n samples. Assuming that $F_i(\theta)$ is the loss function for the training sample of index i, and θ is the parameter vector. Suppose a sample at the index $i \in 1, 2, 3, \ldots, n$ for the sample is randomly sampled. Theoretically, SGD works by picking i randomly at t^{th} iteration and update the parameter by

$$\theta_{t+1} = \theta_t - \alpha_t \nabla F_i(\theta_t). \tag{3.20}$$

where α_t is the learning rate (step size) and $\nabla F_i(\theta)$ is the gradient of the objective function computed to update the θ of sample i .

Mostly, in the implementation the set of all samples are randomly shuffled and then split into different mini-batches which are used for computing gradient and weight update [117].

Stochastic Gradient Descent With Momentum

Stochastic Gradient Descent with momentum is an optimization method that helps accelerate gradient vectors and leads to faster converging. The Stochastic Gradient Descent with momentum works by randomly picking i at the t^{th} iteration and then update the momentum term and the parameter by using

$$m_t = \beta m_{t-1} + (1-\beta)\nabla F_i(\theta_t). \tag{3.21}$$

where m_t is first moment estimates at time t, β is a decay rate. In the implementation, m_0 is initialized as a first-moment vector. $\nabla F_i(\theta_t)$ is the gradient of the objective function with respect to θ evaluated at time step t. The parameters (weights) are updated based on the equation below

$$\theta_{t+1} = \theta_t + \alpha_t m_t. \tag{3.22}$$

where α_t is a learning rate or step size.

For convex functions, the possible ways that can improve the convergence rate is first by exploiting tricks like variance reduction and considering more structure of the problem. For non-convex functions, the problem is quite hard and the methods proposed for improving the convergence rate for non-convex functions are mainly theoretical [117].

Adaptive gradient methods

This is a class of optimization methods popularly known as adaptive gradient methods. Examples of these methods include Root Mean Square Propagation (RMS Prop) and Adaptive Moment Estimation (Adam).
For RMS Prop, i is picked randomly at the t^{th} iteration, compute $g_t = \nabla F_i(\theta_t)$, and then update the second order momentum v_t and then update the parameter by

$$v_t = \beta v_{t-1} + (1-\beta) g_t \circ g_t. \tag{3.23}$$

$$\theta_{t+1} = \theta_t - \alpha_t v_t^{-1/2} \circ g_t. \tag{3.24}$$

Note that $\circ$ denotes entry-wise product.
Adam method is obtained after combining RMS Prop and the momentum method. At the t^{th} iteration of RMS Prop, $g_t = \nabla F_i(\theta_t)$ is computed after randomly picking i then followed by updating the first order momentum m_t, the second order momentum v_t and the parameter θ_t by

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \text{ and } v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t \circ g_t. \tag{3.25}$$

where m_t and v_t are first and second moment estimates at time t respectively. β_1 and β_2 are decay rates. m_0 and v_0 are initialized as first and second moment vectors respectively. g_t is the gradient of the objective function with respect to θ evaluated at timestep t. The parameters (weights) are updated based on equation below

$$\theta_{t+1} = \theta_t - \alpha_t v_t^{-1/2} \circ m_t. \tag{3.26}$$

Adam is one of the optimization methods which are popular for neural network. The advantage of Adam over SGD is being relatively insensitive to hyperparameters but some reasearches show that the well-tuned SGD and SGD with momentum do perform better compared to Adam in both training error and test error [117, 125].

3.3.4.6 Activation Functions

These are functions that are used for transforming the input data into output signals in artificial neural networks. For neural networks, the most preferred activation functions are non-linear because they introduce non-linearity into the network. Examples of common activation function in deep learning include Sigmoid $f(x) = \frac{1}{1+e^{-x}}$, Tanh Hyperpolic $f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$, Rectified Linear Unit (ReLu) $f(x) = max(x, 0)$ and Softmax activation function $\sigma(\mathbf{z}_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$, K is the number of inputs in vector $\mathbf{z}$.

For binary classification and segmentation, the Sigmoid function is the preferred activation function. For classification or segmentation tasks with more than two classes of labels, the preferred activation function is Softmax [79]. It is worth noting that ReLU is also preferred over Sigmoid or Tanh because of its cheap arithmetic operation and excellent convergence properties on the stochastic gradient descent (SGD) algorithm [90].

3.3.4.7 Normalization

This is a technique for standardizing data to make them in the same range. In a deep neural network, it is the preprocessing layer that normalizes the features in the data by shifting and scaling the inputs. The output of the neurons gets normalized before being passed to the activation function [42].

There are different types of normalization in neural networks including batch normalization [44], Layer normalization [6], and group normalization [127].

3.3.4.8 Regularization

These are techniques used for calibrating models. They are introduced to minimize the adjusted loss function and prevent either overfitting or underfitting of the model. These methods are used for enhancing the generalization capability of the model and show the model's ability to perform well on unseen samples. They apply penalties on either layer parameters or layer activity during the optimization process [97].

There are different regularizers and they are named based on the tasks they perform in the model. These include kernel regularizer, bias regularizer, and activity regularizer. The kernel regularizer applies a penalty on the layer's weights. It tries to reduce the weights causing the network to overfit. The bias regularizer applies a penalty on the layer's bias to reduce the bias. The activity regularizer applies a penalty on the layer's output. The activity regularizer tries to reduce the layer's output which will reduce the weights and adjust the bias. The penalties applied on the layers or activity get summed into the loss function being optimized.

3.3.4.9 Hyperparameter tuning

Hyperparameters are the variables that determine the network structure and how the network is trained. They are useful in the process of estimating the model's parameters. The hyperparameters are explicitly defined by the user to control the learning process and optimizing the model. They can be grouped into model and algorithm hyperparameters.

Model hyperparameters: These are parameters related to Network structure that is considered to highly influence the model selection. Examples include the number and width of hidden layers.
Algorithm hyperparameters: These are parameters for the training the algorithm. They highly influence the training speed and quality of the learning algorithm.

3.3.4.10 Examples of CNN Models for medical image segmentation

There are different CNN models which have been developed for medical image segmentation. The popular CNN models are U-Net for biomedical image segmentation [93] and V-Net Model for Volumetric Medical Image Segmentation [74].

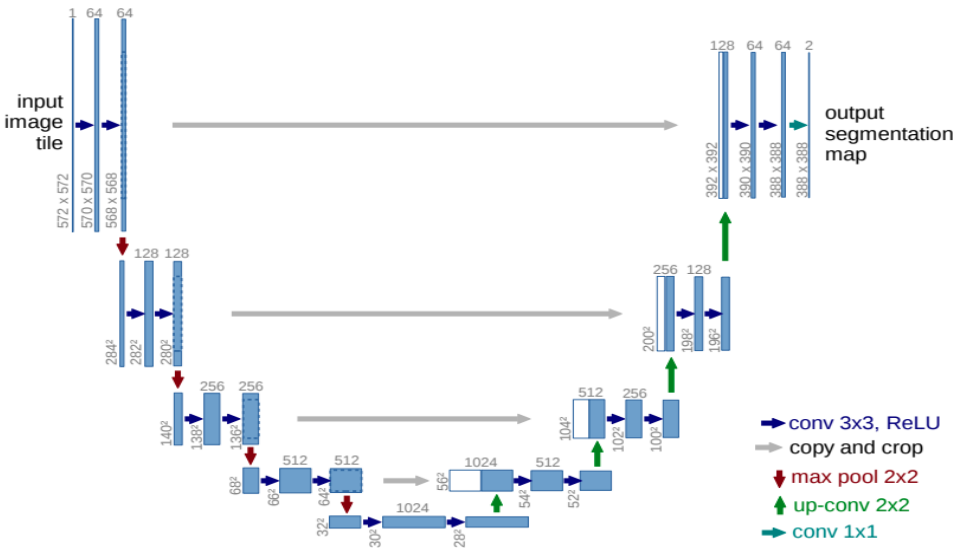


Figure 3.9: A U-Net model for biomedical image segmentation. Figure from [93].

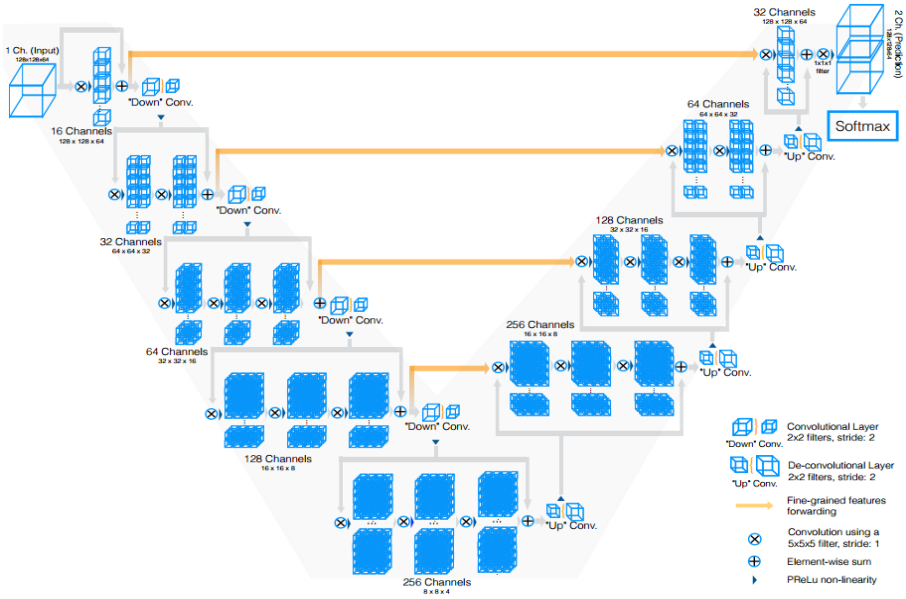


Figure 3.10: A V-Net model for biomedical image segmentation. Figure from [74].

The two models in Figures 3.9 and 3.10 are popularly known for segmenting biomedical images. The U-Net model presented in Figure 3.9 was originally developed and applied to various biomedical image segmentation problems in 2D. The V-Net in Figure 3.10 refrains from processing the input volumes slice-wise instead a volumetric convolutions is applied in that model.

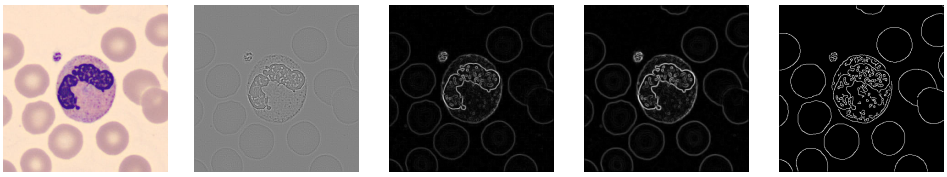
3.4 Edge-based Methods

In edge-based segmentation, an edge filter is applied to the image, and pixels are classified as edge or non-edge depending on the filter output and pixels that are not separated by an edge are allocated to the same category [32]. These methods are based on detecting edges which are paths of rapid changes in intensity. In this context, the edges make the boundary of the region of interest which is the target of segmentation. The segmentation based on edge-detection mainly performs two steps. In the first step, the algorithm finds the pixels that are edge pixels of an object to be segmented. Then, the algorithm refines the edge detected by linking the adjacent edges and combine to form the whole object.

There are different techniques which are used for detecting edges in the images. The popular techniques for detecting edges are based on discontinuity and edge detection gradient operators. The methods include Roberts cross-gradient operators, Sobel Operators, Pre-witt operators, Marr-Hildreth edge detection, Laplacian of a Gaussian(LoG) edge detection and Canny Edge Detection [9, 33, 70, 108]. Edges detection approaches are expected to give results of pixels lying on edges only. But

in practice, the edges break because of the non-uniform illuminations plus other factors that contribute to discontinuities in the intensity values. The edge linking can be performed by global processing using Hough transformation [33].

In practice, the edge-based methods do not perform very well with images in which there are too many edges and when the edges are not welldefined [120]. Figure 3.11 presents example showing the performance of different edges detectors.



(a) Original image (b) Laplacian (c) Roberts (d) Prewitt (e) Canny

Figure 3.11: **Examples of edges detectors.** *(a) image showing a WBC neutrophil (with a nucleus) and red blood cells (RBCs) seen in the background. (b) Laplacian (c) Roberts (d) Prewitt (e) Canny edge detectors. Input image obtained from [1].*

3.5 Region-based

This is a group of segmentation methods which work iteratively by grouping together neighboring pixels with similar values and splitting groups of pixels with different values [32]. The region based method works by using different techniques which are "region growing" and "region split and merge".

In region growing, an area increases from the selected seed pixel by adding neighboring pixels that are similar to the seed. The approach works iteratively until all the pixels in the image belong to a region.

In region split and merging, the idea is to break the image into a set of disjoint regions considered to be coherent within themselves. Initially, the whole image is considered to be the region of interest. Then, look at the region of interest in the image, split it based on constraint satisfied by the region. The process of splitting sub-areas continues until no further splitting is needed. Then, adjacent areas or regions are compared and merged if necessary [37]. This group of segmentation relies more on the uniformity of the pixel values in the region of interest.

This group of methods are said to be sequential and quite expensive both in computational time and memory and for the region growing the method highly depends on the seed selection [120].

Example of method which are region-based include watershed. The method is popular for separating touching objects. Figure 3.12 gives an example of watershed algorithm separating overlapping objects. In this figure, watershed algorithm has been used to separate overlapping RBCs.

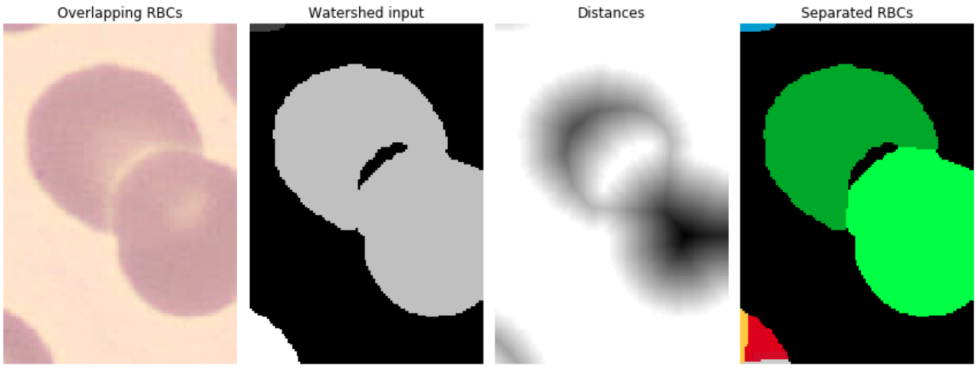


Figure 3.12: **Separation of touching RBCs using watershed.** *From the left: Original image, binarized image, distances and separated red blood cells (RBCs).*

3.6 Clustering based Methods

As stated in the objectives the essence of image segmentation is to represent complex information into segments that are meaningful and simple to understand. Instead of viewing the whole image at one glance, it makes it simple with a few significant segments or clusters. So, the problem of segmenting images may be considered as a clustering problem. In this approach, the pixels in the image are clustered into different clusters after satisfying defined criteria.

So, the clustering methods focus on classifying unlabeled pixels in the image into homogeneous groups such that pixels within the same cluster have high similarity and pixels between cluster should be different. The problem of segmenting images using clustering methods in [33, 75] is mathematically defined as, an image A of size $(M \times N)$ defined over D dimensions generated K clusters $\{C_1, C_2, C_3 \ldots, C_K\}$ subject to the following conditions:

$$C_i \neq \emptyset, \text{ for } i = 1, 2, \ldots K. \tag{3.27}$$

$$C_i \cap C_j \neq \emptyset \text{ for } i \text{ and } j = 1, 2, \ldots K \text{ and } i \neq j. \tag{3.28}$$

$$\cup_{i=1}^{K} C_i = A. \tag{3.29}$$

The condition presented by equation (3.27) takes care of forming non-empty cluster, wheres condition presented by equation (3.28) ensures that the clusters formed are mutually exclusive and equation (3.29) indicates that the clusters formed should cover or represent the whole image.

A number of segmentation methods based on clustering have been developed. However there is no precise definition of a segmentation method based on this technique [75]. The famous clustering approaches which are used in image segmentation is simple linear iterative clustering (SLIC) [2], K-means and fuzzy-means approaches [11, 48, 49, 75]. We briefly present the segmentation approach based on K-means and SLIC which use the modified idea of K-means.

3.6.1 K-means based segmentation methods

K-means is a clustering method that aims at partitioning n observations into K clusters in which each observation belongs to the cluster with the nearest mean cluster centers and the cluster center is considered to be a prototype of the cluster [29, 63, 66, 67, 116]. The K-means concept originally was presented in 1956 [116] but the standard algorithm was first proposed by Stuart Lloyd in 1957 [116] which was not published as a journal article until 1982 [63]. The term K-means was first used explicitly by James MacQueen [67].

Image segmentation is one of the areas in which different methods have developed from the K-means. The methods in this category could be solely based on K-means or partly mixed with other concepts.

Generally, the approach works by partitioning a set of data points, $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, into a number of K clusters [67, 75]. Each observation is a d-dimensional real vector. The objective of K-means is to minimize the sum of square distance between all points and the cluster centers. It is defined as

$$J = \sum_{j=1}^{K} \sum_{i=1}^{n} \parallel \mathbf{x}_i^j - \mathbf{c}_j \parallel^2 . \tag{3.30}$$

where K = number of clusters, n = number of cases, $\mathbf{x}_i$ = case i, and $\mathbf{c}_j$ is a centroid for cluster j, $\parallel x_i^j - \mathbf{c}_j \parallel^2$ is a distance function.

For image segmentation we consider a 2D image $A[x, y]$ which is to be segmented into K segments. The input image can be reshaped to take the form of X data points that K-means processes. The steps for segmenting the images using K-means are summarized from [22, 96] and they are presented as follows

1. Initialize number of K cluster centers.

2. For each pixel of the image, calculate the distance d (mostly Euclidean), between the center and each pixel of the image.

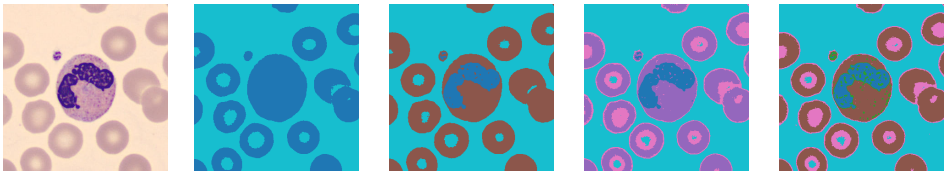
3. Assign each pixel to its nearest center.

4. Find the mean in each cluster to update the clustering centers.

5. Repeat the process in step 2 and 3 of reassigning points to new cluster centers and updating the cluster centers until the convergence criteria is reached.

6. Reshape the cluster pixels into image.

Figure 3.13 gives an example of segmented images using K-means clustering algorithm.



(a) Original image (b) K=2 (c) K=3 (d) K=4 (e) K=5

Figure 3.13: **Segmented image using K-means clustering method with different number of clusters.** *(a) image showing a WBC neutrophil (with a nucleus) and red blood cells (RBCs) seen in the background. The image is segmented by setting different cluster size as indicated is subfigure (b) through (e). Input image obtained from [1].*

3.6.2 Simple linear iterative clustering-SLIC

The algorithm mainly involves three steps which include initialization, assignment, and enforce-connectivity. It performs K-means segmentation using 5 dimension space of color information and image location. The RGB color space is transformed into $CIELAB$ color space [88] and then each of the pixel images is mapped into (l, a, b, x, y).

The algorithm starts by initializing K number of clusters which are sampled on a regular grid. Then, in the assignment step each pixel p_i is associated with the nearest cluster centre. Different from the standard K-means, SLIC updates the cluster centres by searching in a certain neighbourhood and not on the whole image. But the update step depends on the convergence of the error term which is obtained by computing the residual error between the old and the new centres. In case there are disjoint pixels from the previous steps then enforcing connectivity is applied so as to assign them a nearest superpixel [2].

The algorithm uses distance that is obtained from two distances which are combined by normalizing the color proximity and the spatial proximity by maximum distance within their clusters N_c and N_s respectively. Mathematically,

$$d_c = \sqrt{(l_i - l_j)^2 + (a_i - a_j)^2 + (b_i - b_j)^2}. \tag{3.31}$$

$$d_s = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \tag{3.32}$$

So, the color and spatial proximity are normalized and represented by equation (3.33).

$$D' = \sqrt{\left(\frac{d_c}{N_c}\right)^2 + \left(\frac{d_s}{N_s}\right)^2}. \tag{3.33}$$

Since determining the maximum color distance N_c is not so straightforward, because color distances can vary significantly from cluster to cluster and image to image, N_c

is fixed to a constant value. In practice the algorithm uses the equation (3.34).

$$D^{'} = \sqrt{(d_c)^2 + \left(\frac{d_s}{N_s}\right)^2}. \tag{3.34}$$

D' is the distance between the candidate pixel and the cluster center. SLIC algorithm can be applied on different segmentation problems. Figure 3.14 gives an example of SLIC approach for segmenting nucleus from the WBC.

(a) Original image (b) K=12 (c) Segmented nucleus

Figure 3.14: **Segmentation of WBC nucleus using SLIC.** *(a) image showing a WBC neutrophil (with a nucleus) and red blood cells (RBCs) seen in the background, (b) image segmented using SLIC, (c) segmented nucleus after removing the non-interesting objects. Input image obtained from [1].*

3.7 Model based segmentation methods

This is a type of segmentation method that is based on optimizing the shape of a model during the segmentation process. The model-based segmentation methods depend on the choice of deformation strategy which could be either by deforming the embedding space of a shape or by modifying its parameters or Degree of Freedom (DOF) [38]. The former strategy can be qualified as a registration approach whereas the latter strategy can be qualified as a deformable model approach. The geometric representation of the models is specifically important when the deformation is done by modifying parameters.

There are different representations and mostly they are problem-dependent. Some of the deformable model formulations include, simplex meshes, level-sets, triangulated meshes, and spline curves or surfaces [18, 33, 38]. The grouping of different methods based on deformation models is summarized in Figure 3.15.

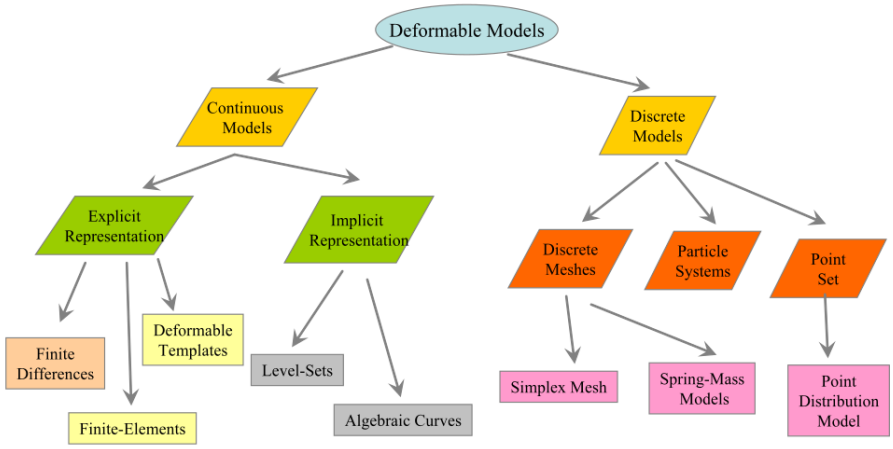


Figure 3.15: **Taxonomy of deformable models.** *The classification scheme of deformable models for medical image segmentation is based on their geometric representation. Figure obtained from [38].*

Figure 3.16 gives an example of one of the model-based segmentation. It can be seen from the figure the way the nucleus has been identified.

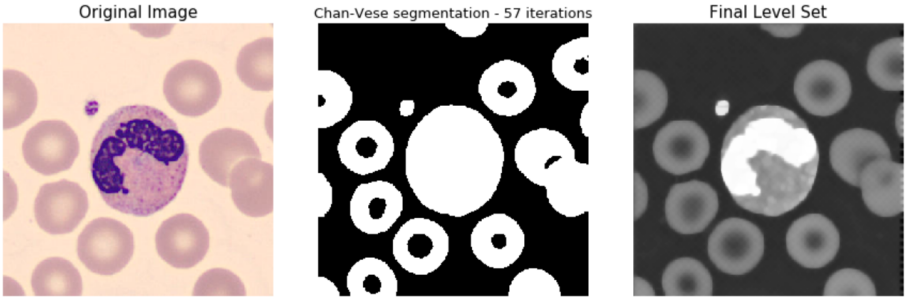


Figure 3.16: **Example model based segmentation:** *From the left: image showing a WBC neutrophil (with a nucleus) and red blood cells (RBCs) seen in the background, Chan-vese segmentation and Final level set segmentation.*

3.8 Graph-based methods

3.8.1 Graph

A graph G is a structure connecting a set of objects in which some pairs of the objects are related. Each object corresponds to a vertex or node v and the Objects are related by a link or an edge e. So, a graph is a pair $G = (V, E)$, such that each $v \in V$ and each $e \in E$.

Some important graph properties are useful before defining any graph-based problem. Graphs could be connected or unconnected; directed or undirected; weighted or unweighted and graphs could be simple or multi-graph.

The popularity of graph-based methods to be applied to different problems including image segmentation is possibly attributed to the following reason: Graphs provide simple representation and are simple to be adapted to solve different problems. This implies that graphs have flexible representation. They are efficient but this fact may depend on the choice of data structure.

3.8.2 Defining the problem

The problem is considered to be a graph partitioning problem. Graph-based approaches solve the problem of image segmentation after mapping an image A into a graph $G = (V, E)$ such that V is the set of nodes and E is the set of edges. Each node in the graph G represents a pixel and an edge e represents a connection between two neighboring pixels in the input image. The weight of each edge $w(e)$ is a measure of similarity between adjacent pixels. Throughout this subsection, we will use the same definition of a graph.

In order to segment the pixels in the image, the graph G is partitioned into n disjoint connected component G_1, G_2, ..., G_n such that nodes in the same connected component G_i are more homogeneous while the homogeneity across different connected components G_i and G_{i+1} is low.

Different methods perform image segmentation after representing the image into a graph. These include normalized cut [109, 110], graph cut [34], methods based on minimum spanning tree [27, 64, 72, 95, 129, 137, 138]. Other graph-based methods included random walker and Isoperimetric Graph Partitioning [138]. Also, some methods are partly based on the graph but combined with other techniques.

3.8.3 Normalized Cut

The problem is defined by a graph G being partitioned into two disjoint sets B_1, B_2, such that $B_1 \cup B_2 = V$ and $B_1 \cap B_2 = \emptyset$. The degree of dissimilarity between these two pieces can be computed as total weight of the edges that have been removed [110]. The cut is defined as

$$cut(B_1, B_2) = \sum_{u \in B_1, v \in B_2} w(u, v). \tag{3.35}$$

The optimal partitioning of a graph is the one that minimizes this cut value. But it has been noticed that the minimum cut criteria favor cutting small sets of isolated nodes in the graph. A new measure was proposed to alleviate this unnatural bias. They proposed a new measure of disassociation between two groups to be partitioned. The new measure computes the cut cost as a fraction of the total edge connections to all the nodes in the graph and defines it a normalized cut ($Ncut$).

$$Ncut(B_1, B_2) = \frac{cut(B_1, B_2)}{assoc(B_1, V)} + \frac{cut(B_1, B_2)}{assoc(B_2, V)}. \tag{3.36}$$

where $assoc(B_1, V)$ is $\sum_{u \in B_1, t \in V} w(u,t)$ is the total connection from nodes in B_1 to all nodes in the graph and $assoc(B_2, V)$ is defined in the same way. Due to the fact that minimizing normalized cut is exactly NP-complete, they embed the normalized cut problem in the real value domain to approximate a discrete solution which seems to be found efficiently [110].

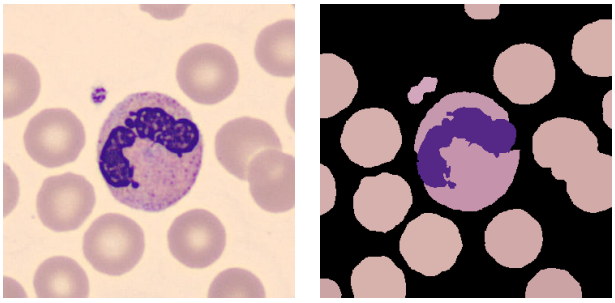
To make the idea clear we can assume that a graph G is partitioned into two sets B_1 and B_2. Let $\mathbf{x}$ be an N-dimensional indicator vector, where $x_i = 1$ if node i belongs to set B_1, and $x_i = -1$ otherwise. Let $\mathbf{W} = [w(i,j)]$ be an $N \times N$ symmetrical matrix, and $\mathbf{d}(i) = \sum_j w(i,j)$ is the row sum of the matrix $\mathbf{W}$, and $\mathbf{D}$ is a diagonal matrix with d on its diagonal [110, 128]. Minimizing the normalized cut over all possible indicator vector x is given by

$$min_{\mathbf{x}} Ncut(\mathbf{x}) = \frac{\mathbf{y}^T(\mathbf{D} - \mathbf{W})\mathbf{y}}{\mathbf{y}^T\mathbf{D}\mathbf{y}}. \tag{3.37}$$

such that $y(i) \in \{1, -b\}$, $\mathbf{y} = (1+\mathbf{x}) - b(1-\mathbf{x})$ when $\mathbf{y}$ is relaxed to take real values, the equation (3.37) can be minimized by solving general eigenvalue system.

$$(\mathbf{D} - \mathbf{W})y = \lambda \mathbf{D} y. \tag{3.38}$$

where λ is the eigenvalue. More information on the derivation of the idea is presented in [110]. Figure 3.17 presents an example of object in the image segmented using normalized-cut.



(a) Original image (b) Normalized cut

Figure 3.17: **Example of segmentation by normalized cut.** *(a) image showing a WBC neutrophil (with a nucleus) and red blood cells (RBCs) seen in the background, (b) Normalize cut results. Input image obtained from [1].*

3.8.4 Graph-cut segmentation method

Graph-cut approach solves the problem of image segmentation by finding the minimum cut of the graph. Different from other graph-based segmentation approaches, graph-cut defines two terminal nodes S and T to the graph G of an image which is used as source and sink node during the partitioning of the graph [8].

$$V = P \cup \{S, T\}. \tag{3.39}$$

where P are the nodes of the graph corresponding to pixels, S is the source node representing the object terminal and T is the sink node representing a background terminal. The edges are

$$E = N \bigcup_{p \in P} \{\{p, S\}, \{p, T\}\}. \tag{3.40}$$

where $\{p, S\}$ and $\{p, T\}$ are $t-$links (these are links connecting to each terminal S and T). An edge connecting two pixels $\{p, q\}$ is an $n-$link in the neighborhood system N. The weight of $n-$link and $t-$link define the boundary and regional term of the cost function respectively.

3.8.5 Minimum spanning tree-based segmentation methods

The minimum spanning tree (MST) is a subset of edge-weighted and undirected graph that connects all the vertices in the graph together, without forming any cycles and with the minimum possible total edge weight [65, 101, 102]. The methods in this group are graph-based methods which partitions images into non-overlapping segments using the MST of the constructed graph.

The idea of MST has existed for many decades and different algorithms have been developed for its construction. The famous algorithms for constructing the MST include Boruvkas algorithm in 1926 [7], Jarnik's algorithm in 1930 [47] (which was later rediscovered and republished by Robert C. Prim in 1957 [87] and Edsger W. Dijkstra in 1959 [23]), Kruskal's algorithm in 1956 [55] and the reverse-delete algorithm (which is the reverse of Kruskals algorithm and it first appeared in [55]).

Different image segmentation algorithms based on MST have been developed. Some of the developed segmentation algorithms based on this idea include the work from [27, 64, 72, 95, 129, 137].

3.8.5.1 MST based segmentation method by Zahn

The work by Charles T. Zahn [137] addresses the problem of detecting an inherent separation between clusters of given points by a distance function. The work is motivated by the perception of separating two-dimensional point sets as separate groups or *gestalt*. The method uses MST for detecting clusters and is applied to point clustering and image segmentation. The segmentation criterion in this method is to break MST edges with large weights which are called inconsistent edges.

Even though the work of Clark and Miller [14] is not focused on image segmentation it is one of the earliest work that uses MST for detecting structures. In his work, Zahn states that "*the use of MST graph as an aid to detecting and describing the structure of points clusters was suggested by the process of spark-chamber photographs reported by Clark and Miller.*"

(a) Original image (b) T = 0.32 (c) T = 0.24 (d) T = 0.21 (e) T = 0.21

Figure 3.18: **Nucleus segmentation by using Zahn MST-based method with different thresholds.** *(a) image showing a WBC neutrophil (with a nucleus) and red blood cells (RBCs) seen in the background. (b) when the threshold $T = 0.32$, number of connected components $CC = 2$, (c) when the threshold $T = 0.24$, number of connected components $CC = 14$, (d) when the threshold $T = 0.21$, number of connected components $CC = 18$, (e) similarly, when the threshold $T = 0.21$, number of connected components $CC = 33$. See explanation in the next paragraph. Original image from [1].*

The graph constructed from image in Figure 3.18 (a) has 130680 nodes so, its MST has 130679 edges. After setting the threshold T the number of inconsistent edges to be removed can still be controlled. For Figure 3.18 (b) $T = 0.32$, the number of edges removed is 4/4, for Figure 3.18 (c) $T = 0.24$, the number of edges removed is 88/88, for Figure 3.18 (d) $T = 0.21$, the number of removed edges is 120/226, and for Figure 3.18 (e) $T = 0.21$, the number of edges removed is 210/226.

Figure 3.18 shows a nucleus segmented by using Zahn's MST-based method. The method separates the connected components by setting a threshold to remove inconsistent edges. Depending on the selected threshold, it appears that some regions within the nucleus are split into multiple regions (see Figure 3.18(c), (d) and (e)). This happens because of the reason that differences between pixels within regions of high variability can be larger than the differences between the ramp and the constant region [27].

3.8.5.2 MST based segmentation method by Felzenszwalb and Huttenlocher

The work by Felzenszwalb and Huttenlocher [27] addresses the problem of segmenting an image into regions. The method is defined similarly to the method by Zahn in [137] but differs by adaptively adjusting the segmentation criterion based on the degree of variability in neighboring regions of the image. It is a greedy algorithm but obeys certain non-obvious global properties [27]. We summarize the idea and use symbols and definitions as presented in [27].

Graph-based image segmentation techniques generally represent the problem in terms of a graph $G = (V, E)$ where

- each node $v_i \in V$ corresponds to a pixel in the image,
- an edge $(v_i, v_j) \in E$ connects vertices v_i and v_j.

- Each edge $(v_i, v_j) \in E$ has a corresponding weight $w(v_i, v_j)$, which is a measure of the dissimilarity between neighboring pixels v_i and v_j.

Segmentation Formulation

The segmentation problem can be formulated as partition of the set of vertices V of the given undirected graph G into components $C_1, C_2, ...$ such that,

- edges between two vertices in the same segment C_i should have relatively lower weights.
- edges between two vertices in different segments C_i and C_j should have higher weights.

Partition Strategy

- **Internal Difference:** The internal difference of a component $C \subseteq V$ is the largest weight in the minimum spanning tree of the component $MST(C, E)$ defined by

$$Int(C) = \max_{e \in MST(C,E)} w(e). \tag{3.41}$$

- **Component Difference:** The difference between two components $C_1, C_2 \subseteq V$ is the minimum weight edge connecting the two components.

$$Dif(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2, (v_i, v_j) \in E} w(v_i, v_j). \tag{3.42}$$

If there is no edge connecting C_1 and C_2, we let

$$Dif(C_1, C_2) = \infty. \tag{3.43}$$

- **Boundary Predicate:** Criteria for existence of the boundary between two components.
There exists a boundary between two components if the Component difference between the components is greater than the Internal Differences of either of the components. The pairwise comparison predicate $D(C_1, C_2)$ is

$$D(C_1, C_2) = \begin{cases} \text{True if } Dif(C_1, C_2) > MInt(C_1, C_2) \\ \text{False Otherwise} \end{cases}$$

where

$$MInt(C_1, C_2) = min(Int(C_1) + \tau(C_1), Int(C_2) + \tau(C_2)). \tag{3.44}$$

τ is a threshold that controls the predicate D. For small components, $Int(C)$ is not a good estimate of the local characteristics of the data. In the extreme

case, when $|C| = 1$, $Int(C) = 0$.
Therefore, we use a threshold function based on the size of the component,

$$\tau(C) = k/|C|. \tag{3.45}$$

where $|C|$ denotes the size of C, and k is some constant parameter which sets the scale of observation and not a minimum component size.

3.8.5.3 Interactive MST based for brain tumor segmentation

The work presented in [72] performs interactive segmentation of brain tumors based on MST. Initially, the image is processed and then a graph is constructed followed by making MST. The segmentation of the ROI is performed interactively.

A path P in graph G is defined as a sequence of edges joining two terminal nodes say $P = \{v_i, \ldots, v_k\}$ where v_i and v_k are the terminal nodes in the path.

Segmentation Criteria

The segmentation criteria highly depends on the existence of the boundary between the ROI and the background.

Let R_1 and R_2 be regions each containing several vertices in the MST. Let v_i and v_j be vertices in the regions R_1 and R_2 respectively. The boundary between R_1 and R_2 is defined by the edge with the maximum weight in the path P connecting $v_i \in R_1$ and $v_j \in R_2$. Then, the segmentation criterion is

$$Bd(R_1, R_2) = \max_{e_k \in P} w(e_k). \tag{3.46}$$

where, $Bd(R_1, R_2)$ represents the boundary between regions R_1 and R_2, $w(e_k)$ is the weight of edge e_k. The segmentation is performed interactively. For more information about the method and the implementation see paper B.

3.8.5.4 GUBS for brain segmentation

GUBS is a graph-based method for brain segmentation in MRI [73]. The proposed method is Graph-based Unsupervised Brain Segmentation (GUBS) which processes 3D MRI images and segments them into brain tissues, non-brain tissues, and background. GUBS first constructs the voxel neighbor graph from a preprocessed MRI image, weights it by a difference in voxel intensities, and computes its MST.

It then uses domain knowledge about the different regions of MRI to sample representative points from the brain (T_B), non-brain (T_{NB}) and background (T_{BG}) regions of the MRI image. Given a graph G and a subset H of the nodes in G we construct a graph where all the nodes in H have been collapsed to a single node. In order to ease the construction of this graph we represent the single collapsed node by a node $h \in H$.

We construct the graph G_H^h from G, h and H as follows:

1. For every edge e in G, if a node v in H appear in e, the edge e is modified by replacing v by h.

2. Remove all the nodes in H except the node h from the modified graph to obtain the graph G_H^h.

We call the graph G_H^h the graph obtained from G, h and H by collapsing the nodes in H onto h.

Segmentation criteria

The segmentation criteria used in paper B is modified by adding more conditions because of the complexity of the structures to be separated. We want to separate the graph G into three subgraphs G_B, G_{NB} and G_{BG} representing the brain, non-brain tissues, and background.

Let $V_B \subset V$, $V_{NB} \subset V$ and $V_{BG} \subset V$ be sets of nodes in a graph G corresponding to coordinate points from T_B, T_{NB} and T_{BG} respectively. We construct the graphs G_B, G_{NB} and G_{BG} so that the nodes V_B, V_{NB} and V_{BG} are in G_B, G_{NB} and G_{BG} respectively.

Let $v_B \in V_B$, $v_{NB} \in V_{NB}$ and $v_{BG} \in V_{BG}$ be randomly sampled nodes. Let $A = G_{V_B}^{v_B}$, $B = A_{V_{NB}}^{v_{NB}}$, $C = B_{V_{BG}}^{v_{BG}}$. The modified graph C has the advantage of reducing the graph G while preserving the containment of the sampled nodes in respective subgraphs.

The MST is constructed from the modified graph C and components are extracted representing brain, non-brain tissues and the background as follows: We modify the MST by removing the edges connecting the brain and the non-brain regions followed by modifying the MST to separate the non-brain tissues from the background. The results are the connected components of the graph with three different labels. The labels are reshaped back to the shape of the input MRI image. For more information about the method find the GUBS manuscript, paper C.

3.8.5.5 Examples of results segmented from different MST based methods.

In this subsection, we give different examples of segmented images comparing MST-based approaches.

Example 1: In this example, we compare the results obtained by segmenting a nucleus of erythroblast using three MST-based methods. Note that the Felzenszwalb and Huttenlocher results presented in Figure 3.19 (b) use the implementation in skimage which adds a parameter for controlling the minimum size of the components which is said to be enforced using postprocessing.

(a) Original image (b) Felzenszwalb (c) Felzenszwalb (d) Interactive (e) GUBS

Figure 3.19: **Nucleus segmentation of erythroblast by using MST-based methods.** *(a) image showing a WBC erythroblast (with a nucleus) and red blood cells (RBCs) seen in the background. (b) and (c) Felzenszwalb segmentation (k=0.3, Minimum component size =2000), (d) segmentation by MST approach presented in paper B (originally developed for segmenting brain tumors), (e) segmented nucleus using GUBS (originally tested on segmenting brain from 3D MRI). For GUBS, the values less than 0.3 are the sampled points in the background and values greater than 0.7 are sampled points in the ROI). Original image from [1].*

Example 2: In this example (Figure 3.20), we compare the segmentation steps used for segmenting the nucleus and cytoplasm of erythroblast using three MST-based methods. All the three methods compared perform well but for this kind of image the interactive MST-based approach is straight forward.

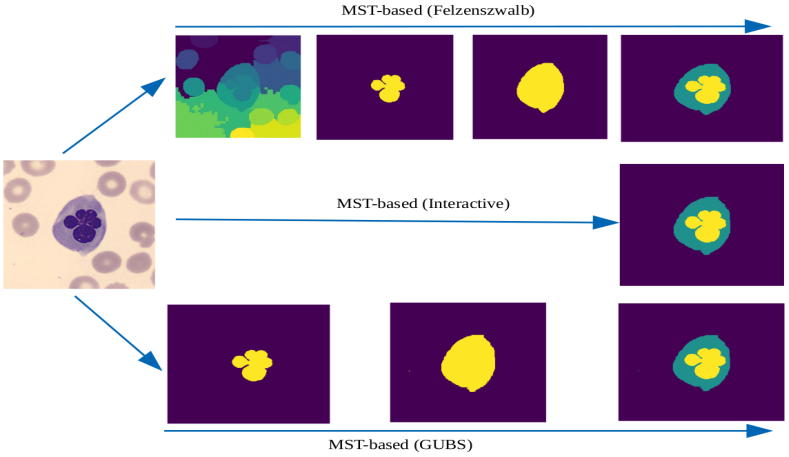
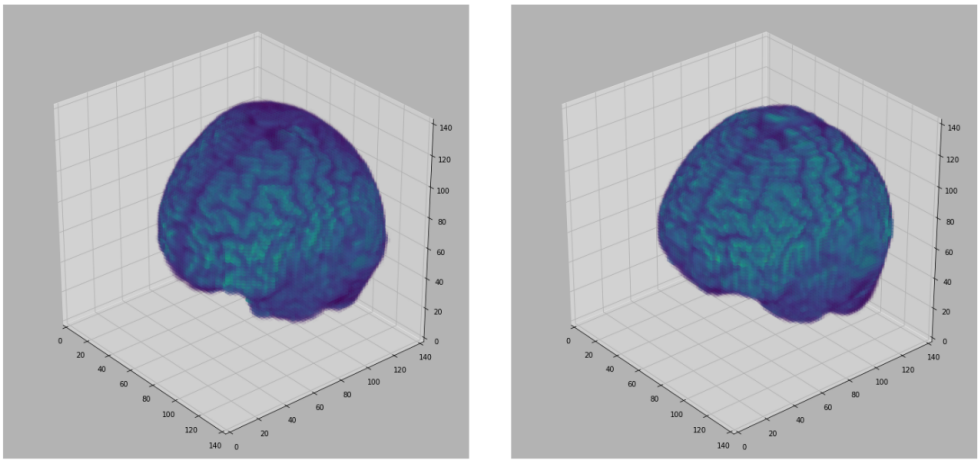


Figure 3.20: **Segmentation of erythroblast by using MST-based methods:** *The original image is erythroblast segmented by three different MST based methods. In each approach the final segmentation is nucleus and cytoplasm.*

Example 3: An example given for experimenting GUBS on 3D MRI data set (see Figure 3.21). The method has been experimented on three different 3D MRI data sets. We Visualize a representative subject from the Open Access Series of Imaging

Studies (OASIS). The GUBS method is flexible and can be adapted to different data sets.



(a) (b)

Figure 3.21: **Segmented Brain (OASIS data):** *One representative subject representing (a) 3D brain segmented using GUBS approach (predicted), (b) 3D brain (ground truth). The masks were segmented using a custom method based on registration to an atlas, and then revised by human experts. Figure from paper C.*

Example 4: Experimenting with GUBS and comparing it with edge-based and region-based segmentation methods using the coin's image. GUBS has been developed and originally tested on 3D MRI data. We found out that it was important to test it on different image data sets for illustration. The segmented labels show the performance of GUBS compared to the two other methods tested [16]. For an illustration of the GUBS method, we use a different image segmented by different methods to compare the results. See Figure 3.22.

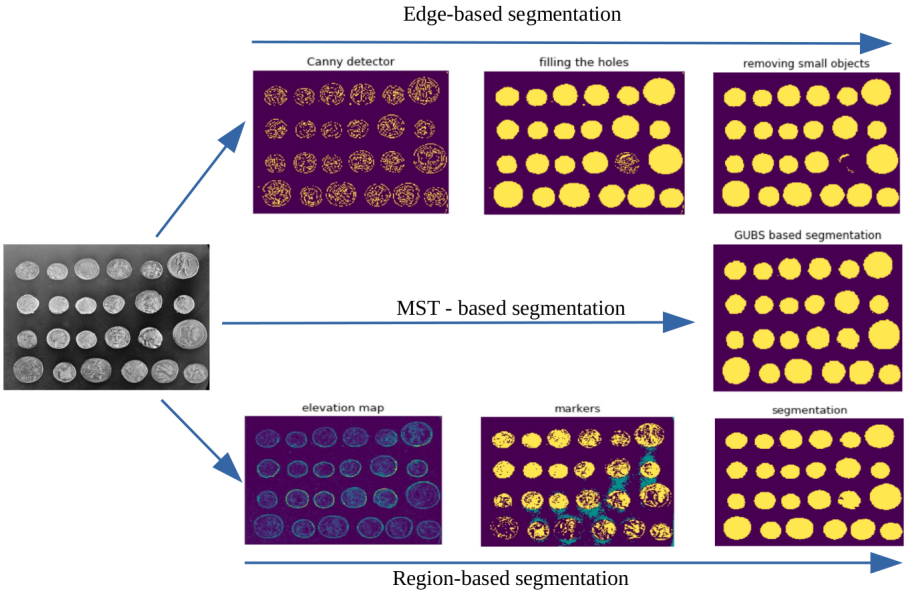


Figure 3.22: **Illustration from coins image:** *The original image and the code used for comparing for edge-based and region based are presented here [16]. We run the GUBS method and compare the segmented labels.*

Generally, the choice of any segmentation method for segmenting any kind of data could be influenced by the type of the data and the problem to be solved. Methods work differently based on how they are formulated, the kind of input and how they process the input. All methods have their strength and weakness. Therefore, the choice of the method for any segmentation problem is an important step before performing the segmentation of ROI.

This chapter has presented different methods which are used for image segmentation. Each methods is presented with some examples of segmented images. Methods in the same group are compared by segmenting an image to check their performance on the same image. The chapter gives more weight on explaining different methods based on thresholding, deep learning and MST including those which have been developed and presented in the included papers. The next chapter presents different strategies which are used for evaluating segmentation algorithms.

Chapter 4

Strategies for evaluating image segmentation algorithms

In the field of medical image segmentation, different algorithm are being developed. It is important to understand their performance by comparing their results to their ground truth. It is also important to evaluate their performance by comparing their results to the most recent existing methods.

4.1 Manual segmentation methods

Manual image segmentation is the process of separating the region of interest from non-interesting regions performed manually by experts. For medical images, manual segmentation is performed by radiologists or specialized clinicians for segmenting specific tissues.

The process involves manual interaction between the expert and the tissues being segmented by using tools developed for that specific task. For 3D reconstructed from stacked 2D slices, the segmentation is performed slice-by-slice. But manual segmentation can also be done directly by experts on 3D volumes. During the development of segmentation methods, the manually segmented images are used as the ground truth or the gold standard of that data.

4.1.1 Ground truth for image segmentation

In the field of image segmentation, the ground truth (gold standard) are segmented masks of ROI prepared by experts for checking the accuracy of the newly developed methods. Mostly, it is done manually by experts who utilize their knowledge in that field. But it is not clear if there are harmonized methods for obtaining the ground truth in each field of study.

The manual segmentation for generating ground truth is prone to error and it is time-consuming. A comparison of manually segmented images from experts shows

that there exist variations (Intra and inter variations). A consensus segmentation for ground truth is suggested in different literature including [69]. Multiple repetitions of segmentation by multiple operators are suggested to reduce the bias in segmented ROI but there are also several ways of averaging the results [119].

Let M_i^j be a mask segmented manually such that i represents an index of an expert and j represents number of repetitions of the segmented mask. We summarize possibilities which can be used for creating consesus mask for ground truth.
Consensus based on averaging the masks:

- **Option one:** Same expert (constant $i = 1$) performing N repetitions of segmentation. That is, averaging masks from the same expert and the masks are repeatedly segmented.

$$\text{Ground truth} = Binary\left(\frac{\sum_{j=1}^{N} M_{i=1}^{j}}{N}\right). \tag{4.1}$$

- **Option two:** Different experts P performing no repetition of segmentation (constant $j = 1$). That is, averaging masks from different experts and the masks are not repeatedly segmented.

$$\text{Ground truth} = Binary\left(\frac{\sum_{i=1}^{P} M_{i}^{j=1}}{P}\right). \tag{4.2}$$

- **Option three:** Multiple repetitions N and different experts P. That is, averaging masks from different experts and the masks are repeatedly segmented by each expert.

$$\text{Ground truth} = Binary\left(\frac{\sum_{i=1}^{P}\sum_{j=1}^{N} M_{i}^{j}}{P \times N}\right). \tag{4.3}$$

Consensus based on intersecting the masks:

- **Option one:** Same expert (constant $i = 1$) performing N repetitions of segmentation. That is, averaging masks from the same expert and the masks are repeatedly segmented.

$$\text{Ground truth} = \bigcap_{j=1}^{N} M_{i=1}^{j}. \tag{4.4}$$

- **Option two:** Different experts P performing no repetition of segmentation (constant $j = 1$). That is, averaging masks from different experts and the masks are not repeatedly segmented.

$$\text{Ground truth} = \bigcap_{i=1}^{P} M_{i}^{j=1}. \tag{4.5}$$

- **Option three:** Multiple repetitions N and different experts P. That is, averaging masks from different experts and the masks are repeatedly segmented by each expert.

$$\text{Ground truth} = \bigcap_{i,j=1}^{P,N} M_i^j. \tag{4.6}$$

Consensus based on union of the masks:

- **Option one:** Same expert (constant $i = 1$) performing N repetitions of segmentation. That is, averaging masks from the same expert and the masks are repeatedly segmented.

$$\text{Ground truth} = \bigcup_{j=1}^{N} M_{i=1}^j. \tag{4.7}$$

- **Option two:** Different experts P performing no repetition of segmentation (constant $j = 1$). That is, averaging masks from different experts and the masks are not repeatedly segmented.

$$\text{Ground truth} = \bigcup_{i=1}^{P} M_i^{j=1}. \tag{4.8}$$

- **Option three:** Multiple repetitions N and different experts P. That is, averaging masks from different experts and the masks are repeatedly segmented by each expert.

$$\text{Ground truth} = \bigcup_{i,j=1}^{P,N} M_i^j. \tag{4.9}$$

The ground truth masks are very useful for evaluating any method. If the ground truth is of poor quality or some information is not correct, it is likely to distort the methods evaluation step.

4.2 Evaluation Methods

Results obtained by using a newly developed algorithm need to be evaluated to check the performance and the accuracy of the new method. These methods depend on the ground truth.

There are different techniques that are used for evaluating segmentation methods. We define Jaccard Index (JI), Dice Similarity Coefficient (DSC), sensitivity, specificity, accuracy, and precision.

Assuming that the pixels in the binary labels that are correctly classified as ROI are represented as true positive (TP). The pixels that are incorrectly classified as

ROI are represented as false positive (FP). The pixels that are correctly classified as non-ROI regions are represented by true negative (TN). The pixels that are in the ROI but incorrectly classified as non-ROI are represented by false negative (FN). The Jaccard Index and Dice Similarity Coefficient are computed by

$$JI = \frac{TP}{TP + FP + FN}, \; DSC = \frac{2TP}{2TP + FP + FN}. \tag{4.10}$$

The sensitivity and specificity show the percentage of ROI and non-ROI voxels recognized respectively. They are computed by

$$Sensitivity = \frac{TP}{TP + FN}, \; Specificity = \frac{TN}{TN + FP}. \tag{4.11}$$

The precision gives the ratio of the correctly positive identified labels against all the labels, whereas accuracy gives the ratio of the correctly identified labels.

$$Precion = \frac{TP}{TP + FP}, \; Accuracy = \frac{TP + TN}{TP + FP + FN + TN}. \tag{4.12}$$

In this thesis, for each data sets segmented in the three papers we have evaluated the results by using the corresponding ground truth which are available together with the data sets. Also, in each paper examples are presented to visualize the performance of the methods developed. The evaluating techniques presented in this subsection were mostly used for evaluating the performance of the developed methods.

4.3 State of the art methods

The state of the art segmentation methods are those which are considered to use very modern and most recent ideas. They are methods at the advanced level which are considered to have high accuracy that can be referred for comparison.

Apart from the traditional [3] segmentation methods, many recent approaches have emerged and proved their efficiency for image segmentation. Most of the recent methods of images segmentation include those which are based on deep learning. Their efficiency relies on the trained annotation together with their corresponding inputs.

Deep learning is a specialized subset of machine learning which is also a subset of artificial intelligence. It relies on structured algorithms known as artificial neural network. The neural networks describe algorithms that analyze data with a logical structure, and this happens either by supervised or unsupervised learning [17]. Also, there are self-supervised methods which do not depend on labels. The famous deep learning architectures for segmenting biomedical images with high accuracy are convolutional neural network (CNN). The popular outstanding segmentation networks based on CNN include U-Net [93], V-Net [74], Mask R-CNN, RefineNet and DeconvNet [61].

In this thesis various classical methods have been developed for segmenting medical images. The methods include an automatic thresholding method for segmenting

nucleus and cytoplasm, a graph based method utilizing minimum spanning tree for segmenting brain tumors from MRI and, a Graph-Based Unsupervised Brain Segmentation (GUBS) for segmenting 3D brain volume. The results obtained by using these methods were compared to the results obtained by using state of the art methods for evaluating their performance.

4.3.1 State of the art methods on Medical images

Due to the need for accuracy of the segmented images in different fields of applications, methods are developed and applied differently. The deep learning methods are widely used for segmenting medical images and they are powerful tools for semantic segmentation [122]. Because of the extensive research done in segmenting medical images using deep learning models, the approaches have been successful in different medical image segmentation tasks. They have been widely used for classification and support the process of disease detection and diagnosis.

The deep learning methods especially the convolutional neural networks are best suited for segmenting biomedical images because of the high accuracy in their performance. The accuracy achieved by the deep learning methods is used for comparing the performance of other methods. In this thesis, the medical images were segmented using classical methods and their results were presented in the three papers. For the results presented in the papers, the deep learning methods present a relatively higher accuracy compared to the classical methods in terms of performance. For example, the results obtained in Paper A the methods segment the nucleus and cytoplasm with an accuracy comparable to the state of the art because of the good contrast between the region of interest and the background. Some segmentation problems do not necessarily need deep learning methods because simple methods including thresholding and graph method can also perform with high accuracy. Deep learning methods are good options for problems that classical methods perform poorly. Generally, thresholding and methods based on minimum spanning trees are simple.

The accuracy achieved by the state of the art method is achieved at the cost of computational resources. Note that, the deep learning methods need a huge amount of data as well as quality data. Because of insufficient images with their corresponding labels augmentation techniques are used for producing more data for training the models. Different from the classical methods, deep learning methods will obtain the results after a long time of training because a huge amount of training data is needed. The results obtained using deep learning methods from different papers which were used for comparing the results presented in paper A, B, and C, the data needed to be augmented before training the models. This is an advantage of the proposed methods over the deep learning methods for segmenting medical images because they do not need a huge amount of data.

Classical segmentation and the deep learning method both depend on ground truth which is obtained by expert labeling. Thresholding methods and those based on minimum spanning trees perform prediction without any reference to the data

to be predicted. They only need the ground truth to compare and check the level of their accuracy. The deep learning especially the supervised methods highly need the ground truth for training the model which is then used for performing prediction. This fact makes deep learning perform with high accuracy. For big data (demand for supervised deep learning methods) unavailability of the ground truth is a drawback because preparing ground truth is time-consuming task for they are manually annotated.

Even though supervised deep learning methods perform with high accuracy their shortcomings have been presented. The need for labels brings a self-supervised method that does not depend on the labels. The self-supervised learning refrains from the demand for annotated labels but still need more resources computationally because they produce their labels. But also, the self-supervised learning model highly compromises the accuracy. The self-supervised learning model makes their labels, if the model predicts a wrong class with high confidence the model will fine-tune the parameters against this false prediction and continue believing that it is correct. This could be challenging for its applicability in the clinical setting because the self-supervised model could be certain in erroneous predictions.

Chapter 5

Overview of the papers

This chapter presents an overview of the developed methods for segmenting medical images. Figure 5.1 shows the interrelations between the methods developed and the corresponding data sets segmented in each paper. The images data sets include microscopic images for normal white blood cells, 2D MRI for brain tumors and different groups of 3D MRI from different studies.

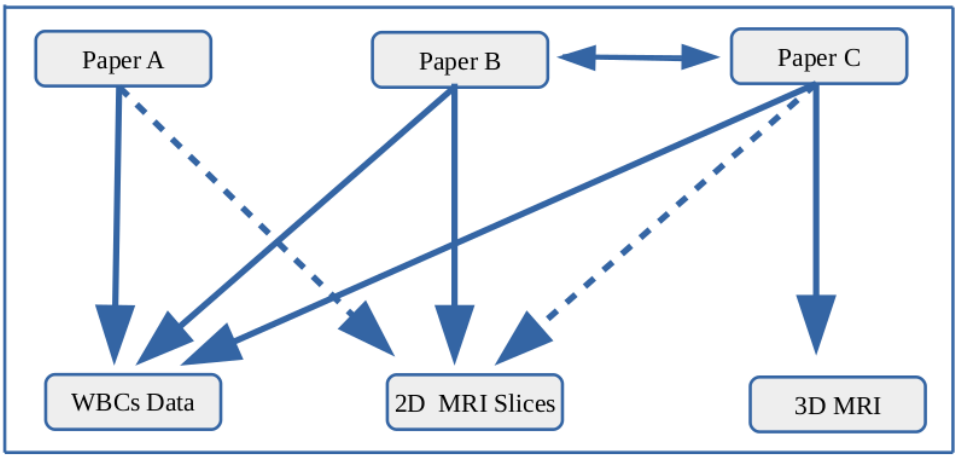


Figure 5.1: **Papers overview and their application to the segmented data set.** *The horizontal double arrow line indicate that the methods in paper B and paper C are related (both are based on Minimum spanning tree). The vertical solid arrow indicate that the method developed in that paper was directly applied on segmenting that data set. The solid slant arrows show that the method in that paper can be used for segmenting the pointed data sets. The dotted slant arrows show that the method in the particular paper can segment the pointed data with high limitation.*

5.1 Summary of the Papers

5.1.1 PAPER A

Threshold-Based: Automatic estimation of the threshold.

Paper A proposes a method for segmenting the nuclei and cytoplasm from white blood cells (WBCs). The proposed method automatically estimates the optimal threshold for segmenting nuclei. The method utilizes an automatic initialization together with defined conditions to estimate the optimal threshold. The initialization is based on the minimum and maximum intensity values in the image to be segmented. The conditions are defined based on the local minima and local maxima of the estimated image histogram. The estimated threshold is applied on image to segment the nucleus followed by post-processing.

The whole WBC is segmented before segmenting the cytoplasm depending on the complexity of the objects in the image. For WBCs that are well separated from red blood cells (RBCs), n thresholds are generated and used for producing n thresholded images. Then, a standard Otsu method is used to binarize the average of the produced images. Post-processing is performed and then a single-pixel point from the segmented nucleus is used to segment the WBC from the red blood cells (RBCs). For images in which RBCs touch the WBCs, the WBC is segmented using SLIC and watershed methods. The cytoplasm is obtained by subtracting the segmented nucleus from the segmented WBC.

The developed method is tested on two different medical images data sets and the results are compared to the state of art methods. The proposed method segments the nucleus and cytoplasm well and hence proving its utility in segmenting nucleus and cytoplasm.

5.1.2 PAPER B

MST-Based: Interactive Segmentation of brain tumor from 2D MRI

Paper B presents MST-based method for segmenting brain tumors. The method performs interactive segmentation based on the MST. The steps involve preprocessing, making a graph, constructing the MST, and a newly implemented way of interactively segmenting the region of interest. A pixel neighbor graph is weighted by intensity differences and then the corresponding MST is constructed. The image is loaded in an interactive window for segmenting the tumor. The region of interest is segmented interactively from the background. The proposed method was tested by segmenting two different 2D brain T1-weighted magnetic resonance image data sets. The comparison confirmed the validity of the approach. The proposed method is simple to implement and the results indicate that it is accurate and efficient.

5.1.3 PAPER C

MST-Based: Automatic segmentation of brain from 3D MRI

Paper C presents a method that processes a 3D MRI volume and partitions it into the brain, non-brain tissues, and background segments. It is a graph-based method that uses MST to segment 3D MRI. The graph is made from a preprocessed 3D MRI volume followed by constructing the MST. The method relies on sampling representative points from the brain, non-brain, and background regions of the 3D MRI.

The corresponding nodes of the sampled points in each region are used as the terminal nodes for paths connecting the regions. The MST is updated twice by disconnecting the path connecting the terminal nodes in the brain and non-brain regions, followed by disconnecting the path connecting non-brain and background regions. The process produces three labeled connected components which are reshaped back to the shape of the 3D MRI. The labels are used to segment the brain, non-brain tissues, and the background. The method was tested on three different publicly available data sets and the results were compared to different state of the art methods. The proposed method gives good and competing results in terms of performance.

Schematic diagram for MST-based approaches in paper B and C

The schematic diagram presented in Figure 5.2 is only for the MST-based approaches presented paper B [72] and paper C [73], not other MST based approaches.

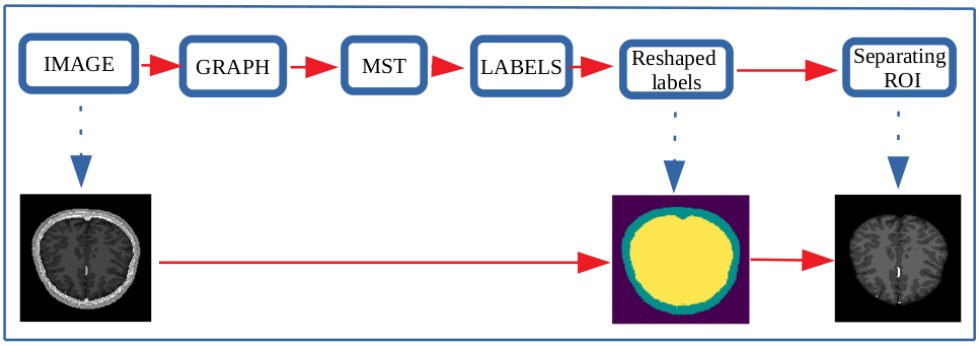


Figure 5.2: **General Flow diagram for the interactive and GUBS MST-based methods.** *The input image is 2D/3D, then a graph is constructed from the input image, MST is the minimum spanning tree of the graph, followed by producing unique labels of the connected component from the MST and then reshape back to the shape of the input image. The reshaped labels are used for separating the regions and extracting the region of interest.*

5.2 Discussions

In this chapter, a discussion of the research findings based on the research questions is presented. The main research question is developing methods based on classical techniques which can segment medical images. For all three papers, the methods for segmenting medical images have been developed and applied to the medical images. The results are compared to the state of art methods, and the conclusion drawn in each research paper shows that the developed methods efficiently segment the medical images. For paper A, the developed method has demonstrated its utility in segmenting nuclei, WBCs, and cytoplasm. The results are satisfactory. For papers B and C the results were also compared to the state of the art and it confirms the validity of the minimum spanning tree approach for segmenting brain tumors from 2D MRI as well as segmenting brain volume from 3D MRI.

There are different existing methods for segmenting medical images. In this thesis, the methods developed for segmenting medical images are based on existing ideas. For each paper, the existing classical methods were reviewed to create conceptual ideas for the new methods. Also, the deep learning methods were reviewed. For example, in paper A, the classical methods for segmenting the nucleus and cytoplasm were reviewed. Also, the convolutional neural network was reviewed. For paper B, methods based on the minimum spanning trees were reviewed. The results based on deep learning were used for comparison. In paper C, the classical methods for segmenting the 3D brain were reviewed as well as the methods based on deep learning together with those which consist of the results for comparison. Each paper presented the contributing ideas for all the methods developed.

In this thesis, different types of images are obtained from different imaging modalities. For each method, the image data were explored, and got a way of representing them to fit in the proposed method. For example, in paper A, two different sets of microscopic images of normal white blood cells were segmented. Images had to be transformed and represented in a form that thresholds can be estimated automatically. For papers B and C, the voxel in each image were mapped and represented as graphs to accommodate the method based on minimum spanning trees. For all the papers, data sets underwent some preprocessing steps before applying the segmentation methods.

For each paper, a method is developed and applied for segmenting the medical images. As described, the developed methods are classical. These methods are based on the pixel values of images and they partition images into nonoverlapping groups based on certain attributes. For example in paper A, the threshold method partitions the nucleus and cytoplasm based on the local minima of the pixel values in the images. Methods presented in paper B and C isolates the regions of interest by breaking the minimum spanning tree. Different from the developed methods, the supervised deep learning, methods depend on the annotated labels which are important for training the model.

The developed methods need to segment the medical images with accuracy that

is comparable to the accuracy of the state of the art methods. For each paper, the results were compared to the state of the art. For example in paper A, the results based on threshold were compared to state of the art results from different studies. Similarly, for papers B and C, the results were compared to state of the art results from different studies. The developed methods can be applied to different data sets. For example, in paper A, the method has been applied to data sets from two different studies. In paper B, the method has been applied to data sets from two different studies. In paper C, the method has been applied to data sets from three different studies. This shows that the developed methods can be applied on different data sets.

In this thesis, we proposed methods for segmenting the nucleus and cytoplasm from microscopic images, brain tumors from 2D MRI, and brain volume from 3D MRI. The methods have shown the ability to segment the regions of interest of the medical images at the level comparable to the state of the art methods from high-quality and poor-quality images. From the results obtained using the thresholding method, it can be observed that the method has the potential of being applied to other related data sets with high accuracy. The shortcoming of this method is that, whenever the cytoplasm color is indistinguishable from the Red Blood Cells and they touch each other, techniques for separating touching objects can be used. Also, for images with very low contrast between the cytoplasm and background, the method struggles to segment the cytoplasm. The method for segmenting the brain tumors shows that it is capable of segmenting other different data sets and it is a user-controlled method. Even though the method gives good results but the disadvantage of this method is user-guided. The results of 3D brain volume show that the method has the potential of being applied in the calculation of brain volume. The approach will be limited in the setting of incomplete 3D brain MRIs. So, poor quality of the data will compromise the results. An extension of the method for the 3D brain to a semi-automated method is required to alleviate this shortcoming.

This research aimed at developing methods for segmenting medical images. The research has shown that the developed methods can segment medical images at a level comparable to the state of the art methods. This might be a potential of having simple explainable and comparable methods for being adapted for solving problems in the medical field with high accuracy. However, the methods need to be tested on many other data sets to get a good generalization. This will improve or resolve the highlighted limitations of the developed methods.

Bibliography

[1] A. Acevedo, A. Merino, S. Alférez, Á. Molina, L. Boldú, and J. Rodellar. A dataset of microscopic peripheral blood cell images for development of automatic recognition systems. *Data in brief*, 30, 2020. 8, 13, 45, 48, 49, 52, 54, 58

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012. 11, 46, 48

[3] An Introduction to Image Segmentation: Deep Learning vs. Traditional. https://www.v7labs.com/blog/image-segmentation-guide#h2, Accessed: 05-07-2022. 12, 64

[4] S. Arora, J. Acharya, A. Verma, and P. K. Panigrahi. Multilevel thresholding for image segmentation through a fast statistical recursive algorithm. *Pattern Recognition Letters*, 29(2):119–125, 2008. 28

[5] B. Di Muzio and A. Murphy. Brain screen protocol (MRI). https://doi.org/10.53347/rID-37784, Accessed: 11-06-2022. 14

[6] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 42

[7] O. Boruvka. O jistm problmu minimlnim (about a certain minimal problem). *Prce mor. prrodoved. spol. v Brne,*, 3(3):37–58, 1926. 53

[8] Y. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*, volume 1, pages 105–112. IEEE, 2001. 52

[9] J. Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986. 44

[10] S. Chatterjee. Convergence of gradient descent for deep neural networks. *arXiv preprint arXiv:2203.16462*, 2022. 40

[11] Z. Chen, Z. Qi, F. Meng, L. Cui, and Y. Shi. Image segmentation via improving clustering algorithms with density and distance. *Procedia Computer Science*, 55:1015–1022, 2015. 46

[12] J. Cheng. Brain tumor dataset. figshare. dataset, 2017. 8

[13] K. Clark, B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, M. Pringle, et al. The cancer imaging archive (tcia): maintaining and operating a public information repository. *Journal of digital imaging*, 26(6):1045–1057, 2013. 15

[14] R. Clark and W. Miller. Computer based data analysis systems at argonne. *Methods in computational physics*, 5, 1966. 53

[15] D. Comaniciu and P. Meer. Cell image segmentation for diagnostic pathology. *Advanced algorithmic approaches to medical image segmentation*, pages 541–558, 2002. 11

[16] Comparing edge-based and region-based segmentation. `https://scikit-image.org/docs/stable/auto_examples/applications/plot_coins_segmentation.html`, Accessed: 13-06-2022. 59, 60

[17] Deep Learning vs. Machine Learning Whats The Difference? `https://levity.ai/blog`, Accessed: 05-07-2022. 64

[18] H. Delingette, M. Hebert, and K. Ikeuchi. Image segmentation and shape representation using deformable surfaces. In *IEEE Conference on Computer Vision and Pattern Recognition, ser. Geometric Methods in Computer Vision*, volume 1507. Citeseer, 1991. 49

[19] J. A. Delmerico, P. David, and J. J. Corso. Building facade detection, segmentation, and parameter estimation for mobile robot localization and guidance. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1632–1639. IEEE, 2011. 12

[20] L. Deng, D. Yu, et al. Deep learning: methods and applications. *Foundations and trends® in signal processing*, 7(3–4):197–387, 2014. 35

[21] I. Despotović, B. Goossens, and W. Philips. Mri segmentation of the human brain: challenges, methods, and applications. *Computational and mathematical methods in medicine*, 2015. 17

[22] N. Dhanachandra, K. Manglem, and Y. J. Chanu. Image segmentation using k-means clustering algorithm and subtractive clustering algorithm. *Procedia Computer Science*, 54:764–771, 2015. 47

[23] E. W. Dijkstra et al. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959. 53

[24] Downsampling and Upsampling in CNN. https://iq.opengenus.org/downsampling-and-upsampling-in-cnn/, Accessed: 24-04-2023. 38

[25] W. Doyle. Operations useful for similarity-invariant pattern recognition. *Journal of the ACM (JACM)*, 9(2):259–267, 1962. 29

[26] R. Easton. Basic principles of imaging science ii. *Chester F. Carlson Center for Imaging Science*, 2005. 18, 19, 24

[27] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181, 2004. 11, 27, 51, 53, 54

[28] R. Firdousi and S. Parveen. Local thresholding techniques in image binarization. *International Journal Of Engineering And Computer Science*, 3(3): 4062–4065, 2014. 27, 28

[29] E. W. Forgy. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *biometrics*, 21:768–769, 1965. 47

[30] E. B. George and M. Karnan. Mr brain image segmentation using bacteria foraging optimization algorithm. *International Journal of Engineering and Technology*, 4(5):295–301, 2012. 11

[31] C. A. Glasbey. An analysis of histogram-based thresholding algorithms. *CVGIP: Graphical models and image processing*, 55(6):532–537, 1993. 30

[32] C. A. Glasbey and G. W. Horgan. *Image analysis for the biological sciences*, volume 1. Wiley Chichester, 1995. 7, 8, 13, 44, 45

[33] R. C. Gonzalez. *Digital image processing*. Pearson education india, 2009. 7, 8, 12, 17, 18, 19, 20, 21, 23, 25, 27, 28, 44, 45, 46, 49

[34] D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society: Series B (Methodological)*, 51(2):271–279, 1989. 51

[35] R. Grosse. Lecture 5: Multilayer perceptrons. *inf. téc*, 2019. 34, 35

[36] Y. Guo and A. S. Ashour. Neutrosophic sets in dermoscopic medical image segmentation. In *Neutrosophic set in medical image analysis*, pages 229–243. Elsevier, 2019. 11

[37] R. M. Haralick and L. G. Shapiro. Image segmentation techniques. *Computer vision, graphics, and image processing*, 29(1):100–132, 1985. 45

[38] T. Heimann and H. Delingette. Model-based segmentation. In *Biomedical Image Processing*, pages 279–303. Springer, 2010. 49, 50

[39] R. Hemalatha, T. Thamizhvani, A. J. A. Dhivya, J. E. Joseph, B. Babu, and R. Chandrasekaran. Active contour based segmentation techniques for medical image analysis. *Medical and Biological Image Analysis*, 4(17):2, 2018. 17, 27

[40] J. Howard and S. Gugger. *Deep Learning for Coders with fastai and PyTorch*. O'Reilly Media, 2020. 34

[41] H. Huang, G. Yang, W. Zhang, X. Xu, W. Yang, W. Jiang, and X. Lai. A deep multi-task learning framework for brain tumor segmentation. *Frontiers in Oncology*, page 2095, 2021. 27

[42] L. Huang, J. Qin, Y. Zhou, F. Zhu, L. Liu, and L. Shao. Normalization techniques in training dnns: Methodology, analysis and application. *arXiv preprint arXiv:2009.12836*, 2020. 42

[43] L.-K. Huang and M.-J. J. Wang. Image thresholding by minimizing the measures of fuzziness. *Pattern recognition*, 28(1):41–51, 1995. 29

[44] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015. 42

[45] S. Jadon. A survey of loss functions for semantic segmentation. In *2020 IEEE conference on computational intelligence in bioinformatics and computational biology (CIBCB)*, pages 1–7. IEEE, 2020. 39

[46] R. Jain, R. Kasturi, B. G. Schunck, et al. *Machine vision*, volume 5. McGraw-hill New York, 1995. 23, 24, 25

[47] V. Jarnik. O jistm problému minimálním.(z dopisu panu o. borúvkovi). 1930. 53

[48] V. Jumb, M. Sohani, and A. Shrivas. Color image segmentation using k-means clustering and otsus adaptive thresholding. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 3(9):72–76, 2014. 46

[49] B.-Y. Kang, D.-W. Kim, and Q. Li. Spatial homogeneity-based fuzzy c-means algorithm for image segmentation. In *International Conference on Fuzzy Systems and Knowledge Discovery*, pages 462–469. Springer, 2005. 46

[50] J. N. Kapur, P. K. Sahoo, and A. K. Wong. A new method for gray-level picture thresholding using the entropy of the histogram. *Computer vision, graphics, and image processing*, 29(3):273–285, 1985. 29

[51] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019. 10

[52] J. Kittler and J. Illingworth. Minimum error thresholding. *Pattern recognition*, 19(1):41–47, 1986. 29

[53] D. T. Klaus. *Guide to Medical Image Analysis: Methods and Algorithms*. Springer London Limited, 2018. 11, 13

[54] S. KM and M. Prah. Data from brain-tumor-progression. *The Cancer Imaging Archive*, 2018. 15

[55] J. B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1): 48–50, 1956. 53

[56] R. Kumar, R. Srivastava, and S. Srivastava. Detection and classification of cancer from microscopic biopsy images using clinically significant and biologically interpretable features. *Journal of medical engineering*, 2015, 2015. 12

[57] M. Lazar and A. Hladnik. Implementation of global and local thresholding algorithms in image segmentation of coloured prints. In *35th international research conference IARIGAI*, volume 35, 2008. 27

[58] C. Li and P. K.-S. Tam. An iterative algorithm for minimum cross entropy thresholding. *Pattern recognition letters*, 19(8):771–776, 1998. 29

[59] C. H. Li and C. Lee. Minimum cross entropy thresholding. *Pattern recognition*, 26(4):617–625, 1993. 29

[60] P.-S. Liao, T.-S. Chen, P.-C. Chung, et al. A fast algorithm for multilevel thresholding. *J. Inf. Sci. Eng.*, 17(5):713–727, 2001. 11, 27, 28, 31

[61] X. Liu, L. Song, S. Liu, and Y. Zhang. A review of deep-learning-based medical image segmentation methods. *Sustainability*, 13(3):1224, 2021. 7, 9, 64

[62] Z. Liu, L. Wang, G. Hua, Q. Zhang, Z. Niu, Y. Wu, and N. Zheng. Joint video object discovery and segmentation by coupled dynamic markov networks. *IEEE Transactions on Image Processing*, 27(12):5840–5853, 2018. 12

[63] S. Lloyd. Least square quantization in pcm. bell telephone laboratories paper. published in journal much later: Lloyd, sp: Least squares quantization in pcm. *IEEE Trans. Inform. Theor.(1957/1982)*, 18:5, 1957. 47

[64] X. Long and J. Sun. Image segmentation based on the minimum spanning tree with a novel weight. *Optik*, 221:165308, 2020. 27, 51, 53

[65] B. Ma, A. Hero, J. Gorman, and O. Michel. Image registration with minimum spanning tree algorithm. In *Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101)*, volume 1, pages 481–484. IEEE, 2000. 53

[66] D. J. MacKay, D. J. Mac Kay, et al. *Information theory, inference and learning algorithms*. Cambridge university press, 2003. 47

[67] J. MacQueen. Classification and analysis of multivariate observations. In *5th Berkeley Symp. Math. Statist. Probability*, pages 281–297, 1967. 47

[68] Magnetic Resonance Imaging (MRI) of the Brain and Spine: Basics. https://case.edu/med/neurology/NR/MRI%20Basics.htm., Accessed: 10-06-2022. 14

[69] D. Mahapatra. Consensus based medical image segmentation using semi-supervised learning and graph cuts. *arXiv preprint arXiv:1612.02166*, 2016. 62

[70] D. Marr and E. Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 207(1167):187–217, 1980. 44

[71] S. Mayala and J. B. Haugsøen. Threshold estimation based on local minima for nucleus and cytoplasm segmentation. *BMC Medical Imaging*, 22(1):1–12, 2022. 11, 32, 33

[72] S. Mayala, I. V. Herdlevær, J. Bull Haugsøen, S. Anandan, S. Gavasso, and M. Brun. Brain tumor segmentation based on minimum spanning tree. *Frontiers in Signal Processing*, page 11. 11, 27, 51, 53, 56, 69

[73] S. Mayala, I. Herdlevær, J. B. Haugsøen, S. Anandan, N. Blaser, S. Gavasso, and M. Brun. Gubs: Graph-based unsupervised brain segmentation in mri images. *Journal of Imaging*, 8(10):262, 2022. 56, 69

[74] F. Milletari, N. Navab, and S.-A. Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. Ieee, 2016. 43, 44, 64

[75] H. Mittal, A. C. Pandey, M. Saraswat, S. Kumar, R. Pal, and G. Modwel. A comprehensive survey of image segmentation: clustering methods, performance parameters, and benchmark datasets. *Multimedia Tools and Applications*, pages 1–26, 2021. 7, 46, 47

[76] N. Moreau, C. Rousseau, C. Fourcade, G. Santini, A. Brennan, L. Ferrer, M. Lacombe, C. Guillerminet, M. Colombié, P. Jézéquel, et al. Automatic segmentation of metastatic breast cancer lesions on 18f-fdg pet/ct longitudinal acquisitions for treatment response assessment. *Cancers*, 14(1):101, 2021. 12

[77] F. Nazari and W. Yan. Convolutional versus dense neural networks: Comparing the two neural networks performance in predicting building operational energy use based on the building shape. *arXiv preprint arXiv:2108.12929*, 2021. 34

[78] W. Niblack. *An introduction to digital image processing*. Strandberg Publishing Company, 1985. 28

[79] L. Nieradzik, G. Scheuermann, D. Saur, and C. Gillmann. Effect of the output activation function on the probabilities and errors in medical image segmentation. *arXiv preprint arXiv:2109.00903*, 2021. 42

[80] N. Nikopoulos and I. Pitas. An efficient algorithm for 3d binary morphological transformations with 3d structuring elements of arbitrary size and shape. In *Proceedings of 1997 IEEE Workshop on Nonlinear Signal and Image Processing (NSIP97)*, 1997. 20, 21

[81] S. Niyas, S. Pawan, M. A. Kumar, and J. Rajan. Medical image segmentation using 3d convolutional neural networks: A review. *arXiv preprint arXiv:2108.08467*, 2021. 35, 36, 37, 38

[82] N. Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979. 28, 30

[83] N. OMahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. V. Hernandez, L. Krpalkova, D. Riordan, and J. Walsh. Deep learning vs. traditional computer vision. In *Advances in Computer Vision: Proceedings of the 2019 Computer Vision Conference (CVC), Volume 1 1*, pages 128–144. Springer, 2020. 35

[84] N. Padmasini, R. Umamaheswari, and M. Y. Sikkandar. State-of-the-art of level-set methods in segmentation and registration of spectral domain optical coherence tomographic retinal images. *Soft Computing Based Medical Image Analysis*, pages 163–181, 2018. 28

[85] A. Pepe, G. F. Trotta, C. Gsaxner, A. Brunetti, G. D. Cascarano, V. Bevilacqua, D. Shen, and J. Egger. Deep learning and generative adversarial networks in oral and maxillofacial surgery. In *Computer-Aided Oral and Maxillofacial Surgery*, pages 55–82. Elsevier, 2021. 11

[86] J. M. Prewitt and M. L. Mendelsohn. The analysis of cell images. *Annals of the New York Academy of Sciences*, 128(3):1035–1053, 1966. 30

[87] R. C. Prim. Shortest connection networks and some generalizations. *The Bell System Technical Journal*, 36(6):1389–1401, 1957. 53

[88] C. Pub. Technical report 15: 2004: Colorimetry. *Vienna: CIE Central Bureau*, 2004. 48

[89] S. Pyo, J. Lee, M. Cha, and H. Jang. Predictability of machine learning techniques to forecast the trends of market index prices: Hypothesis testing for the korean stock markets. *PloS one*, 12(11):e0188107, 2017. 34

[90] C. Rao and Y. Liu. Three-dimensional convolutional neural network (3d-cnn) for heterogeneous material homogenization. *Computational Materials Science*, 184:109850, 2020. 42

[91] T. Ridler, S. Calvard, et al. Picture thresholding using an iterative selection method. *IEEE trans syst Man Cybern*, 8(8):630–632, 1978. 30

[92] J. Rogowska. Overview and fundamentals of medical image segmentation. *Handbook of medical imaging, processing and analysis*, pages 69–85, 2000. 27

[93] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 43, 64

[94] A. Rosenfeld, T. Y. Kong, and A. Y. Wu. Digital surfaces. *CVGIP: Graphical Models and Image Processing*, 53(4):305–312, 1991. 18

[95] A. Saglam and N. A. Baykan. Sequential image segmentation based on minimum spanning tree representation. *Pattern Recognition Letters*, 87:155–162, 2017. 51, 53

[96] C. Sammut and G. I. Webb. *Encyclopedia of machine learning*. Springer Science & Business Media, 2011. 47

[97] K. Santosh, N. Das, and S. Ghosh. *Deep learning models for medical imaging*. Academic Press, 2021. 42

[98] C. Saueressig, A. Berkley, E. Kang, R. Munbodh, and R. Singh. Exploring graph-based neural networks for automatic brain tumor segmentation. In *International Symposium: From Data to Models and Back*, pages 18–37. Springer, 2020. 27

[99] J. Sauvola and M. Pietikäinen. Adaptive document image binarization. *Pattern recognition*, 33(2):225–236, 2000. 28

[100] scipy.sparse.csgraph.minimum spanning tree - Manual. https://openmedscience.com/medical-imaging/, Accessed: 30-05-2022. 12

[101] scipy.sparse.csgraph.minimum spanning tree - Manual. https://docs.scipy.org/doc/scipy/reference/generated/scipy.sparse.csgraph.minimum_spanning_tree.html, Accessed: 30-05-2022. 53

[102] R. Sedgewick and K. Wayne. *Algorithms: Part I*. Addison-Wesley Professional, 2014. 53

[103] J. Serra. Introduction to mathematical morphology. *Computer vision, graphics, and image processing*, 35(3):283–305, 1986. 20, 21, 23

[104] M. Sezgin and B. Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic imaging*, 13(1): 146–165, 2004. 29

[105] W. Shan, J. Guo, X. Mao, Y. Zhang, Y. Huang, S. Wang, Z. Li, X. Meng, P. Zhang, Z. Wu, et al. Automated identification of skull fractures with deep learning: a comparison between object detection and segmentation approach. *Frontiers in Neurology*, 12, 2021. 11

[106] L. G. Shapiro and C. George. Stockman–computer vision-pretice hall. *New Jersey*, 2001. 8, 9

[107] G. Sharp, K. D. Fritscher, V. Pekar, M. Peroni, N. Shusharina, H. Veeraraghavan, and J. Yang. Vision 20/20: perspectives on automated image segmentation for radiotherapy. *Medical physics*, 41(5):050902, 2014. 11

[108] J. Shen and S. Castan. An optimal linear operator for step edge detection. *CVGIP: Graphical models and image processing*, 54(2):112–133, 1992. 44

[109] J. Shi and J. Malik. Normalized cuts and image segmentation. computer vision and pattern recognition. In *IEEE Computer Society Conference on, 0*, volume 731, page 11, 1997. 27, 51

[110] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000. 27, 51, 52

[111] T. R. Singh, S. Roy, O. I. Singh, T. Sinam, K. Singh, et al. A new local adaptive thresholding technique in binarization. *arXiv preprint arXiv:1201.5227*, 2012. 27

[112] Slice Thickness and Interval Explained. `https://www.radtrain.com.au/post/copy-of-ct-slice-thickness-and-interval-explained`, Accessed: 11-06-2022. 14

[113] P. Sprawls. *Magnetic resonance imaging: principles, methods, and techniques*. Medical Physics Publishing Madison, 2000. 13, 14

[114] R. Srisha and A. Khan. Morphological operations for image processing: understanding and its applications. *NCVSComs-13*, 13:17–19, 2013. 19

[115] D. Stalling, M. Westerhoff, H.-C. Hege, et al. Amira: A highly interactive system for visual data analysis. *The visualization handbook*, 38:749–67, 2005. 9

[116] H. Steinhaus. Sur la division des corps matériels en parties: Bulletin de lacadémie polonaise des sciences. 1957. 47

[117] R.-Y. Sun. Optimization for deep learning: An overview. *Journal of the Operations Research Society of China*, 8(2):249–294, 2020. 40, 41, 42

[118] W.-H. Tsai. Moment-preserving thresolding: A new approach. *Computer vision, graphics, and image processing*, 29(3):377–393, 1985. 29

[119] J. K. Udupa, V. R. LaBlanc, H. Schmidt, C. Imielinska, P. K. Saha, G. J. Grevera, Y. Zhuge, L. M. Currie, P. Molholt, and Y. Jin. Methodology for evaluating image-segmentation algorithms. In *Medical Imaging 2002: Image Processing*, volume 4684, pages 266–277. SPIE, 2002. 8, 62

[120] J. Vergés Llahí. State-of-the-art survey on color segmentation methods. 2005. 28, 45

[121] X. Wang, Y. Liu, and H. Xin. Bond strength prediction of concrete-encased steel structures using hybrid machine learning method. In *Structures*, volume 32, pages 2279–2292. Elsevier, 2021. 35

[122] Z. Wang. Deep learning for image segmentation: veritable or overhyped? *arXiv preprint arXiv:1904.08483*, 2019. 65

[123] J. A. Webb. Global image processing operations on parallel architectures. In *Real-Time Image Processing II*, volume 1295, pages 176–187. SPIE, 1990. 19

[124] What is the difference between slice thickness and slice increment? https://www.materialise.com/en/faq/what-difference-between-slice-thickness-and-slice-increment, Accessed: 11-06-2022. 14

[125] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht. The marginal value of adaptive gradient methods in machine learning. *Advances in neural information processing systems*, 30, 2017. 42

[126] W. Wu, A. Y. Chen, L. Zhao, and J. J. Corso. Brain tumor detection and segmentation in a crf (conditional random fields) framework with pixel-pairwise affinity and superpixel-level features. *International journal of computer assisted radiology and surgery*, 9(2):241–253, 2014. 11

[127] Y. Wu and K. He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 42

[128] B. Xiang. *Knowledge-based image segmentation using sparse shape priors and high-order MRFs*. PhD thesis, Châtenay-Malabry, Ecole centrale de Paris, 2013. 52

[129] Y. Xu and E. C. Uberbacher. 2d image segmentation using minimum spanning trees. *Image and Vision Computing*, 15(1):47–57, 1997. 51, 53

[130] R. Z. Ye, C. Noll, G. Richard, M. Lepage, É. E. Turcotte, and A. C. Carpentier. Deepimagetranslator: A free, user-friendly graphical interface for image translation using deep-learning and its applications in 3d ct image analysis. *SLAS technology*, 27(1):76–84, 2022. 11

[131] J.-C. Yen, F.-J. Chang, and S. Chang. A new criterion for automatic multilevel thresholding. *IEEE Transactions on Image Processing*, 4(3):370–378, 1995. 29

[132] J. Yi, P. Wu, M. Jiang, Q. Huang, D. J. Hoeppner, and D. N. Metaxas. Attentive neural cell instance segmentation. *Medical image analysis*, 55:228–240, 2019. 10

[133] X. Yin, Y. Li, and B.-S. Shin. Tgv upsampling: a making-up operation for semantic segmentation. *Computational Intelligence and Neuroscience*, 2019, 2019. 38

[134] X. Ying, N. J. Barlow, and A. Tatiparthi. Micro-ct and volumetric imaging in developmental toxicology. In *Reproductive and developmental toxicology*, pages 1261–1285. Elsevier, 2022. 12

[135] I. T. Young, J. J. Gerbrands, and L. J. Van Vliet. Fundamentals of image processing. 1998. 17, 19

[136] G. W. Zack, W. E. Rogers, and S. A. Latt. Automatic measurement of sister chromatid exchange frequency. *Journal of Histochemistry & Cytochemistry*, 25(7):741–753, 1977. 29

[137] C. T. Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on computers*, 100(1):68–86, 1971. 51, 53, 54

[138] C. Zeng, W. Jia, X. He, and M. Xu. Recent advances on graph-based image segmentation techniques. *Graph-Based Methods in Computer Vision: Developments and Applications*, pages 140–154, 2013. 51

[139] D. Zhang, Y. Song, D. Liu, H. Jia, S. Liu, Y. Xia, H. Huang, and W. Cai. Panoptic segmentation with an end-to-end cell r-cnn for pathology image analysis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 237–244. Springer, 2018. 10

Part II

Included papers

Paper A

Threshold estimation based on local minima for nucleus and cytoplasm segmentation.

Simeon Mayala, Jonas Bull Haugsøen

BMC Medical Imaging, 22:77, (2022)

Mayala and Haugsøen *BMC Medical Imaging* (2022) 22:77
https://doi.org/10.1186/s12880-022-00801-w

BMC Medical Imaging

RESEARCH

Open Access

Threshold estimation based on local minima for nucleus and cytoplasm segmentation

Simeon Mayala[1*] and Jonas Bull Haugsøen[2,3]

Abstract

Background: Image segmentation is the process of partitioning an image into separate objects or regions. It is an essential step in image processing to segment the regions of interest for further processing. We propose a method for segmenting the nuclei and cytoplasms from white blood cells (WBCs).

Methods: Initially, the method computes an initial value based on the minimum and maximum values of the input image. Then, a histogram of the input image is computed and approximated to obtain function values. The method searches for the first local maximum and local minimum from the approximated function values in the order of increasing of knots sequence. We approximate the required threshold from the first local minimum and the computed initial value based on defined conditions. The threshold is applied to the input image to binarize it, and then post-processing is performed to obtain the final segmented nucleus. We segment the whole WBC before segmenting the cytoplasm depending on the complexity of the objects in the image. For WBCs that are well separated from red blood cells (RBCs), *n* thresholds are generated and then produce *n* thresholded images. Then, a standard Otsu method is used to binarize the average of the produced images. Morphological operations are applied on the binarized image, and then a single-pixel point from the segmented nucleus is used to segment the WBC. For images in which RBCs touch the WBCs, we segment the whole WBC using SLIC and watershed methods. The cytoplasm is obtained by subtracting the segmented nucleus from the segmented WBC.

Results: The method is tested on two different public data sets and the results are compared to the state of art methods. The performance analysis shows that the proposed method segments the nucleus and cytoplasm well.

Conclusion: We propose a method for nucleus and cytoplasm segmentation based on the local minima of the approximated function values from the image's histogram. The method has demonstrated its utility in segmenting nuclei, WBCs, and cytoplasm, and the results are satisfactory.

Keywords: Segmentation, Local minima, Nucleus and cytoplasm

Background

White blood cells (WBCs) are cells of the immune system that take part in the body's defense against infectious disease and foreign material [1, 2]. WBCs can be categorized based on structure (granulocytes or agranulocytes) and cell lineage (myeloid or lymphoid cells). Broadly, there are five types of WBCs; three types of granulocytes—neutrophils, eosinophils, and basophils, and two types of agranulocytes—lymphocytes and monocytes. Granulocytes and monocytes are of myeloid lineage, whereas lymphocytes are, as the name implies, of lymphoid lineage.

In microscopic images of stained blood smears, WBCs can be differentiated from red blood cells and platelets by having nuclei and their (in most cases) larger size. They also stain darker using common dyes such as hematoxylin and eosin (H&E). Three main characteristics are used

*Correspondence: simeon.mayala@uib.no
[1] Department of Mathematics, University of Bergen, Allégaten 41, 5007 Bergen, Norway
Full list of author information is available at the end of the article

© The Author(s) 2022. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/. The Creative Commons Public Domain Dedication waiver (http://creativecommons.org/publicdomain/zero/1.0/) applies to the data made available in this article, unless otherwise stated in a credit line to the data.

to identify the different types of WBCs - the shape of their nuclei, their granularity, and their staining.

All the granulocytes are large, granular cells with lobulated nuclei [2]. Neutrophils stain neutrally and have nuclei with multiple (2–5) lobes. Eosinophils stain red and have nuclei with 2–4 lobes. Basophils stain blue and have nuclei with 2–3 lobes. The shapes of these lobes are characteristic for each cell type, as seen in Fig. 1. Monocytes are large agranular cells with kidney-shaped nuclei. Lymphocytes are also agranular and are smaller than the other WBCs. Their nuclei are round, often eccentric, and stain dark blue. Examples of the different types of WBCs can be seen in Fig. 1.

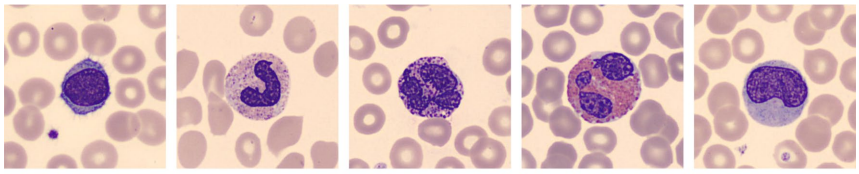
Image segmentation is a process that separates a region of interest (ROI) from the background to simplify further analysis [3, 4]. Different methods for segmenting images have been developed. They are classified differently based on how they perform the segmentation. Example of the popular methods include thresholding [5–9], edge detection [10], morphologically based [11], graph based [12–14], clustering based [15], watershed [16], level-set based [17] and a combination of different methods [2, 18–20]. Convolutional neural networks (CNN) are deep learning techniques and they have been widely used for WBCs segmentation [21–23]. We review related literature concerning WBC and nucleus segmentation.

Mittal et al. [24] presents a comprehensive review of different computer-aided methods for analyzing blood smear images and leukemia detection. The paper reviews 149 papers by presenting different techniques used for preprocessing images and WBC segmentation methods. It provides different workflow pipelines for segmenting WBC based on knowledge, deformable models, and machine learning. Also, the review gives the merits and demerits of each method.

Li et al. [9] proposed a dual-threshold method for segmenting WBC based on a strategic combination of the RGB and HSV colour spaces by searching for an optimal threshold using a golden section search method. Ghane et al. [18], proposed a method for segmenting WBC based on a combination of thresholding, K-means clustering and a modified watershed algorithm. Also, Kuse et al. [19] proposed a method that segments cells using mean shift-based clustering for color approximation and then thresholding. Features are extracted and then used to train a support vector machine (SVM) classifier for classifying lymphocytes and non-lymphocytes. Prinyakupt et al. [2] proposed a system that pre-processes the images by locating the WBCs and then segmenting them into nucleus and cytoplasm.

Theera-Umpon [11] proposed a method that uses a fuzzy C-means (FCM) algorithm and mathematical morphological operations to segment WBCs. Miao and Xiao [16] proposed a marker-controlled watershed algorithm for segmenting WBCs and RBCs. Yung-Kuan et al. [10] propose a method for WBC nucleus segmentation and counting the lobes in a nucleus that works by object contour detection. Furthermore, Salem [15] proposed a WBC segmentation method based on the K-means clustering technique. The method converts RGB to the $L*a*b$ color space and then the a and b components are used as features in the clustering algorithm. Sadeghian et al. [20] propose a framework that integrates several digital image processing algorithms for segmenting nucleus and cytoplasm. Khamael et al. [17] propose a method for the segmenting nuclei of WBC from the cytoplasm and the cell wall. The method performs segmentation based on level set methods and geometric active contours.

Banik et al. [21] proposed a method that segments WBC nuclei based on a $L*a*b$ color space conversion and K-means algorithm. Then, WBCs are located using the segmented nucleus. A convolutional neural network (CNN) is used to classify the localized WBC image. Fan et al. [22] proposed a method for localization and segmentation of WBCs. The method uses pixel-level information for training a deep convolutional neural network to locate and segment the region of interest. Lu et al. [23] proposed a WBC-NET based on the UNet++ and ResNet to improve the accuracy of the



(a) Lymphocyte (b) Neutrophil (c) Basophil (d) Eosinophil (e) Monocyte

Fig. 1 Normal white blood cells. The cells seen in the background are red blood cells identified by their red color, thin (often almost transparent) center, and lack of nuclei

WBCs segmentation. Furthermore, Long [25] proposed an enhanced, light-weighted U-Net with a modified encoded branch. The method explores the possibility of performance improvement of cell nucleus segmentation algorithms through deep learning, requiring less pre-and post-processing of images.

Our contribution in this work concerns the way the method estimates the threshold. A histogram of the input image is computed, and the function values are approximated. The threshold is estimated based on the local minima of the approximated function values. The estimated threshold is applied to the input image to segment the nucleus. Also, we develop a simple strategy for segmenting the WBC whenever is well separated from surrounding red blood cells. We generated n different thresholds and each threshold is applied to the input image to produce n thresholded images. The produced images are combined by taking their average and then a standard Otsu method is used to binarize it. We perform post-processing and use a single-pixel point from the nucleus to extract the WBC. The cytoplasm is obtained by subtracting the segmented nucleus from the segmented WBCs.

For images whose WBCs touch RBCs, we opt for classical techniques to separate the touching objects. We use the Simple Linear Iterative Clustering (SLIC) approach based on superpixels [26]. Since the focus is on the WBCs, we utilize the superpixel's strength of boundary adherence to segment the WBCs. We also apply a watershed transformation to segment the WBC [27]. The number of local maxima is chosen automatically. When the WBC is detached from the uninteresting objects, we perform post-processing so that only the WBC remains.

Methods

In this section, we establish a method for segmenting the nucleus and cytoplasm based on the distribution of the intensity values of the input image. The technique estimates the threshold automatically based on the local minima of the estimated function values. The summarized steps for nucleus, WBC, and cytoplasm segmentation are visualized in Fig. 2.

Threshold estimation for nucleus segmentation

Assuming that a gray-scale image has intensity values that can be classified into several gray levels. Let a gray-scale image A be a matrix of $M \times N$ dimension such that $f(x, y)$ gives an intensity value at position (x, y). We define,

$$A = \left[f(x,y)\right]_{M\times N} \text{ for } x = 0, 1, \ldots N-1; y = 0, 1, \ldots M-1 \quad (1)$$

Considering the input image A in Eq. 1, we compute a value T_{nc0} that will be used to search for a threshold value for segmenting the nucleus. The T_{nc0} value is computed from the input image by

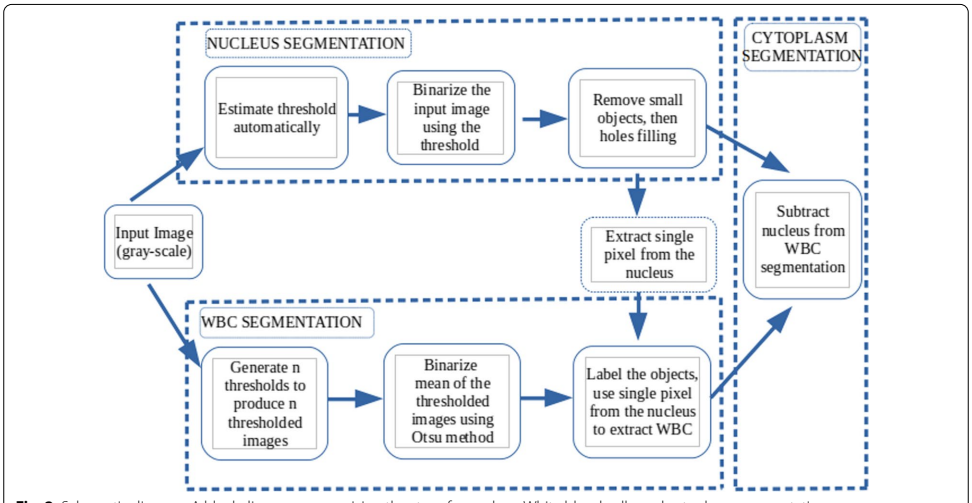


Fig. 2 Schematic diagram. A block diagram summarizing the steps for nucleus, White blood cells, and cytoplasm segmentation

$$T_{nc0}(n_1, n_2) = \frac{Max(A)}{n_1} + \frac{Min(A)}{n_2} \tag{2}$$

Where n_1 and n_2 are integers greater than 0. A simple analysis on how to choose n_1 and n_2 is presented in Additional file 1. Assuming that T_{nc0} is an intensity value on the histogram of the input image A. We want to estimate T_{nc0} such that $Min(A) < T_{nc0} < Max(A)$. We represent the input image A in a form that increases monotonically to locate the value T_{nc0}. Let $\tilde{A}$ be a vectorized matrix of A, then the intensity values $f(x, y)$ at each position (x, y) in the vectorized matrix are

$$\tilde{A} = \left[f(x,y)_0, f(x,y)_1, f(x,y)_2, \ldots, f(x,y)_{k-2}, f(x,y)_{k-1}\right]^T \tag{3}$$

Where k is the total number of pixels in the image A and T is a transpose. We construct a non-decreasing sequence of values in $\tilde{A}$ by using the intensity values $f(x, y)$ at each position in $\tilde{A}$. Then, the sequence will be

$$\tilde{A}_{seq} = \left[f(x,y)_i \le f(x,y)_{i+1} \le \cdots \le f(x,y)_{k-2} \le f(x,y)_{k-1}\right] \tag{4}$$

The constructed sequence of intensity values in $\tilde{A}_{seq}$ are ordered sets and preserve a non-decreasing order, so it takes the form of a monotonically increasing function. For each $f(x,y) \in \tilde{A}_{seq}$ we can write $f(x,y)_i \to I_i$. The form has been changed from the function of two variables $f(x, y)$ in image A into the function of one variable in $\tilde{A}_{seq}$

$$\tilde{A}_{seq} = \left[I_i \le I_{i+1} \le \cdots \le I_{k-2} \le I_{k-1}\right] = \left[I_i \ldots, T_{nc0}, \ldots, I_{k-1}\right] \tag{5}$$

T_{nc0} is not necessarily one of the values in $\tilde{A}_{seq}$ but it is within the range of $\tilde{A}_{seq}$. A threshold $\tilde{\varepsilon}_t$ will be estimated after representing the intensity values of the input image in the form of a histogram. T_{nc0} together with some conditions will be used to approximate the required threshold value.

Approximating threshold $\tilde{\varepsilon}_t$

Let $h(x)$ be the function representing the histogram of an image A, where x is an intensity value I, and $h(x)$ is the frequency of the intensity values (see Fig. 3). Let $\tilde{h}(x)$ be the approximation of $h(x)$ (see Fig. 4). We approximate $h(x)$ without distorting the general tendency of the histogram function. We can decompose $\tilde{h}(x)$ into

$$\tilde{h}(x) = \tilde{h_1}(x) + \tilde{h_2}(x) \tag{6}$$

Let $\left(x_i, \tilde{h_1}(x_i)\right)_{i=1}^{m}$ be the given points from the approximated histogram. We need to compute a spline g such that

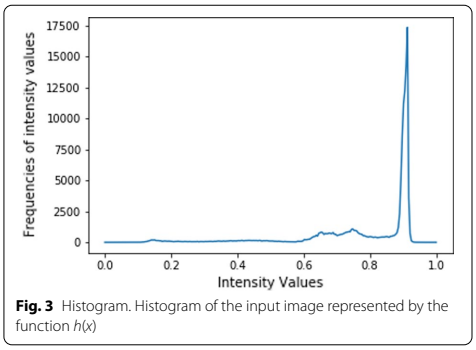


Fig. 3 Histogram. Histogram of the input image represented by the function $h(x)$

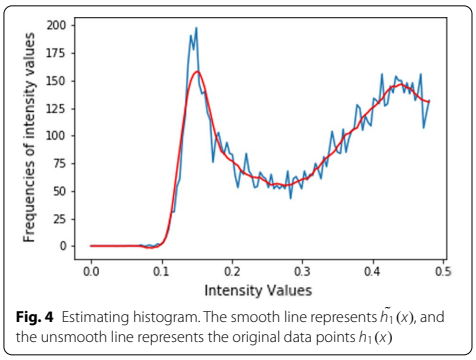


Fig. 4 Estimating histogram. The smooth line represents $\tilde{h_1}(x)$, and the unsmooth line represents the original data points $h_1(x)$

$$g(x_i) = \tilde{h_1}(x_i), \text{ for } i = 1, \ldots, m \tag{7}$$

The spline concept and basis spline (B-spline) is presented in Additional file 1 [28, 29]. Assuming that the spline is smooth and we are interested in the number of turning points produced from the spline $g(x)$. We utilize the local propagation of B-Spline to choose any degree we want. So, a $g(x)$ polynomial of n degrees gives $n-1$ turning points. Setting d be the degree of $g(x)$, the turning points will be $\left(x_1, g(x_1)\right), \left(x_2, g(x_2)\right), \ldots, \left(x_{d-1}, g(x_{d-1})\right)$.

Let $(x_c, g(x_c))$, $(x_{c+1}, g(x_{c+1}))$ be first local maximum and minimum of the function $g(x)$ respectively. Based on the filtered data points of Fig. 5, the threshold $\tilde{\varepsilon}_t$ is expected to be closer to the first local minimum of the function $g(x)$. Also, the threshold $\tilde{\varepsilon}_t$ must appear after the first local maximum of the function $g(x)$ in the order of increase of the knots.

Also, from Eq. 2, the sensitivity analysis of T_{nc0} in the Additional file 1 guarantees that the value of T_{nc0} will not

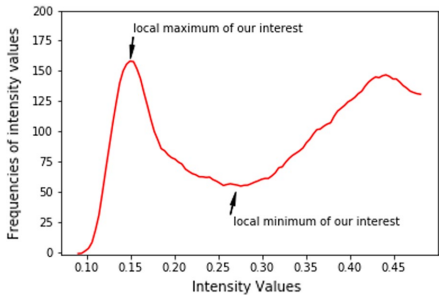


Fig. 5 Estimated histogram. Figure showing the approximation $\tilde{h_1}(x)$ after removing the long flat tail. It shows the turning point (local minimum and maximum of interest)

be very far from $\frac{1}{3}$. We introduce a parameter Er which is a small value chosen to control T_{cn0} and the local minimum value during threshold estimation. Then, the threshold $\tilde{\varepsilon}_t$ is computed by checking the following conditions:

(a) $x_c < T_{nc0}$.

(b) If $|T_{nc0} - x_{c+1}| < Er$ then, $\tilde{\varepsilon}_t = \dfrac{T_{nc0} + x_{c+1}}{2}$.

(c) If $|T_{nc0} - x_{c+1}| > Er$ and $T_{nc0} > x_{c+1}$ then, $\tilde{\varepsilon}_t = \dfrac{T_{nc0} + x_{c+1} + Er}{2}$.

(d) If $|T_{nc0} - x_{c+1}| > Er$ and $T_{nc0} < x_{c+1}$ then, $\tilde{\varepsilon}_t = \dfrac{T_{nc0} + x_{c+1} - Er}{2}$.

The threshold $\tilde{\varepsilon}_t$ is applied on the input image A to produce image A_{recons}. We call it a reconstructed image, and it is mathematically obtained by

$$A_{recons} = \begin{cases} Min(A) \text{ if} f(x,y) \leq \tilde{\varepsilon}_t \\ f(x,y) \text{ elsewhere} \end{cases}$$

The image A_{recons} is binarized to segment the nucleus of the WBCs. Then, post-processing operations are applied to the binarized image to obtain the final segmented nuc leus.

Threshold estimation for WBC segmentation

We develop a strategy for segmenting the WBC after preprocessing the input image. The input image is preprocessed by averaging n thresholded images of the input image to reduce the variability of intensity values in the WBC region. The n images are obtained using different thresholds produced in a defined range by using

$$T^i_{wbc} = \frac{Max(A) + T_{nc0}}{2} + er_i \tag{8}$$

where er_i is a step size between consecutive thresholds in a defined interval, $i = 0, 1, 2, \ldots, n-1$.

Let a_l and a_u be lower and upper limit of the intensity values in the chosen interval respectively. We denote er_i by $\bigtriangleup h$. Then, $\bigtriangleup h$ is a step size between two consecutive thresholds T^i_{wbc} and T^{i+1}_{wbc} in the interval between a_l and a_u. To generate n equal sub-intervals in the defined interval we use

$$n = \frac{a_u - a_l}{\bigtriangleup h} \tag{9}$$

as $\bigtriangleup h \to 0$, $n \to \infty$ and $(a_u - a_l)$ is fixed. T^i_{wbc} will be generated at the interval $\bigtriangleup h$. So, for the interval $a_l \leq er_i \leq a_u$ divided into $n-1$ equal subintervals is represented as

$$a_l = T^0_{wbc} < T^1_{wbc} < T^2_{wbc} < \cdots < T^{n-1}_{wbc} = a_u \tag{10}$$

where $T^i_{wbc} = a_l + i \bigtriangleup h$, for $i = 0, 1, 2, \ldots, n-1$ and $\bigtriangleup h = \frac{a_u - a_l}{n}$. Each T^i_{wbc} is used to produce one thresholded image from the input image A by

$$A_j = \begin{cases} T^i_{wbc} \text{ if} f(x,y) \leq T^i_{wbc} \\ f(x,y) \text{ elsewhere} \end{cases} \tag{11}$$

Since i varies from 0 to $n-1$, then the reconstructed (preprocessed) image for WBC is produced by

$$A_{wbc} = \frac{1}{n} \sum_{j=0}^{n-1} A_j \tag{12}$$

The image A_{wbc} reduces the contrast in the WBCs region and the plasma membrane of the RBCs. The image A_{wbc} is binarized to segment the WBCs from RBCs and other background. Note that the strategy works well for images whose WBCs are well separated from RBCs as well as the choice of n. For images whose WBCs touch the RBCs, we opt for classical techniques to segment the WBCs. The cytoplasm is obtained by subtracting the segmented nucleus from the segmented WBCs.

Descriptions of data

The proposed method is tested on two different WBC image datasets. The first image dataset contains a total of 17,092 images of individual normal cells, which were acquired using the analyzer CellaVision DM96 in the Core Laboratory at the Hospital Clinic of Barcelona [30]. The dataset is organized into eight different groups. It includes neutrophils, eosinophils, basophils, lymphocytes, monocytes, immature granulocytes, erythroblasts, and platelets (thrombocytes). It is a high-quality labeled dataset that can be used for benchmarking, training, and testing models. They are JPG images with a size of (360×363) pixels for each image. For more information refer to [30].

The second image dataset used in [31] contains 300 images together with the manually segmented images. The dataset was originally obtained from iangxi Tecom Science Corporation, China. The nuclei, cytoplasms, and background including red blood cells are marked in white, gray, and black respectively. The images were acquired using a Motic Moticam Pro 252A optical microscope camera with an N800-D motorized autofocus microscope, and the blood smears were processed with a newly-developed hematology reagent for rapid WBC staining [31]. They are 120×120 images of WBCs.

Implementation and experimental results

Nucleus segmentation

The input images are converted into grayscale. For each grayscale image, a value T_{nc0} is computed using Eq. 4(a) in Additional file 1. Then, a *NumPy function histogram* is used for computing a histogram of the input image [32]. The histogram is decomposed into two equal parts and then run a script on the first part of decomposition to check and remove the long flat tail at the beginning (see Figs. 4 and 5). Then, a SciPy function *savgol filter* is applied to obtain the approximated function values [33]. A *B-spline* (3^{rd} order polynomial) interpolation function is applied on the approximated function values and then the *argrelextrema* function is used to compute the local minima and maxima [33]. Note that if the local minima and maxima are not found in the first run, the length of decomposition is incremented, and start searching again until the first local maximum and minimum are found. The values of the first local maximum and minimum together with the value T_{nc0} are used for computing the threshold $\tilde{\varepsilon}_t$ based on the conditions given in the methodology subsections. The threshold $\tilde{\varepsilon}_t$ is applied on the input image to preprocess and binarize it to segment the nucleus. The binarized image is post-processed by applying a morphological operation to remove small objects and close holes if they exist. The result is the segmented nucleus of the input image. Some results for the implementation steps are visualized in Fig. 6.

WBC and cytoplasm segmentation

We generate n values automatically in the interval $[Min(A), \frac{Max(A)}{3}]$ which are referred as step size er_i in Eq. 8. For each value in the interval we compute threshold T^i_{wbc} using Eq. 8. Each threshold is applied to the input image to generate n thresholded images using Eq. 11. The n thresholded images are averaged using Eq. 12 to obtain a preprocessed image A_{wbc} and then binarize it using a standard Otsu thresholding method [34]. Then, morphological operations are applied to separate the WBC from the background and RBCs. From the segmented nucleus (in the previous sub-section), we take a single pixel value location and use it to extract the whole WBC from the binarized image. The cytoplasm is obtained by subtracting the segmented nucleus from the WBC. Figure 7 visualizes some images summarizing the steps for nucleus and WBC segmentation.

For images whose WBCs touch the RBCs we opt for two classical approaches and use them to segment the WBCs.

We apply the Simple Linear Iterative Clustering (SLIC) approach based on superpixel [26, 35]. Since the superpixel algorithm adheres to boundaries, we focus on identifying the WBCs boundaries whenever the RBCs touch the WBCs. We allow under segmentation in order to segment well the WBCs. The results are visualized in Fig. 8 fifth column.

We also apply watershed transformation for segmenting the WBCs when the RBCs touch the WBCs. The approach is a transformation on grayscale images [27]. It aims at detaching the ROI from the non interesting objects when the edges of these objects touch each other. We concentrate on the marker construction step where it is easier to control it. We allow an automatic choice of the number of local maxima and retain the results if the RBCs are detached from the WBCs. In this step, we do not care much about the over-segmentation of the ROI, instead, we focus on detaching the uninteresting objects from the object of interest (the WBC). The results are visualized in Fig. 8 third column.

Note that the input for watershed can be either image gradient or binarized image. Both options work but the challenging part is to control the markers. Looking at Fig. 8 third column it could be challenging to identify the required labels and then remove the unwanted ones. Figure 8 fifth column shows that SLIC is straightforward because it identifies the boundaries of the WBCs.

Quantitative analysis

We present a simple analysis to assess the efficiency and effectiveness of the proposed method. The performance evaluation is done by a quantitative measure of the segmentation results. We analyze the accuracy by evaluating the similarity between the segmentation results obtained using the proposed method and the ground truth.

We compute Jaccard Index (JI), Dice similarity coefficients (DSC), Sensitivity, Specificity, and Precision to check the level of similarity between the ground truth and the segmentation from the proposed method (predicted segmentation). The Jaccard Index (*JI*) is defined by

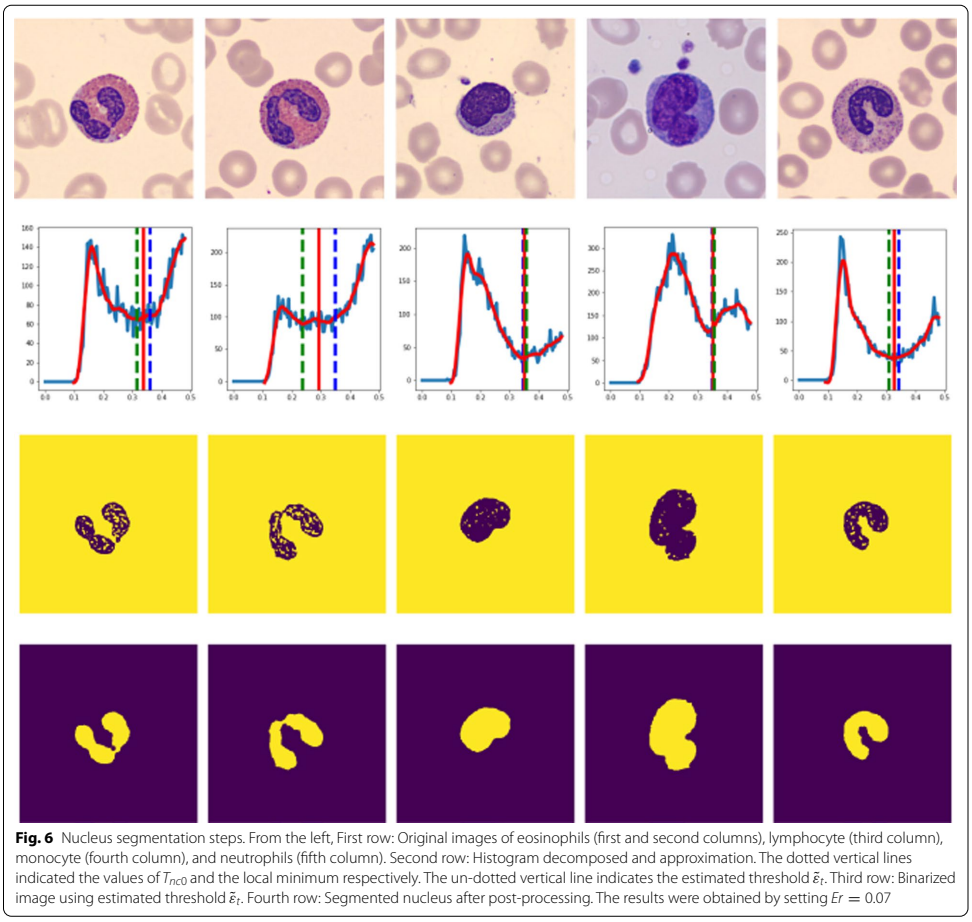


Fig. 6 Nucleus segmentation steps. From the left, First row: Original images of eosinophils (first and second columns), lymphocyte (third column), monocyte (fourth column), and neutrophils (fifth column). Second row: Histogram decomposed and approximation. The dotted vertical lines indicated the values of T_{nc0} and the local minimum respectively. The un-dotted vertical line indicates the estimated threshold $\tilde{\varepsilon}_t$. Third row: Binarized image using estimated threshold $\tilde{\varepsilon}_t$. Fourth row: Segmented nucleus after post-processing. The results were obtained by setting $Er = 0.07$

$$JI(L_M, L_P) = \frac{|L_M(x,y) \cap L_P(x,y)|}{|L_M(x,y) \cup L_P(x,y)|} = \frac{TP}{TP + FP + FN} \tag{13}$$

where $|L_M(x,y)|$ is the number of labels in the ROI from the ground truth. $|L_P(x,y)|$ is the number of labels in ROI obtained using the proposed method. $|L_M(x,y) \cap L_P(x,y)|$ is the number of labels appearing in the ROI from the ground truth and the predicted segmentation. The Dice Similarity Coefficient is defined by

$$DSC(L_M, L_P) = \frac{2|L_M(x,y) \cap L_P(x,y)|}{|L_M(x,y)| + |L_P(x,y)|} = \frac{2TP}{2TP + FP + FN} \tag{14}$$

We can also use the concept of true positive (TP), false positive (FP), true negative (TN), and false-negative (FN) to check the performance of the method. We compute Sensitivity, Specificity, and Precision as follows.

$$\begin{aligned} Sensitivity &= \frac{TP}{TP + FN}, \; Specificity \\ &= \frac{TN}{TN + FP}, \; \text{Precision} = \frac{TP}{TP + FP} \end{aligned} \tag{15}$$

In the quantitative analysis we use the second data set used in [31] because the first data set in [30] does not have the ground truth images. In the analysis, some

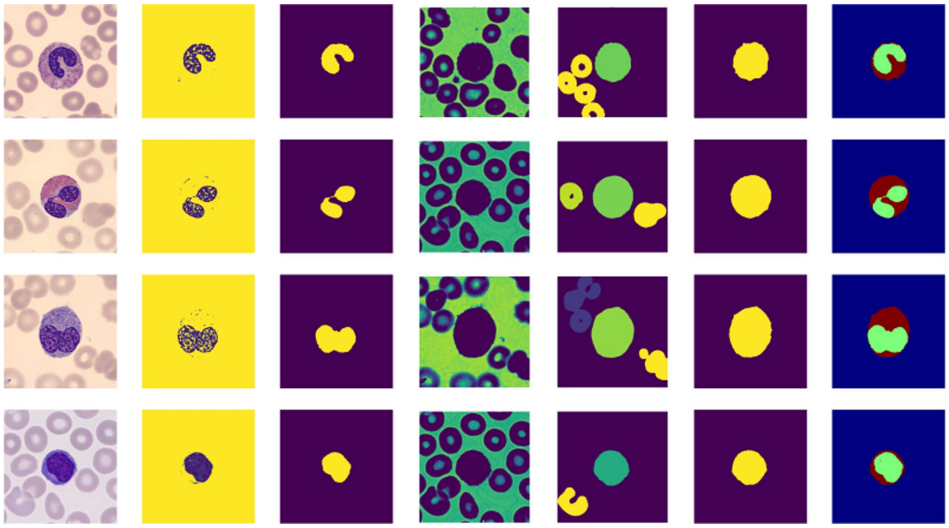


Fig. 7 WBC and cytoplasm segmentation steps. From the left, First column: input images, Second column: Reconstructed images and then binarized using estimated threshold $\tilde{\varepsilon}_t$ for nucleus segmentation. Third column: Segmented nucleus after post-processing. Fourth column: Reconstructed images after averaging different thresholded images. Fifth column: Assigning new labels to the objects and removing some objects. Sixth column: Segmented WBC using a single-pixel value location from the segmented nucleus. Seventh column: Segmented nucleus and cytoplasm. From top to down: the first row is the neutrophil, the second row is the eosinophil, the third row is the monocyte, the fourth row is the lymphocytes

images were not included because of different reasons. Some of the images not considered in this section include the ones visualized in Fig. 9 showing the original and the ground truth,

A quantitative evaluation was performed by comparing the level of similarity between the ground truth and the segmentation obtained using the proposed method. The JI, DSC, Sensitivity, Specificity, and precision metrics range from 0 to 1. Zero indicates that there is no overlap between the predicted segmentation and the ground truth whereas 1 indicates a perfect overlap between predicted segmentation and the ground truth.

Also, we augmented 600 (masks included) images to generate 1860 images from the second data set. The data set for training, validation, and testing were divided in the ratio of 7:2:1 respectively. We use the U-net model for biomedical image segmentation presented in [36]. All the implementation were performed using Keras [37]. To see the setting of the hyperparameters, performance training of the model, and visualization of the predicted masks see Additional file 1. The prediction was performed using the testing data set (not included in the training). The predicted results are included in Tables 2 and 3. Other results obtained in other papers were also included for comparison.

We also present graphs in Figs. 10 and 11 to show the correlation of the size of the segmented area (nucleus and WBC) obtained using the proposed method and the ground truth.

The implementation of the proposed method was done by writing scripts in the python programming language, and it was run on a PC processor (Core i7-8650U CPU @ 1.90GHz $\times$ 8). To process 40 images of 360×363 size, it takes 13.06518197059 seconds, which is approximately 0.326629549 seconds are used to process one image.

Discussion and conclusion

We propose a method for segmenting the nucleus and cytoplasm of the WBCs based on the local minima. The method estimates the threshold automatically from the input image by checking different conditions that allow a wide range of searching for a good approximation. The threshold is applied to the input image to segment the nucleus. The WBC is segmented and then the cytoplasm

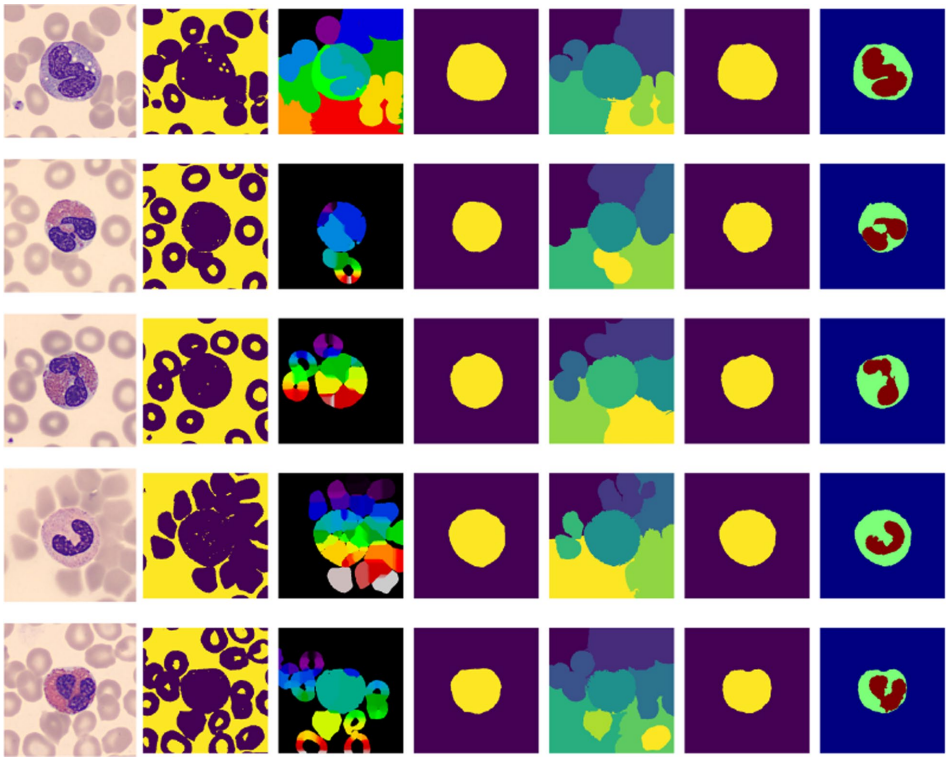


Fig. 8 WBC segmentation using SLIC and marker-controlled watershed. From the left, First column: input images, Second column: converted into grayscale and then binarized using the Otsu thresholding method. Third column: marker-controlled watershed segmentation using a distance transformation. Fourth column: Segmented WBC after post-processing the watershed results. Fifth column: segmentation of the input image using SLIC. Sixth column: Segmented WBC from SLIC results. Seventh column: Segmented nucleus and cytoplasm

is obtained by subtracting the segmented nucleus from the segmented WBC. The segmentation result is compared to ground truth to check the level of accuracy of the proposed method.

We segmented individual normal cells from the first data set which does not have ground truth. We also segmented images from the second dataset which has ground truth. The two data sets are significantly different from each other in terms of the colors of cytoplasm and background. But the results obtained from both data sets indicate that the proposed method can be applied to different image data sets. Figure 6 presents the results showing the nucleus segmentation steps whereas Fig. 7 presents the segmentation steps of cytoplasm and WBC. Figure 8 visualizes the results for cell segmentation using SLIC and marker-controlled watershed. Table 1 gives a summary of the threshold and contrast means for the WBCs.

We provide a summary of the performance analysis of the proposed method. We use Jaccard indices, Dice similarity coefficients, sensitivity, specificity, and Precision to evaluate the performance of the proposed method. The performance analysis results are summarized in Tables 2 and 3. We also compare the result obtained using the proposed method to the state of art methods. Table 2 compares the performance of the method on segmenting nucleus to the results obtained using CNN, and SVM [21]. Table 3 compares the performance of the method on segmenting WBCs to the results obtained using a Deep neural network [22], U-Net [22] and WBC-Net [23].

Table 1 Summary for 976 sample images (251 eosinophils, 201 neutrophils, 208 monocytes, and 316 lymphocytes)

	Eosinophils	Neutrophils	Monocyte	Lympoyctes
Thresholds mean for Nucleus segmentation	0.338991314	0.356037820	0.359911376	0.362329243
Thresholds mean for WBC segmentation	0.894988532	0.887716863	0.892829014	0.890228957
Contrast Mean between nucleus and cytoplasm	0.555997217	0.531679043	0.532917638	0.527899714

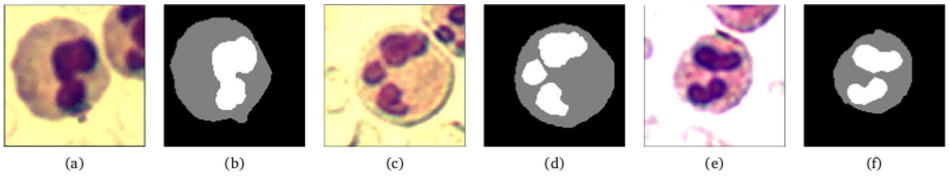


Fig. 9 Sample of the excluded images. From the left: **a, c, e** are original images, whereas **b, d, f** are their corresponding ground truth images. Each original image seems to have two WBCs (one is showing only part of it) but the ground truth considered one cell which is at the center. Since the suggested approach is based on the intensity values it segments both WBCs

Table 2 The performance of the proposed method for nucleus segmentation compared to U-Net, CNN, support-vector machine (SVM) based on the data set two presented using average in each measure of similarity

	JI	DSC	Specificity	Sensitivity	Precision
Proposed method	0.902925	0.948079	0.993110	0.952951	0.948903
U-Net [36]	0.917556	0.956456	0.995364	0.950505	0.964065
CNN [21]	–	0.910000	0.979200	0.960800	0.876300
SVM [21]	–	0.860000	0.818600	0.989700	0.934300

Also, we train the U-Net model in [36] on the augmented data and the results are included in Tables 2 and 3. Figure 10 provides a correlation of nuclei size between the predicted segmentation and the ground truth. Also, Fig. 11 gives a correlation of WBCs size between the predicted segmentation and the ground truth. Additional file 1: Figs. 12 and 13 in present the convergence of the normalized forward sensitivity index.

We performed a visual representation of segmentation results obtained using the proposed method. We provide more segmentation results in Additional file 1: Figs. 14 to 17. Also, Additional file 1: Fig. 18 provides subgraphs showing the impact of the control parameters by tuning different values. Additional file 1: Figure 19 shows the training performance, Additional file 1: Figs. 20 and 21 shows the prediction of the nuclei and WBC masks respectively. The proposed method has demonstrated its effectiveness in segmenting high-quality and poor-quality images. For images in which the RBCs do not touch the WBCs, the method works well even if the cytoplasm color is indistinguishable from the RBC's color. For images in which the RBCs touch the WBCs, the method works well when the cytoplasm color is distinguishable from the RBCs.

Table 3 The performance of the proposed method for WBC segmentation compared to deep neural network (LeukocyteMask(Aug-ET)), U-Net (implemented in [22]) and WBC-Net based on the data set two presented using average in each measure of similarity

	JI	DSC	Specificity	Sensitivity	Precision
Proposed method	0.942908	0.970176	0.975164	0.992475	0.950133
U-Net [36]	0.941467	0.969628	0.989037	0.962368	0.977735
Deep Neural Network [22]	–	0.981960	–	–	0.995440
U-Net [22]	–	0.961630	–	–	0.933190
WBC-Net [23]	–	0.989200	–	–	0.989700

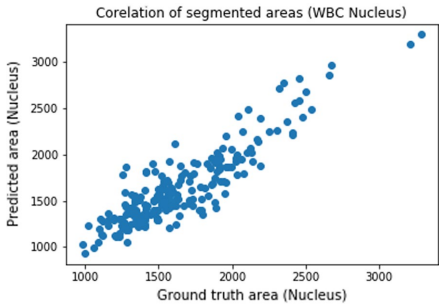


Fig. 10 Correlation of the segmented nuclei size. Correlation for the size of the predicted nuclei to the size of the nuclei segmented manually

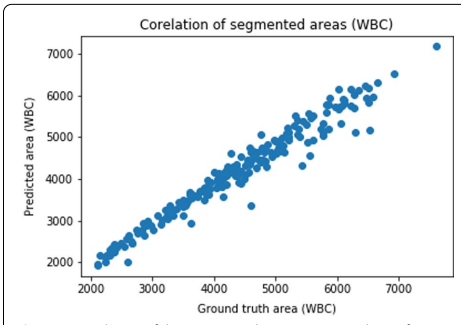


Fig. 11 Correlation of the segmented WBCs size. Correlation for the size of the predicted WBCs to the size of the WBCs segmented manually

Whenever the cytoplasm color is indistinguishable from the RBCs and they touch each other, techniques for separating touching objects can be used. For images with very low contrast between the cytoplasm and background, the method can successfully segment the nucleus, but not the cytoplasm.

Abbreviations

WBC: White blood cells; RBC: Red blood cells; SLIC: Simple linear iterative clustering; H and E: Hematoxylin and eosin; FCM: Fuzzy C- means algorithm; ALL: Acute lymphoblastic leukemia; RGB: Red, green, and blue; HSV: Hue saturation value; SVM: Support vector machine; ROI: Region of interest; JI: Jaccard index; DSC: Dice similarity coefficients; CIELAB: Implies L*a*b* color space, L* for perceptual lightness, a* and b* for the four unique colors of human vision.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12880-022-00801-w.

Additional file 1. The file presents sensitivity analysis of parameters, Fig. 14 to 18 presentmore experimental results. Also, the file presents brie y theconcepts of B-Spline interpolation of the data points and the SLICalgorithm. Furthermore, the file presents results for theperformance of the U-Net model training and the predicted masks fornuclei and WBCs.

Author contributions

SM implemented the proposed method and run the analysis to generate the results. JBH reviewed the biological concepts and clarity of the data set used. Interpretation of the results was done by all authors. Drafting, writing, and proofreading to approve the manuscript were done by all authors. All authors read and approved the final manuscript.

Funding

This research work is funded by the University of Bergen, Norway.

Availibility of data and materials

All the data set analyzed in this study are freely available online. The first data set is available at "https://data.mendeley.com/datasets/snkd93bnjr/draft?a=d9582c71-9af0-4e59-9062-df30df05a121" and the second data set is available at is "https://github.com/zxaoyou".

Declarations

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Department of Mathematics, University of Bergen, Allégaten 41, 5007 Bergen, Norway. [2]Department of Clinical Medicine, Neuro-SysMed, University of Bergen, PO box 7804, 5020 Bergen, Norway. [3]Department of Neurology, Neuro-SysMed, Haukeland University Hospital, Jonas Lies vei 71, 5053 Bergen, Norway.

Received: 25 October 2021 Accepted: 12 April 2022
Published online: 26 April 2022

References

1. Mathur A, Tripathi AS, Kuse M. Scalable system for classification of white blood cells from Leishman stained blood stain images. J Pathol Inform. 2013;4(Suppl):S15.
2. Prinyakupt J, Pluempitiwiriyawej C. Segmentation of white blood cells and comparison of cell morphology by linear and Naïve Bayes classifiers. Biomed Eng. 2015;14(1):63.
3. Hemalatha R, Thamizhvani T, Dhivya AJA, Joseph JE, Babu B, Chandrasekaran R. Active contour based segmentation techniques for medical image analysis. Med Biol Image Anal. 2018;17.
4. Gonzalez RC, Woods RE. Digital image processing, 4th edn. 330 Hudson Street, New York, NY 10013;2008.
5. Ismail A, Marhaban M. A simple approach to determine the best threshold value for automatic image thresholding. In: 2009 IEEE international conference on signal and image processing applications, pp. 162–6 (2009). IEEE.
6. Lazar M, Hladnik A. Implementation of global and local thresholding algorithms in image segmentation of coloured prints. In: 35th international research conference IARIGAI, vol. 35 (2008).
7. Singh TR, Roy S, Singh OI, Sinam T, Singh K, et al. A new local adaptive thresholding technique in binarization. arXiv preprint arXiv:1201.5227 (2012).

8. Firdousi R, Parveen S. Local thresholding techniques in image binarization. Int J Eng Comput Sci. 2014;3(03):54.
9. Li Y, Zhu R, Mi L, Cao Y, Yao D. Segmentation of white blood cell from acute lymphoblastic leukemia images using dual-threshold method. Comput Math Methods Med. 2016;2016.
10. Chan Y-K, Tsai M-H, Huang D-C, Zheng Z-H, Hung K-D. Leukocyte nucleus segmentation and nucleus lobe counting. BMC Bioinform. 2010;11(1):1–18.
11. Theera-Umpon N. White blood cell segmentation and classification in microscopic bone marrow images. In: International conference on fuzzy systems and knowledge discovery, pp. 787–796 (2005). Springer.
12. Felzenszwalb PF, Huttenlocher DP. Efficient graph-based image segmentation. Int J Comput Vis. 2004;59(2):167–81.
13. Shi J, Malik J. Normalized cuts and image segmentation. In: Proceedings of IEEE computer society conference on computer vision and pattern recognition, pp. 731–737 (1997). IEEE.
14. Shi J, Malik J. Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell. 2000;22(8):888–905.
15. Salem NM. Segmentation of white blood cells from microscopic images using k-means clustering. In: 2014 31st national radio science conference (NRSC), pp. 371–376 (2014). IEEE.
16. Miao H, Xiao C. Simultaneous segmentation of leukocyte and erythrocyte in microscopic images using a marker-controlled watershed algorithm. Comput Math Methods Med. 2018;2018.
17. Al-Dulaimi K, Tomeo-Reyes I, Banks J, Chandran V. White blood cell nuclei segmentation using level set methods and geometric active contours. In: 2016 international conference on digital image computing: techniques and applications (DICTA), pp. 1–7 (2016). IEEE.
18. Ghane N, Vard A, Talebi A, Nematollahy P. Segmentation of white blood cells from microscopic images using a novel combination of k-means clustering and modified watershed algorithm. J Med Signals Sens. 2017;7(2):92.
19. Kuse M, Sharma T, Gupta S. A classification scheme for lymphocyte segmentation in h&e stained histology images. In: International conference on pattern recognition, pp. 235–243 (2010). Springer.
20. Sadeghian F, Seman Z, Ramli AR, Kahar BHA, Saripan M-I. A framework for white blood cell segmentation in microscopic blood images using digital image processing. Biol Proced. 2009;11(1):196.
21. Banik PP, Saha R, Kim K-D. An automatic nucleus segmentation and CNN model based classification method of white blood cell. Expert Syst Appl. 2020;149:113211.
22. Fan H, Zhang F, Xi L, Li Z, Liu G, Xu Y. Leukocytemask: an automated localization and segmentation method for leukocyte in blood smear images using deep neural networks. J Biophotonics. 2019;12(7):201800488.
23. Lu Y, Qin X, Fan H, Lai T, Li Z. Wbc-net: a white blood cell segmentation network based on unet++ and resnet. Appl Soft Comput. 2021;101:107006.
24. Mittal A, Dhalla S, Gupta S, Gupta A. Automated analysis of blood smear images for leukemia detection: a comprehensive review. ACM Comput Surv (CSUR) (2022).
25. Long F. Microscopy cell nuclei segmentation with enhanced u-net. BMC Bioinform. 2020;21(1):1–12.
26. Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Süsstrunk S. Slic superpixels compared to state-of-the-art superpixel methods. IEEE Trans Pattern Anal Mach Intell. 2012;34(11):2274–82.
27. Beucher S, et al. The watershed transformation applied to image segmentation. Scan Microsc Suppl. 1992;299.
28. Prautzsch H, Boehm W, Paluszny M. Bézier and B-spline techniques, vol. 6. Berlin: Springer; 2002.
29. Lyche T, Morken K. Spline methods draft. Oslo: Department of Informatics, Center of Mathematics for Applications, University of Oslo; 2008. p. 3–8.
30. Acevedo A, Merino A, Alférez S, Molina Á, Boldú L, Rodellar J. A dataset of microscopic peripheral blood cell images for development of automatic recognition systems. Data Brief. 2020;105474.
31. Zheng X, Wang Y, Wang G, Liu J. Fast and robust segmentation of white blood cell images by self-supervised learning. Micron 107.
32. ...Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, Wieser E, Taylor J, Berg S, Smith NJ, Kern R, Picus M, Hoyer S, van Kerkwijk MH, Brett M, Haldane A, Fernández del Río J, Wiebe M, Peterson P, Gérard-Marchant P, Sheppard K, Reddy T, Weckesser W, Abbasi H, Gohlke C, Oliphant TE. Array programming with NumPy. Nature. 2020;585:357–62. https://doi.org/10.1038/s41586-020-2649-2.
33. ...Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, van der Walt SJ, Brett M, Wilson J, Millman KJ, Mayorov N, Nelson ARJ, Jones E, Kern R, Larson E, Carey CJ, Polat I, Feng Y, Moore EW, VanderPlas J, Laxalde D, Perktold J, Cimrman R, Henriksen I, Quintero EA, Harris CR, Archibald AM, Ribeiro AH, Pedregosa F, van Mulbregt P. SciPy 1.0 contributors: SciPy 1.0: fundamental algorithms for scientific computing in python. Nat Methods. 2020;17:261–72. https://doi.org/10.1038/s41592-019-0686-2.
34. Liao P-S, Chen T-S, Chung P-C, et al. A fast algorithm for multilevel thresholding. J Inf Sci Eng. 2001;17(5):713–27.
35. Stutz D, Hermans A, Leibe B. Superpixels: an evaluation of the state-of-the-art. Comput Vis Image Underst. 2018;166:1–27.
36. Ronneberger O, Fischer P, Brox T. U-net: convolutional networks for biomedical image segmentation. In: International conference on medical image computing and computer-assisted intervention, pp. 234–241 (2015). Springer.
37. Chollet F, et al. Keras. https://github.com/fchollet/keras.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Ready to submit your research? Choose BMC and benefit from:

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

At BMC, research is always in progress.

Learn more biomedcentral.com/submissions

1. Supplementary Information

1.1. Sensitivity analysis of T_{nc0}

The concept of sensitivity analysis and parameter tuning for image segmentation has been used in [1]. We use the formula for normalized forward sensitivity index [2] of the variable T_{nc0} to a parameter n which is defined as

$$\Upsilon_{n}^{T_{nc0}} = \frac{\partial T_{nc0}}{\partial n} \times \frac{n}{T_{nc0}} \tag{1a}$$

From Eq. 2 in the article. For n_1

$$\Upsilon_{n_1}^{T_{nc0}} = \frac{\partial T_{nc0}}{\partial n_1} \times \frac{n_1}{T_{nc0}} = -\frac{Max(A)}{n_1^2} \times \frac{n_1}{T_{nc0}} \tag{2a}$$

For n_2

$$\Upsilon_{n_2}^{T_{nc0}} = \frac{\partial T_{nc0}}{\partial n_2} \times \frac{n_2}{T_{nc0}} = -\frac{Min(A)}{n_2^2} \times \frac{n_2}{T_{nc0}} \tag{3a}$$

By considering a series of images we investigate the convergence of $\Upsilon_{n_1}^{T_{nc0}}$ and $\Upsilon_{n_2}^{T_{nc0}}$ as $n_1 \to \infty$ and $n_2 \to \infty$ respectively.

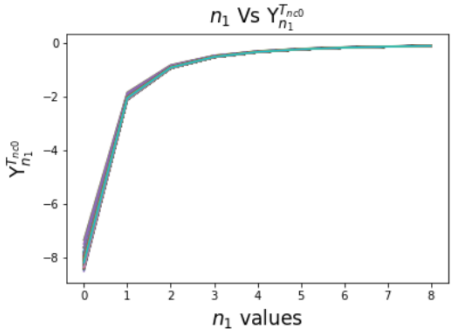


Figure 12: The values of $\Upsilon_{n_1}^{T_{nc0}}$ start stabilizing around $n_1 = 3$. The number of images tested is 8960.

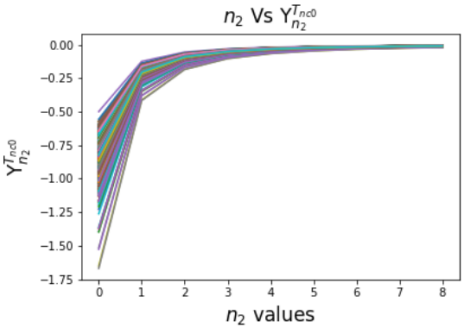


Figure 13: The values of $\Upsilon_{n_2}^{T_{nc0}}$ stabilizes around $n_2 = 3$. The number of images tested is 8960.

For $n_1 = n_2 = 3$, Eq. 2 in the article becomes

$$T_{nc0} \approx \frac{Max(A)}{3} + \frac{Min(A)}{3} \tag{4a}$$

From Eq. 4a, for images whose intensity values are in the intervals $[Min(A), Max(A)] = [0, 1]$, $T_{nc0} = \frac{1}{3}$.
For images whose intensity values are in the interval of either $(Min(A), Max(A)) = (0, 1)$, $(Min(A), Max(A)] = (0, 1]$

or $[Min(A), Max(A)) = [0, 1)$, either $T_{nc0} \leq \frac{1}{3}$ or $T_{nc0} \geq \frac{1}{3}$.
Looking at Fig. 12 and 13, the values of $\Upsilon_{n_2}^{T_{nc0}}$ and $\Upsilon_{n_1}^{T_{nc0}}$ show that n_1 is more sensitive compared to n_2. So, n_1 needs to be chosen carefully.

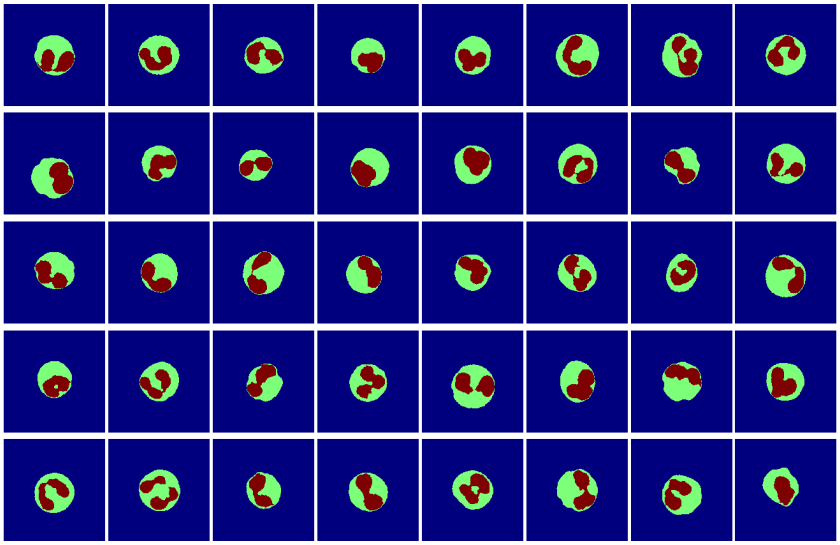


Figure 14: Segmented nucleus and cytoplasm of white blood cells. Eosinophil images.

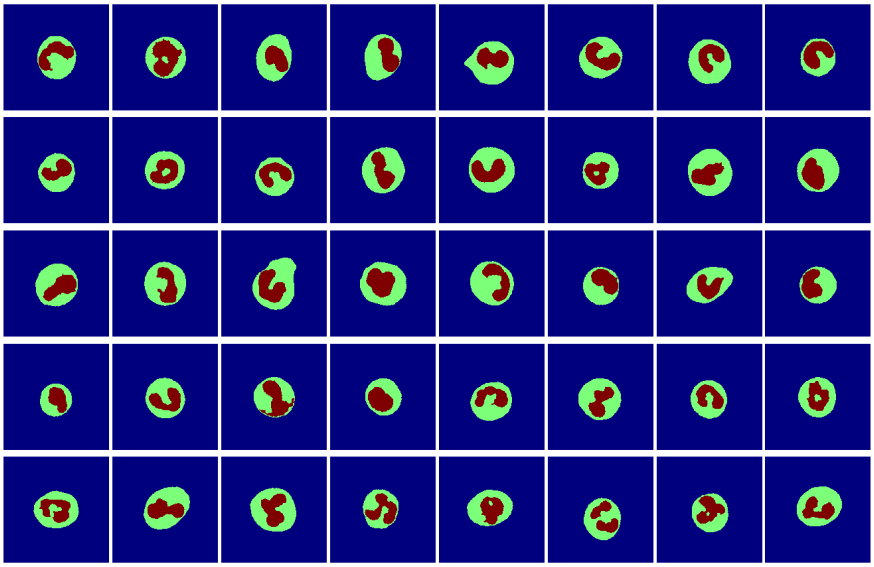


Figure 15: Segmented nucleus and cytoplasm of the white blood cells. Neutrophils images.

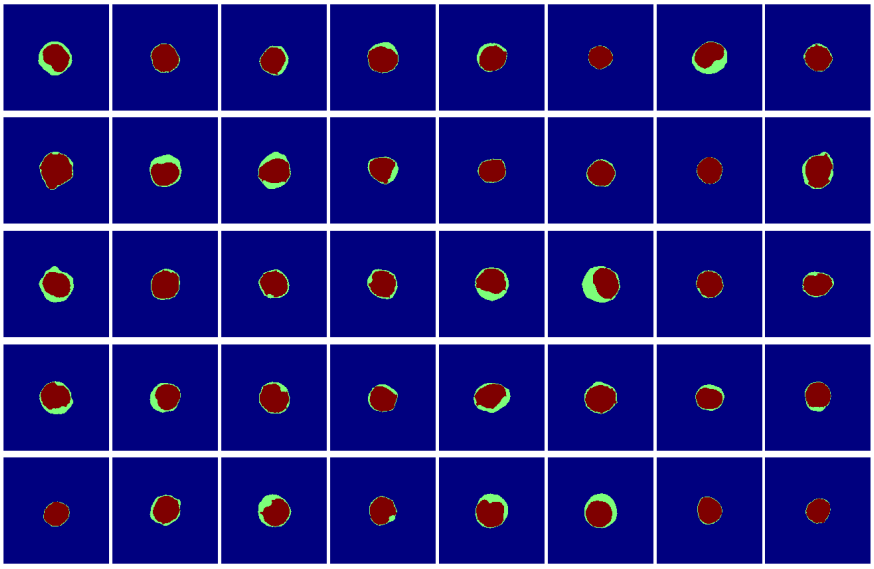


Figure 16: Segmented nucleus and cytoplasm of white blood cells. Lymphocytes images.

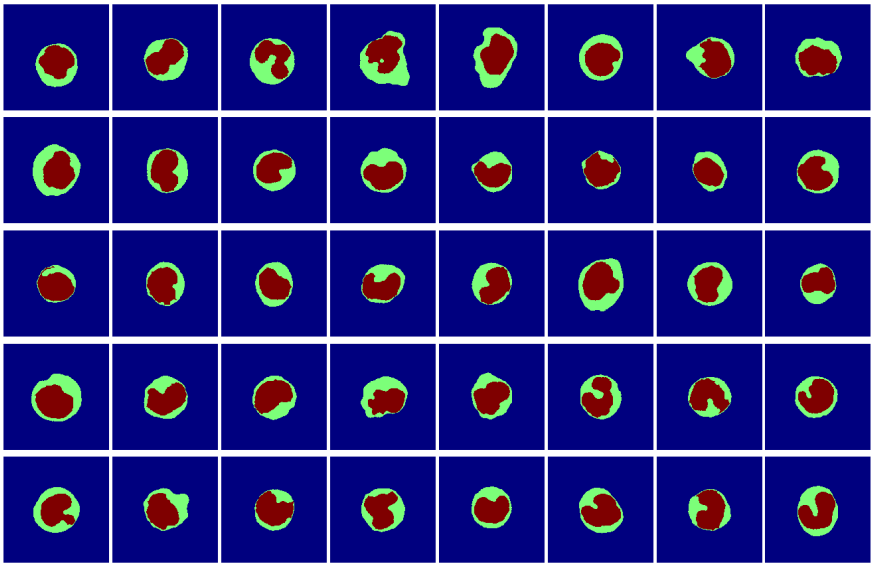


Figure 17: Segmented nucleus and cytoplasm of the white blood cells. Monocyte images.

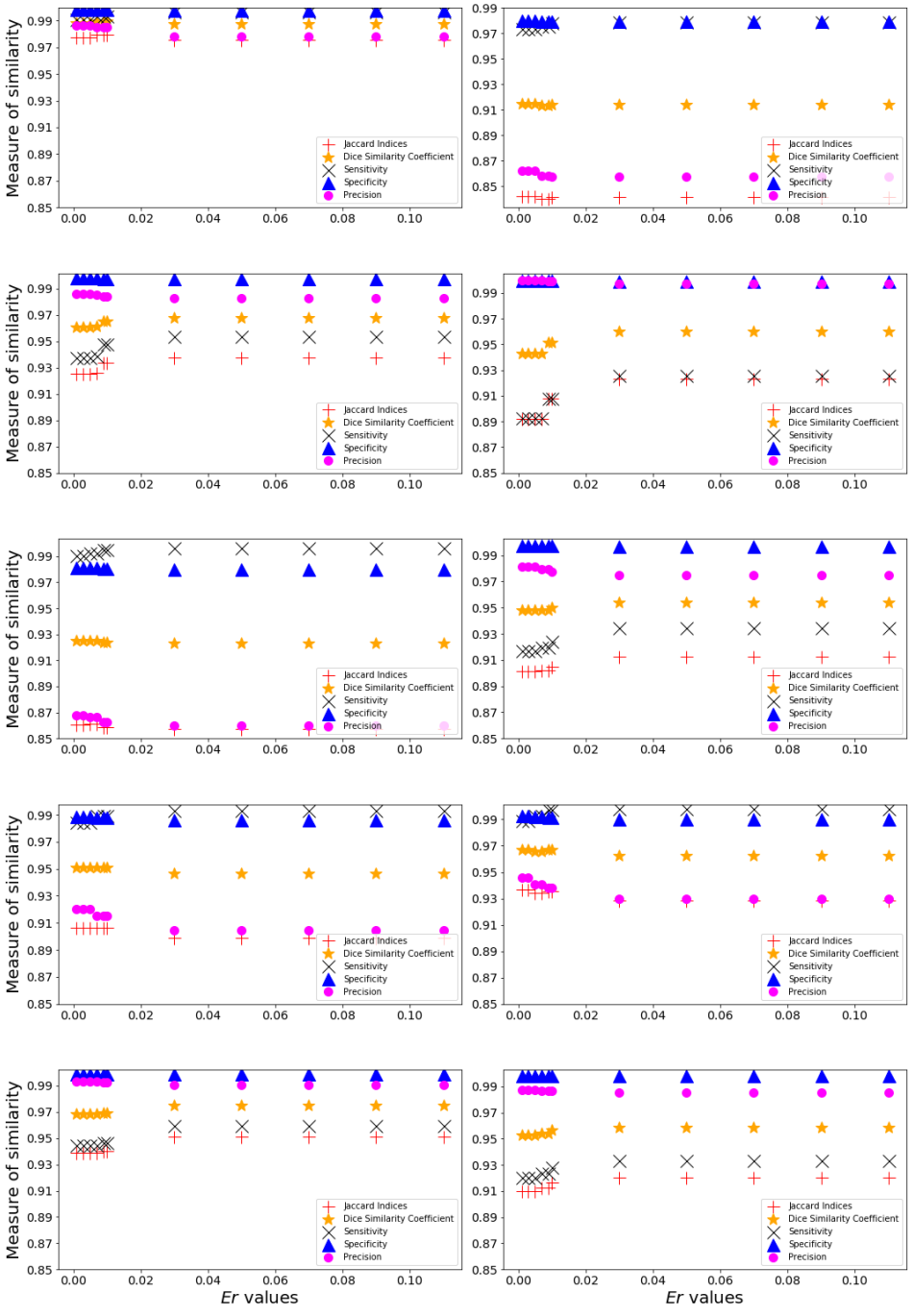


Figure 18: Graphs showing measures of similarity (vertical axis) of the segmented nucleus using the proposed method in relation to the tuned Er values (horizontal axis). The Er values tested include: 0.001, 0.003, 0.005, 0.007, 0.009, 0.01, 0.03, 0.05, 0.07, 0.09, 0.11.

1.1.1. B-Spline interpolation of the data points

A spline is a piecewise polynomial function that is constructed as a means to interpolate a set of data points . Each piece of the function is confined between knots $t_i < t_{i+1}$ with strict continuity conditions between each piece at the knots. These polynomials are strung together at the knots to form the completed curve . Now, B-splines (basis splines) are a set of piecewise functions that form a basis for the set of all possible spline functions. Any spline function can be described as a linear combination of B-splines.

Let U be a set $m+1$ non-decreasing numbers $t_0 < t_1 < t_2 < \cdots < t_m$ over the interval which we want to interpolate the data. The t_is are called knots and the set U is called knot vector. Then, the i^{th} B-spline function of degree 0 is given by:

$$B_i^0(x) = \begin{cases} 1 \text{ if } t_i < x < t_{i+1} \\ 0 \text{ elsewhere} \end{cases}$$

The subsequent higher-level B-splines are constructed from the recurrence relation:

$$B_i^d(x) = \frac{x - t_i}{t_{i+d} - t_i} B_i^{d-1}(x) + \frac{t_{i+d+1} - x}{t_{i+d+1} - t_{i+1}} B_i^{d-1}(x) \tag{5a}$$

Equation 5a creates the basis functions for a d degree spline. So, a linear combination of B-Splines or spline function is a combination of form

$$g(x) = \sum_i^n c_i^d B_i^d(x) \tag{6a}$$

where c_is are the coefficients of the basis functions. The computation of these coefficients can be found in the reference sources. A linear combination of these functions in equation 6a can be used to interpolate a data set. In order to interpolate the data, the B-spline curve must span the interval of the data points to be interpolated.

1.1.2. Summary of SLIC Algorithm

Mainly, the algorithm involves three steps which include initialization, assignment, and enforce-connectivity. It performs K-means segmentation using 5 dimension space of color information and image location. The RGB color space is transformed into $CIELAB$ color space [3] and then each of the pixel images is mapped into (l, a, b, x, y). The algorithm starts by initializing a k number of clusters which are sampled on a regular grid. Then in the assignment step each pixel p_i is associated with the nearest cluster centre. Different from the standard k-means, SLIC updates the cluster centres by searching in a certain neighbourhood and not on the whole image. But the update step depends on the convergence of the error term which is obtained by computing the residual error between the old and the new centres. In case there are disjoint pixels from the previous steps then enforcing connectivity is applied so as to assign them a nearest superpixel.

1.2. Training U-Net Model for segmenting Nucleus and WBCs

The model training was run on google colab [4] using NVIDIA-SMI TESLA k80. The model was trained by using activation 'relu', kernel initializer 'he_normal', and padding is the same. The activation function used is sigmoid and the initial learning rate is set to be 0.0001, and we use the Adam optimizer. The training performance is visualized in Figure 19. The predicted nuclei and WBCs masks are visualized in Figure 20 and 21 respectively.

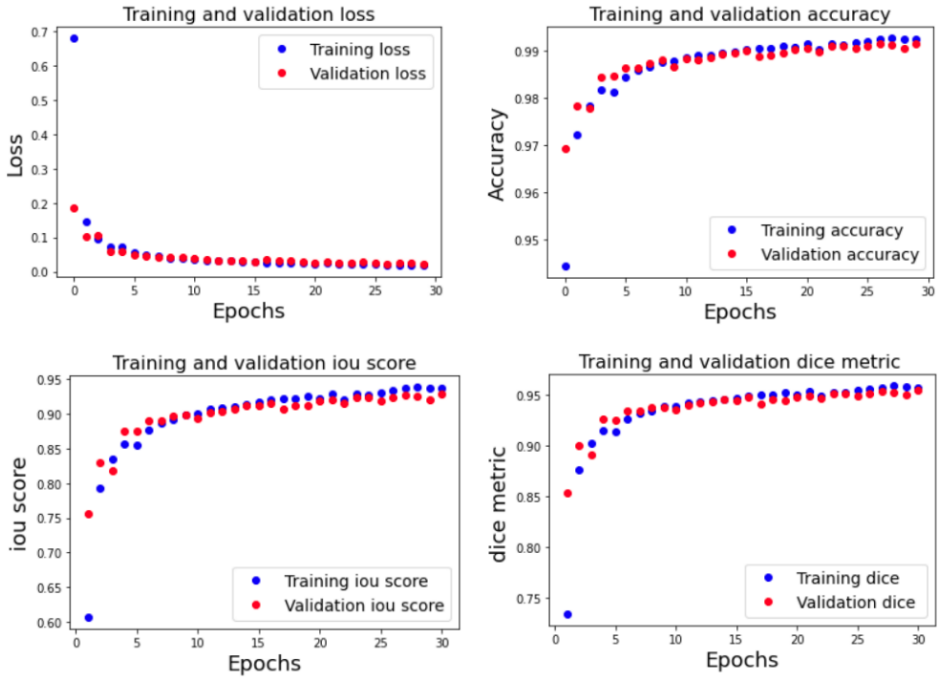


Figure 19: Training performance for predicting nuclei masks. First row: Loss and accuracy for both training and validation. Second row: Dice similarity coefficient and iou score during the training and validation. The results for performance training of the WBCs masks is not included.

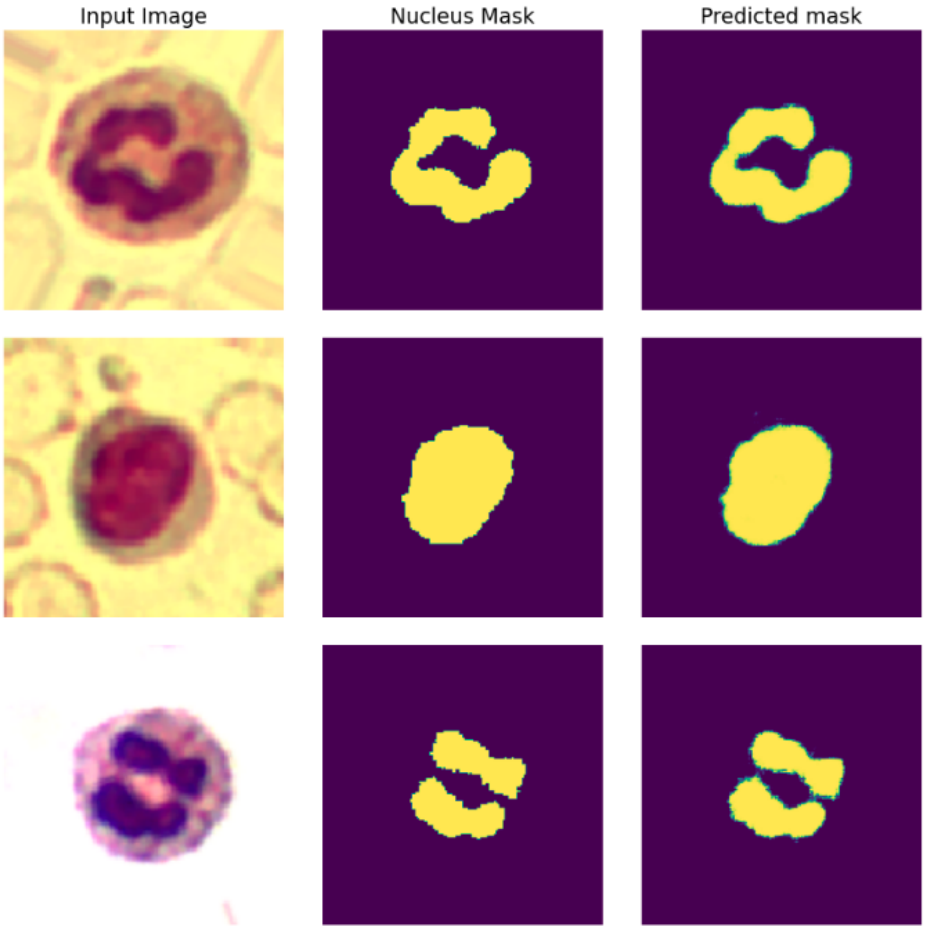


Figure 20: Prediction of nuclei masks: The first column presents input images, the second column presents the nuclei masks, and the third column shows the predicted nuclei masks.

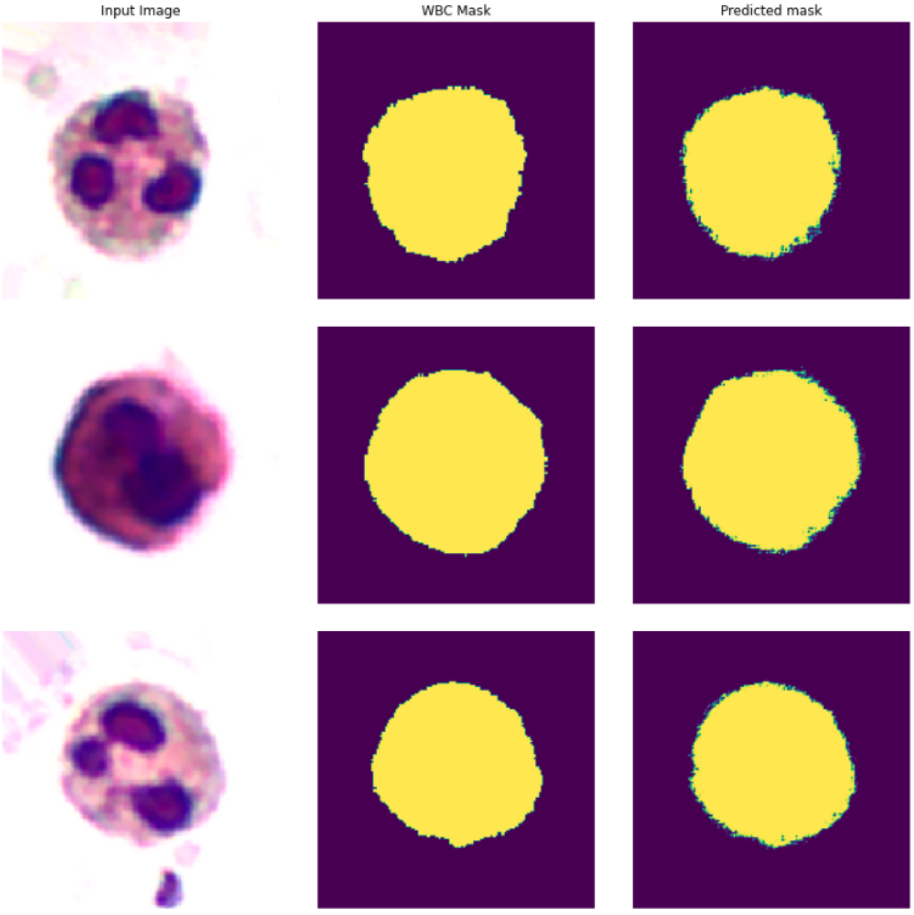


Figure 21: Prediction of WBCs masks:The first column presents input images, the second column presents the WBCs masks, and the third column shows the predicted WBCs masks.

References

[1] G. Teodoro, T. M. Kurç, L. F. Taveira, A. C. Melo, Y. Gao, J. Kong, and J. H. Saltz, "Algorithm sensitivity analysis and parameter tuning for tissue image segmentation pipelines," *Bioinformatics*, vol. 33, no. 7, pp. 1064–1072, 2017.

[2] N. Chitnis, J. M. Hyman, and J. M. Cushing, "Determining important parameters in the spread of malaria through the sensitivity analysis of a mathematical model," *Bulletin of mathematical biology*, vol. 70, no. 5, p. 1272, 2008.

[3] C. Pub, "Technical report 15: 2004: Colorimetry," 2004.

[4] E. Bisong, *Building machine learning and deep learning models on Google cloud platform*. Springer, 2019.

Paper B

Brain Tumor Segmentation Based on Minimum Spanning Tree.

Simeon Mayala, Ida Herdlevær, Jonas Bull Haugsøen, Shamundeeswari Anandan, Sonia Gavasso, and Morten Brun

Frontiers in Signal Processing, **Volume 2**, Article 816186 (2022)

frontiers in Signal Processing

ORIGINAL RESEARCH
published: 11 March 2022
doi: 10.3389/frsip.2022.816186

Brain Tumor Segmentation Based on Minimum Spanning Tree

Simeon Mayala[1]*, *Ida Herdlevær*[2,3], *Jonas Bull Haugsøen*[2,3], *Shamundeeswari Anandan*[2,3], *Sonia Gavasso*[2,3] *and Morten Brun*[1]

[1]Department of Mathematics, University of Bergen, Bergen, Norway, [2]Department of Clinical Medicine, University of Bergen, Bergen, Norway, [3]Neuro-SysMed, Department of Neurology, Haukeland University Hospital, Bergen, Norway

In this paper, we propose a minimum spanning tree-based method for segmenting brain tumors. The proposed method performs interactive segmentation based on the minimum spanning tree without tuning parameters. The steps involve preprocessing, making a graph, constructing a minimum spanning tree, and a newly implemented way of interactively segmenting the region of interest. In the preprocessing step, a Gaussian filter is applied to 2D images to remove the noise. Then, the pixel neighbor graph is weighted by intensity differences and the corresponding minimum spanning tree is constructed. The image is loaded in an interactive window for segmenting the tumor. The region of interest and the background are selected by clicking to split the minimum spanning tree into two trees. One of these trees represents the region of interest and the other represents the background. Finally, the segmentation given by the two trees is visualized. The proposed method was tested by segmenting two different 2D brain T1-weighted magnetic resonance image data sets. The comparison between our results and the gold standard segmentation confirmed the validity of the minimum spanning tree approach. The proposed method is simple to implement and the results indicate that it is accurate and efficient.

Keywords: brain tumor, brain tumor segmentation, minimum spanning tree, segmentation, image processing

OPEN ACCESS

Edited by:
Tao Lei,
Shaanxi University of Science and Technology, China

Reviewed by:
Zhenghao Shi,
Xi'an University of Technology, China
Baha Şen,
Yıldırım Beyazıt University, Turkey

***Correspondence:**
Simeon Mayala
simeon.mayala@uib.no

Specialty section:
This article was submitted to Image Processing, a section of the journal Frontiers in Signal Processing

Received: *16 November 2021*
Accepted: *31 January 2022*
Published: *11 March 2022*

Citation:
Mayala S, Herdlevær I, Haugsøen JB, Anandan S, Gavasso S and Brun M (2022) Brain Tumor Segmentation Based on Minimum Spanning Tree. Front. Sig. Proc. 2:816186. doi: 10.3389/frsip.2022.816186

1 INTRODUCTION

A brain tumor is a collection of abnormal cells in the brain: they may be malignant (cancerous) or benign (noncancerous) and can be categorized as primary or secondary. Primary brain tumors originate in the brain and are either glial or non-glial. Glial brain tumors are the most common, arising from the supporting cells of the brain. Non-glial tumors may originate from any other tissue in the brain, such as meninges, neurons, blood vessels, and glands. Primary tumors can be malignant (cancerous) or benign. Secondary brain tumors develop in another part of the body and metastasize to the brain. Cancers that commonly metastasize to the brain include lung cancer, breast cancer, kidney cancer, and skin cancer. Secondary brain tumors are always malignant. Brain tumors may be located in any part of the brain. Magnetic resonance imaging (MRI) is important in diagnosing and monitoring brain tumors. Brain tumor segmentation is an essential step in analyzing and interpreting such images (Ciesielski and Udupa, 2011; Banerjee et al., 2016). Segmenting brain tumors using automatic techniques is challenging because of factors that cause complexity during segmentation. These factors include the location in the brain, irregular shapes, different sizes, types of tumors, blurred boundaries, and noise.

The problem of brain tumor segmentation is studied in different research works. Skull- stripping is a fundamental preprocessing step to isolate the brain tissue before brain tumor segmentation

(Kalavathi and Prasath, 2016). It removes the non-brain tissues such as skin, fat, muscle, neck, and eyeballs from the image. This step simplifies the complexity of the brain image and increases the speed and accuracy of the segmentation process. Some popular tools developed for skull stripping include: Freesurfer's strip skull (FSS) (Dale et al., 1999), Brain Surface Extractor (BSE) (Roy and Maji, 2015), Brain Extraction Tool (BET) (Smith, 2000), Hybrid Approach (HWA) (Ségonne et al., 2004), Robust Brain Extraction (ROBEX) (Iglesias et al., 2011) and Hahn and Peitgen's Watershed Algorithm (WAT) (Hahn and Peitgen, 2000).

Different methods are used for brain tumor segmentation. They are categorized based on their formulation and how they perform the segmentation. These include traditional image segmentation, machine learning, deep learning (Huang et al., 2021), and graph-based methods such as those based on the minimum spanning tree (MST) (Long and Sun 2020; Kang 2021). The limited flexibility of many existing segmentation methods, especially those reviewed below, necessitates careful fine-tuning of parameters. In many works, an MST is used for segmenting images with control parameters such as specifying thresholds, specifying the size of the regions to be segmented and the number of neighbors to be considered. We propose a method that uses the MST without tuning parameters except a smoothing parameter used in the preprocessing step. Moreover, our method segments brain tumors without the necessity of skull stripping. Concerning what is stated in the article Apropos of Signal Processing (Nandi, 2021), we adopt the existing theoretical ideas and algorithms and use them to segment brain tumors.

Before describing the proposed method, we review related literature to show how the MST is applied in image segmentation. Zahn (1971) proposed an MST-based approach for addressing the problem of detecting and separating different inherent clusters. Zahn's method was aimed at clustering of point clouds and segmenting images. The method obtains segments by dropping inconsistent edges, with weights below the threshold, to break the minimum spanning tree into a collection of trees. This idea is a powerful tool for point clustering and image segmentation. However, depending on the chosen threshold, the high variability regions are likely to be split into multiple regions that should be merged. To address the shortcoming, Urquhart (1982) proposed a non-parametric hierarchic clustering method based on the concept of limited neighborhood sets and Gabriel graph.

Xu and Uberbacher (1997) proposed another method for image segmentation based on the MST. The method constructs a weighted planar graph from a 2D gray-level image. Then, the MST is constructed from the graph so that the connected homogeneous regions correspond to one sub-tree of the spanning tree. The algorithm partitions the tree into a set of subtrees and each subtree consists of the nodes with similar gray levels. To avoid forming many small regions, conditions are introduced. Partitions are controlled by the condition that each partitioned region has at least a specified number of pixels and that two adjacent regions have average gray levels that differ by more than a specified value.

Felzenszwalb and Huttenlocher (2004) proposed a graph-based method for image segmentation. The method defines a predicate for measuring the evidence for a boundary between two regions in the image. Based on the defined predicate, an efficient segmentation algorithm is developed. It considers two quantities to measure the evidence for the boundary. The paper defines two criteria whether there is evidence for a boundary between components or partitions. The two criteria include the internal difference (the largest weight in the MST of the component) and the difference between the two components. A special property of the algorithm is its ability to preserve details in "low variability regions while ignoring details in high variability regions." The algorithm segments the images efficiently and produces segments that capture the global properties.

Other researchers improved a successful data clustering method based on Prim's MST representation for performing image segmentation (Saglam and Baykan, 2017). The algorithm scans the complete MST structure of the entire image to obtain and cut the inconsistent edges. Also, they develop a cutting criterion that considers several local and global features. The proposed method competes with other algorithms in terms of execution time. Also, Long and Sun (2020) proposed an algorithm to tackle the challenge of the ill-posedness of image segmentation. The proposed algorithm is based on the MST. They propose a different formula for RGB color space with respect to angular distance color as the weight of judge standard of the segmentation. The judging standard involves the spatial distance and vector relationship between two pixels. The experimental results show that the proposed algorithm is effective.

In summary, our focus is to segment brain tumors without skull stripping from the T1-weighted MRI and compare the result to their ground truth segmentation. We utilize the MST efficiency to extract the brain tumor directly without skull stripping. Also, we test the method by segmenting the brain on simulated MRI for the brain and compare the result to the ground truth. The proposed method is mainly composed of the following steps: (I) making a graph and constructing a minimum spanning tree and (II) interactively segmenting the region of interest.

2 MATERIALS AND METHODS

2.1 Material

In this paper, we use two different image data sets. The first data set includes 3,064 slices of 2D brain T1-weighed (T1W) contrast-enhanced (CE)-MRI from 233 patients collected at Nanfang Hospital, Guangzhou, China, and General Hospital, Tianjin Medical University, China. The images are classified into three types of brain tumors, labeled as follows: 1) meningioma, 2) glioma, and 3) pituitary tumors (Cheng et al., 2015; Cheng et al., 2016). The images were acquired with a slice thickness of 6 mm and the slice gap is 1 mm. The image dataset is provided in the Matlab format (*.mat*). Each file stores a struct containing different fields for an image. The fields included in each file are labels for the tumor type, an anonymized patient ID, the image data, tumor borders, and tumor masks. The reader is referred to (Cheng, 2017) for additional images and original images.

The second data sets consist of 20 simulated T1W MR images of normal brains from the BrainWeb website. They are anatomical models consisting of a set of 3D tissue membership volumes, one for each tissue class: background, cerebrospinal fluid (CSF), gray Matter, white matter, fat, muscle, muscle/skin, skull, vessels, around fat, dura mater, bone marrow. Each label at a voxel in the anatomical model represents the tissue that contributes the most to that voxel. They have a 0.5 mm isotropic voxel size and they are stored in the *MINC* format. For more information refer to Cocosco et al. (1997) and Aubert-Broche et al. (2006).

2.2 Methods

In this section, we establish a segmentation method based on the MST. We define important terms that will be referred to when using the proposed method.

2.2.1 Graph

Let $G = (V, E)$ be connected, undirected and weighted graph with nodes $V = \{v_1, v_2, v_3, \ldots, v_n\}$ and edges $E = \{e_1, e_2, e_3, \ldots, e_m\}$ such that e_i is a weighted link between two neighboring nodes, and $|V| = n$, $|E| = m$. The graph is obtained after mapping an image into a graph, and each node in the graph G represents a pixel in the input image. If the pixel's intensity value of the image at position (x, y) is represented by $I_{x,y}$, then, the corresponding node's value is

$$v_i = I_{x,y} \tag{1}$$

The associated weight w_j to the edge e_j is a measure of the similarity between two neighboring pixels. The edge's weight is the absolute value of the difference between the intensity values of the pixels v_i and v_i' that constitute the edge (Morris et al., 1986). Then, e_j weight is

$$w_j = |v_i - v_i'| \tag{2}$$

Note that v_i and v_i' are neighboring nodes to avoid the confusion of indices. In this paper, we consider the 4-connected neighborhood of each node.

2.2.2 Path in a Graph

A path P in graph G is a sequence of edges joining two terminal nodes say $P = \{v_i, \ldots, v_k\}$ where v_i and v_k are the terminal nodes in the path. It is a non-empty subgraph or graph consisting of nodes in the form of

$$V^* = \{v_i, \ldots, v_k\} \tag{3}$$

$$E^* = \{v_i v_{i+1}, v_{i+1} v_{i+2}, \ldots, v_{k-1} v_k\} \tag{4}$$

where $V^* \subseteq V$ and $E^* \subseteq E$. Since a path is a natural seguence of its vertices then $v_i, \ldots, v_k$ and $v_k, \ldots, v_i$ denote the same thing (Diestel, 2000). Since we are considering a connected graph then, there exist at least one path between any two pairs of nodes in the graph.

2.2.3 Tree and Spanning Tree in a Graph G

A tree T is a connected graph without any cycle. A tree with n nodes has $n - 1$ edges. For any two nodes in the tree T there exist a unique path P linking the two nodes. Since a tree is minimally connected then for every edge $e \in T$, $T - e$ is disconnected. Also, for any two non-adjacent vertices v_i, $v_k \in T$, T plus the edge connecting v_i and v_k is cyclic because T is maximally acyclic (Diestel, 2000). The spanning tree of the connected graph G is a tree in G which contains all nodes of G. Note that a graph can have many spanning trees.

2.2.4 Minimum Spanning Tree of a Graph G

A minimum spanning tree of a graph G is a spanning tree whose weight is minimum among all spanning trees of the graph G. It is the shortest spanning tree with the least total weight of all edges among all possible spanning trees of the graph G (Morris et al., 1986). Assuming that w_i is the weight associated with the edge e_i in E, we can define an MST to be $T = (V, E')$, $E' \subseteq E$ such that $\sum_{i=1}^{m} w_i$ is minimum for $e_i \in E'$ (Cheriton and Tarjan, 1976).

2.2.5 Segmentation Criteria

Pixels within the region of interest (ROI) have relatively similar intensity values, so the edges connecting nodes in the ROI have relatively small weights differences. Likewise, pixels in the background have relatively similar intensity values, so the edges connecting nodes in the background have relatively small weights differences. Then, a higher difference is expected at the boundary of the ROI and the background. We use this fact to establish that there is a boundary between the ROI and the background.

Let R_1 and R_2 be regions each containing several vertices in the MST. Let v_i and v_j be vertices in the regions R_1 and R_2, respectively. The boundary between R_1 and R_2 is defined by the edge with the maximum weight in the path P connecting $v_i \in R_1$ and $v_j \in R_2$. Then, the segmentation criterion is

$$Bd(R_1, R_2) = \max_{e_k \in P} w(e_k) \tag{5}$$

where, $Bd(R_1, R_2)$ represents the boundary between regions R_1 and R_2, $w(e_k)$ is the weight of edge e_k. We implement the segmentation method and produce results by considering three situations. The first situation is when the regions R_1 and R_2 are loosely connected (the boundary is clear). The second situation is when regions R_1 and R_2 are connected (the boundary is not clear but it exists). The third situation is when the regions R_1 and R_2 are strongly connected (there is no boundary).

2.3 Implementation Steps

2.3.1 Construction of Minimum Spanning Tree

The input image is preprocessed by applying a Gaussian filter to reduce or remove artifacts because it is important to reduce the influence of noise in the computed edges' weights. Edges of the graph are computed together with corresponding weights in which each vertex represents a pixel from the image. The weight of each edge is computed by taking the absolute difference of two neighboring nodes. We construct a sparse graph from which an MST is constructed. We use Scipy package (Virtanen et al., 2020) to compute both the graph and its MST. The Kruskal algorithm is used for computing the MST.

2.3.2 Interactive Part

Vertices Identification

The ROI is identified by visual inspection. Then a node is selected from the ROI and another from the background. To simplify this

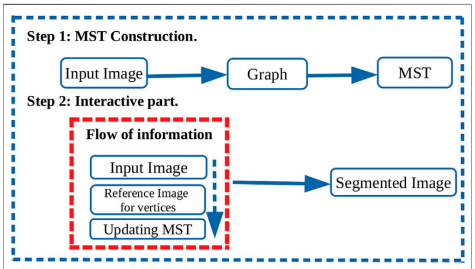


FIGURE 1 | The schematic flow diagram shows the steps involved in the segmentation process. Step 1: The image is filtered using a Gaussian filter and then weights are computed for the pixel neighbor graph. Then, the minimum spanning tree (MST) is constructed from the graph. Step 2: The region of interest (ROI) is segmented interactively. The ROI and the background are selected by clicking. Then, the MST is updated to separate the ROI from the background. New labels are assigned and reshaped back to the shape of the input image. The reshaped labels are visualized to show the segmented ROI from the background.

step, we create a graphical user interface for selecting nodes by clicking. Also, we create a reference image for storing the vertices in their respective positions. We bind the reference image behind the input image so that when a pixel is clicked on the input image the pixel coordinate extracts a node in the reference image. To identify the nodes, click once inside the ROI and then the background. For efficiency, the point selected as background should be close to the ROI. We use a Python standard graphical user interface package, Tkinter for creating the interactive window (Lundh, 1999).

Separating the ROI and the Background

The identified vertices are used for generating a path from the MST. We use the breadth-first search tree to find the path from the MST. We use Scipy package to extract the path in the MST connecting the two vertices (Virtanen et al., 2020). We run a script (only on the path) to search and identify the edge with the maximum weight and remove it from the MST. This step updates the MST by splitting it into subtrees. The updated MST is mapped back to the image to visualize the final segmentation. The steps in the interactive part can be performed repeatedly to separate more regions of interest and the MST will continue updating. The steps are summarized in **Figure 1**.

3 RESULTS

3.1 Steps of Segmenting the Region of Interest

We used the MST approach to interactively segment brain tumors from different brain regions (**Figure 2**). The first data set provided in . *mat* files was converted into *.jpg* format. The images were preprocessed using sigma = 0.1, except for a few images that needed a higher value. For the interactive part, the image was converted using Python Imaging Library (PIL) to give it a format compatible with Tkinter (Umesh, 2012). The interactive window was used to select the ROI and background (**Figures 2B,C**), which resulted in tumor segmentation (**Figure 2D**).

3.2 Segmentation Results: Region of Interest is Well Separated From the Background

Images from the first data set were used to evaluate the MST approach when segmenting ROIs loosely connected to the background (**Figure 3**). We followed the steps described in (**Figure 2**) to segment the brain tumors. In the instances/cases where the ROIs are loosely connected to the background, a sigma value of 0.1 is used in the preprocessing step to obtain the segmentation.

Figure 4 presents segmentation results obtained after segmenting MRI brain images without tumors. The images are from T1W simulated brain MRI volume. They are axial T1-weighted MRI images filtered by tuning a sigma = 0.3. To segment the brain from each image we undergo the same steps as described in (**Figure 2**) but in this case, we select the brain as the ROI and the skull as the background.

3.3 Segmentation Results: Region of Interest is Not Well Separated From the Background

The tumor's location in the brain is one of the obstacles that hinder the segmentation process. Filtering can improve the boundary between the tumor and the neighboring tissues. **Figure 5** presents results for the tumor segmented by tuning different values of sigma.

Figure 6 shows how the input image in **Figure 5** can be efficiently segmented by using the proposed method without tuning different values of sigma. The image is filtered by using sigma = 0.1 and then the brain tumor is segmented interactively without changing values of sigma.

Another example of segmenting a tumor interactively without tuning different values of sigma is presented in **Figure 7**.

3.4 Performance Analysis

We use the Jaccard Index, Dice similarity Coefficient, Sensitivity, and Specificity of the binary label to evaluate the method's performance. Let L be the set of labels in the MRI 2D slice. We classify them into labels representing the object of interest and the background. We binarize the set of labels into two unique labels such that

$$L(x,y) = \begin{cases} 0 \text{ is a background label at position } (x,y) \\ 1 \text{ is the object of interest label at position } (x,y) \end{cases}$$

Let L_T be the binarized labels in a ground truth brain tumor mask and L_P be the binarized labels in a predicted brain tumor segmented using the MST approach. We can also use the concept of true positive (TP), false positive (FP), true negative (TN) and

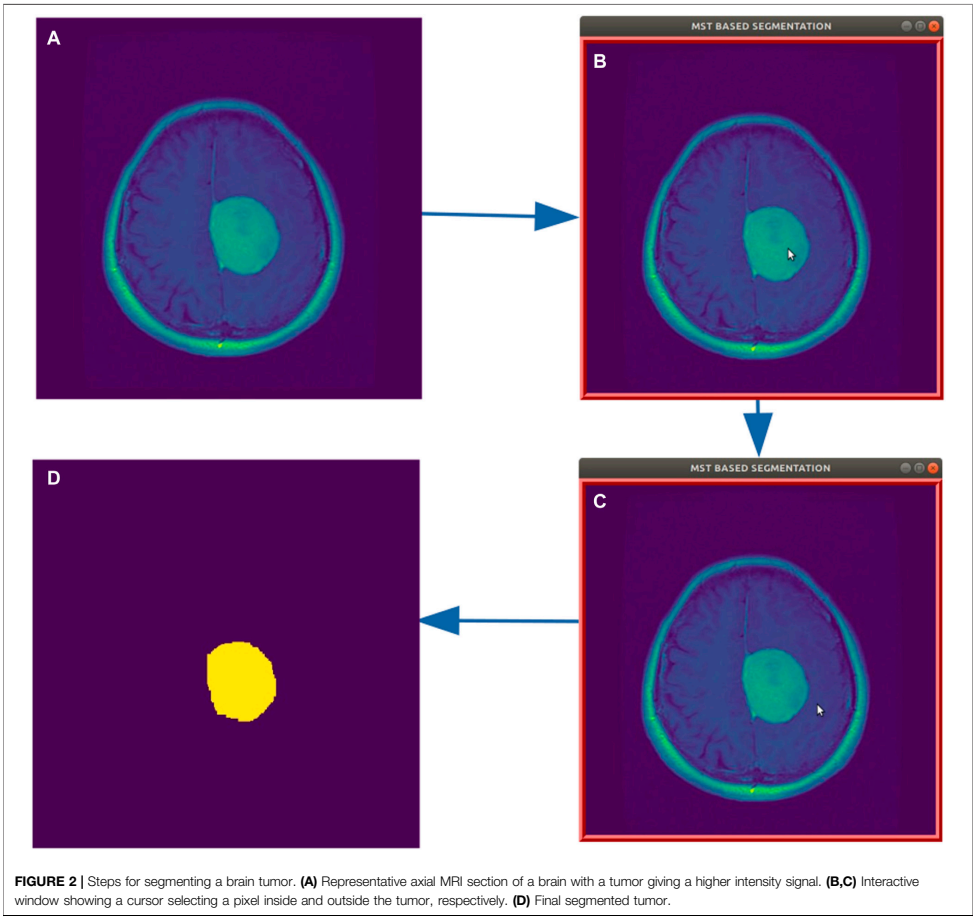


FIGURE 2 | Steps for segmenting a brain tumor. **(A)** Representative axial MRI section of a brain with a tumor giving a higher intensity signal. **(B,C)** Interactive window showing a cursor selecting a pixel inside and outside the tumor, respectively. **(D)** Final segmented tumor.

false-negative (FN) to check the performance of the method. TP represents the labels that are correctly classified as brain tumor. FP represents the labels that are incorrectly classified as brain tumor. They are not in the tumor region but classified as being in a brain tumor region. TN are labels that are correctly classified as non-tumor material, FN represents the labels that are incorrectly classified as non-tumor materials (they are tumor region labels but classified as being in a non-tumor region). The concept is paraphrased from Hua et al. (2020). The Jaccard Index (*JI*) is given by

$$JI(L_T, L_P) = \frac{|L_T(x,y) \cap L_P(x,y)|}{|L_T(x,y) \cup L_P(x,y)|} = \frac{TP}{TP + FP + FN} \quad (6)$$

The Dice Similarity Coefficient (Wang et al., 2019) is computed by using

$$DSC(L_T, L_P) = \frac{2|L_T(x,y) \cap L_P(x,y)|}{|L_T(x,y)| + |L_P(x,y)|} = \frac{2TP}{2TP + FP + FN} \quad (7)$$

We define $|L_T(x, y) \cap L_P(x, y)|$ to be the number of similar labels appearing at similar positions (x, y) in both L_T and L_P. $|L_T(x, y)|$ is the number of labels at (x, y) positions in the ground truth labels and $|L_P(x, y)|$ is the number of labels at (x, y) positions in the predicted labels. $|L_T(x, y) \cup L_P(x, y)|$ represents the number of labels which are in L_P or in L_T or in both.

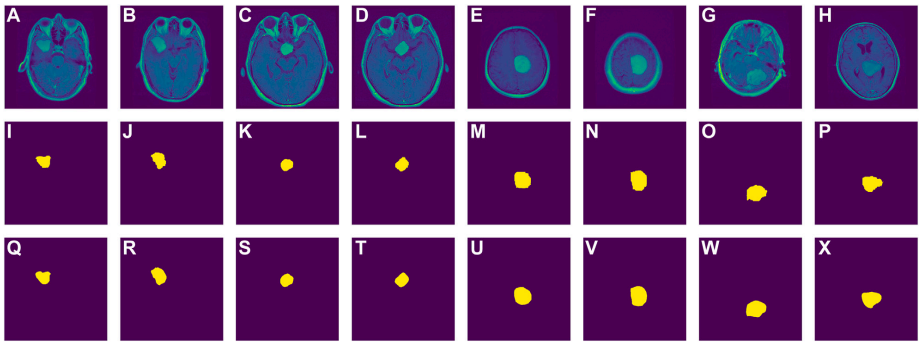


FIGURE 3 | Brain tumor segmentation of images from the first data set using minimum spanning tree. **(A–H)** Representative axial MRI sections of brains with tumors giving a higher intensity signal. **(I–P)** Segmented images **(A–H)** using minimum spanning tree. **(Q–X)** Ground truth images.

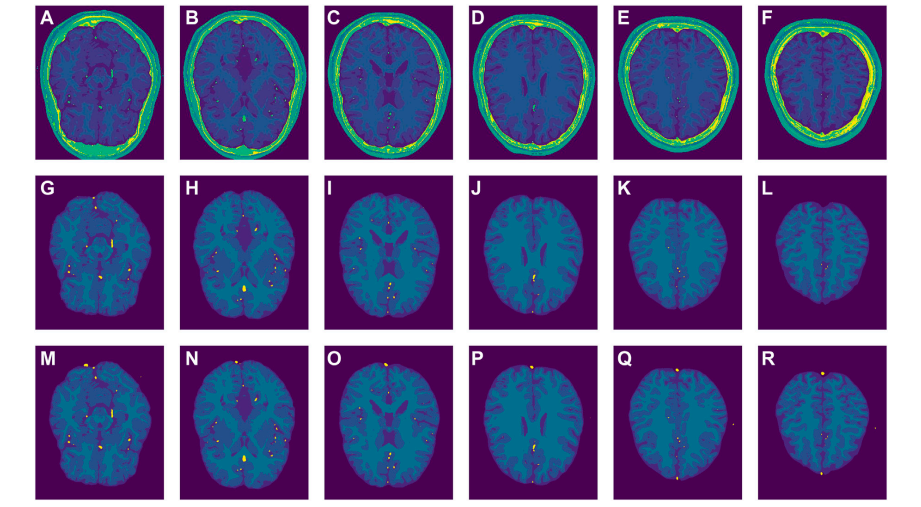


FIGURE 4 | Brain segmentation of images from the second data set using the minimum spanning tree. **(A-F)** Representative axial MRI sections of brains. **(G–L)** Segmented images **(A–F)** using the minimum spanning tree. **(M–R)** Ground truth images: obtained from the labels representing cerebrospinal fluid, gray matter, white matter and vessels.

We also compute the sensitivity and specificity which show the percentage of brain tumor and non-brain tumor voxels recognized respectively

$$Sensitivity = \frac{TP}{TP + FN}, \; Specificity = \frac{TN}{TN + FP} \tag{8}$$

We present the quantitative measurement of the segmentation accuracy by comparing the results obtained using the MST-based approach to the ground truth. We compute the Jaccard Indices, Dice Similarity coefficients, Sensitivity, and Specificity. The Jaccard indices and Dice Similarity metrics range from zero to one. A value zero means there is no overlap between the

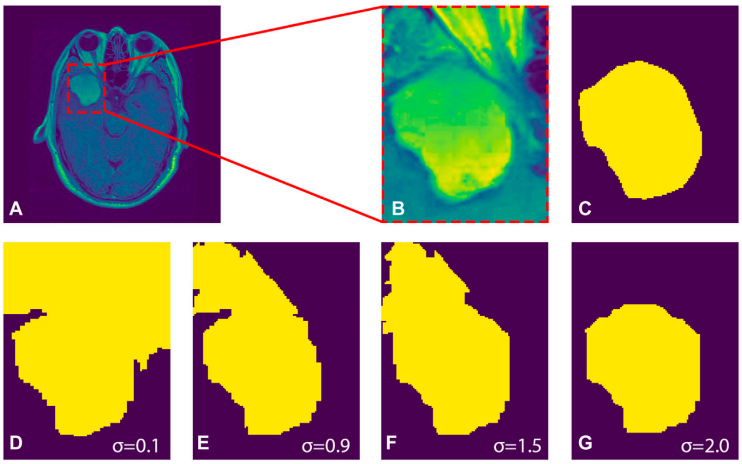


FIGURE 5 | Effect of filtering on the segmented region of interest. **(A)** Representative axial MRI section of a brain with a tumor giving a higher intensity signal. **(B)** The zoomed-in part of the image from **(A)**. **(C)** Ground truth image. **(D–G)** Segmented image when sigma is 0.1, 0.9, 1.5, and 2.0, respectively.

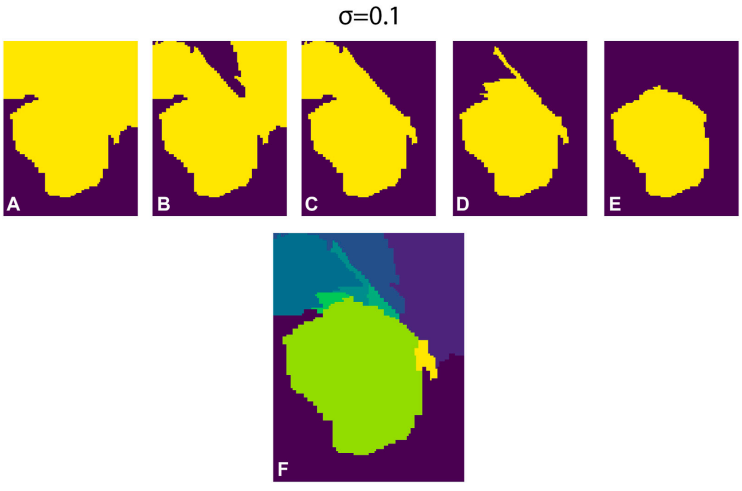


FIGURE 6 | Segmenting the tumor interactively without tuning parameters. **(A–E)** Segmenting the brain tumor interactively when sigma = 0.1. **(F)** Visualized labels of the segmented tumor and the background.

segmented region using the proposed method and the ground truth. The value 1 means there is a perfect overlap between ground truth segmentation and the one obtained using the proposed method.

We randomly sampled 300 slices from the first data set and segmented them using the proposed method and compared to the results in a preprint by Kasar et al. (2021) in **Table 2**.

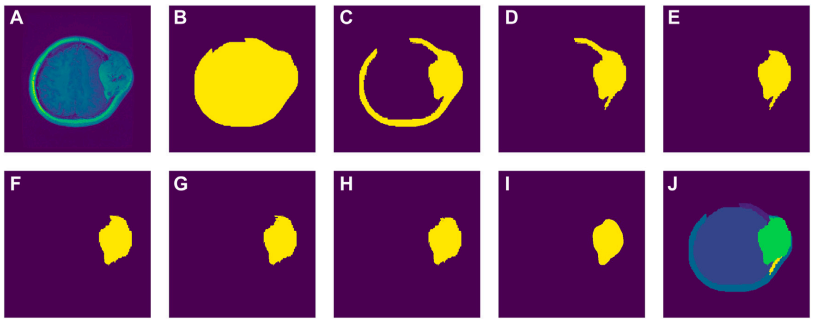


FIGURE 7 | Segmenting a brain tumor that is strongly connected to the skull. **(A)** Representative axial MRI section of a brain with a tumor giving a higher intensity signal. **(B–G)** Segmenting the brain tumor interactively sigma = 0.9. **(H)** Segmented image using minimum spanning tree. **(I)** Ground truth. **(J)** Visualized labels of the segmented tumor and the background.

Figures 8, **9** present graphs for representative images showing how filtering can affect the final segmentation. We use Jaccard indices, Dice similarity coefficients, sensitivity, and specificity as measures of similarity of the segmented ROI to the ground truth. Looking at the range of sigma in relation to the measure of similarity, it indicates that for images whose ROI is well separated from the background a value of sigma close to zero provides satisfactory results with a single click. For images whose ROI is not well separated from the background it can be challenging to obtain acceptable results because it may require additional interactive steps (see **Figures 8**, **9**).

3.4.1 Time Complexity Analysis

The time complexity for the graph construction is documented by Virtanen et al. (2020) and the time complexity construction of the MST is described (Pettie and Ramachandran, 2002). The implementation was done by writing scripts in the python programming language, and it was run on a PC processor (Core i7-8650UCPU @ 1.90GHz × 8). The input size is defined by the number of pixels in the image, corresponding to vertices in the graph. The largest size of the images segmented is 512 × 512, which gives less than 300,000 vertices. The proposed method can construct a graph and MST of more than 1,000,000 vertices in less than a second, illustrated in **Figure 10**.

4 DISCUSSION

Brain tumor segmentation is a tedious task, especially when the tumor is in brain regions with highly complex tissue structures. Some of the obstacles listed in different literature include the skin, fat, muscle, neck, and eyeballs because they hinder automatic segmentation of the brain tumors. In addition, the quality of the images could compromise the results. Here, we propose an MST-based approach for segmenting the region of interest interactively.

In this paper, we have applied the MST-based approach to segment brain tumors in human patient MRI data set and compared the results to the ground truth. **Figure 2** summarizes the steps used for segmenting the brain tumor interactively. **Figure 3** demonstrates the strength of the MST approach by comparing our results to the ground truth provided in the data set. All the slices are transverse (axial) planes from different patients showing different tumor locations. Most of the tumors in these slices have clear boundaries and can be segmented by applying a small sigma value in the preprocessing step.

Additionally, we test the approach by segmenting the brain from the non-brain tissue. The brain is segmented by clicking inside the brain and then on the skull. The MST efficiently extracts the brain referred to in **Figure 4**. We compare the obtained results to the ground truth of the brain materials. We use the known labels of the brain tissues given in the data as the ground truth and the approach gives promising results.

Also, we assess the impact of filtering the image before segmenting the ROI. The results in **Figure 5** indicate that the choice of sigma can highly influence the segmented tumor. Therefore, the parameter sigma needs to be tuned carefully, which is disadvantageous because the MST must be reconstructed every time sigma is changed. To reduce the burden of trial and error, we utilize the efficiency of the MST approach to segment the tumor without having to tune parameters repeatedly. **Figure 6** shows the results of the segmented tumor obtained interactively without changing the values of sigma. **Figure 6F** shows how labels were changing. The advantage of the interactive part is that the MST is constructed only once and we keep updating the same MST. The MST approach requires a boundary between the ROI and the background. A single step is enough to segment the tumor for images whose regions of interest are well separated from the background. The method works for images with weak boundaries between the region of interest

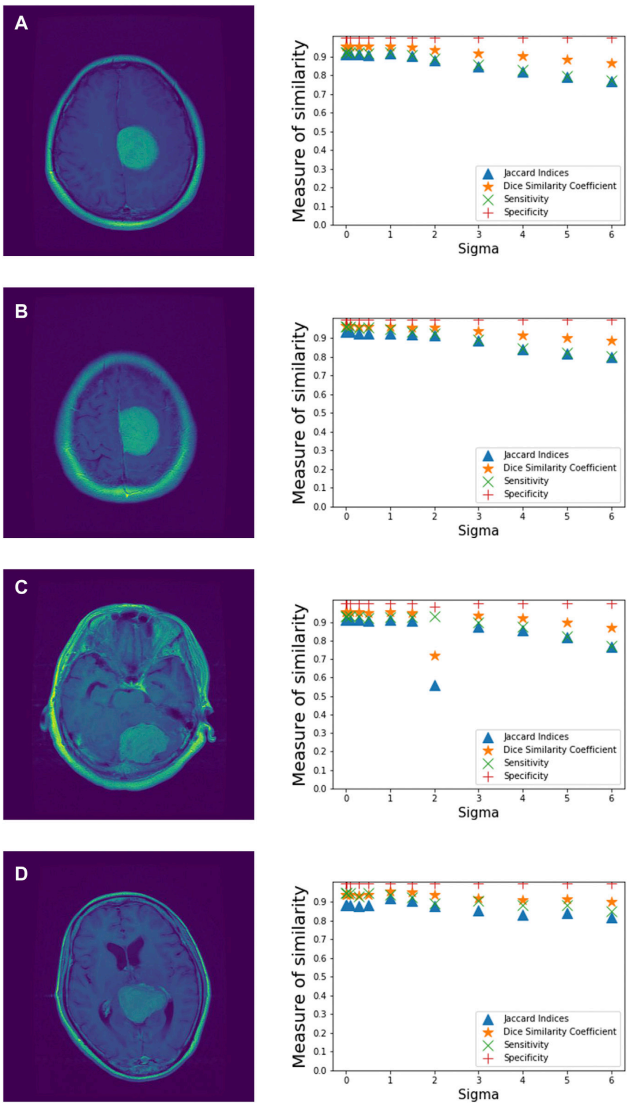


FIGURE 8 | Influence of sigma on the performance of the minimum spanning tree-based method. **(A–D)** Images and the corresponding graphs showing measures of similarity (vertical axis) of the segmented images using the minimum spanning tree method in relation to the tuned sigma (horizontal axis). The sigma values tested include: 0, 0.01, 0.1, 0.3, 0.5, 1.0, 1.5, 2.0, 3.0, 4.0, 5.0, and 6.0.

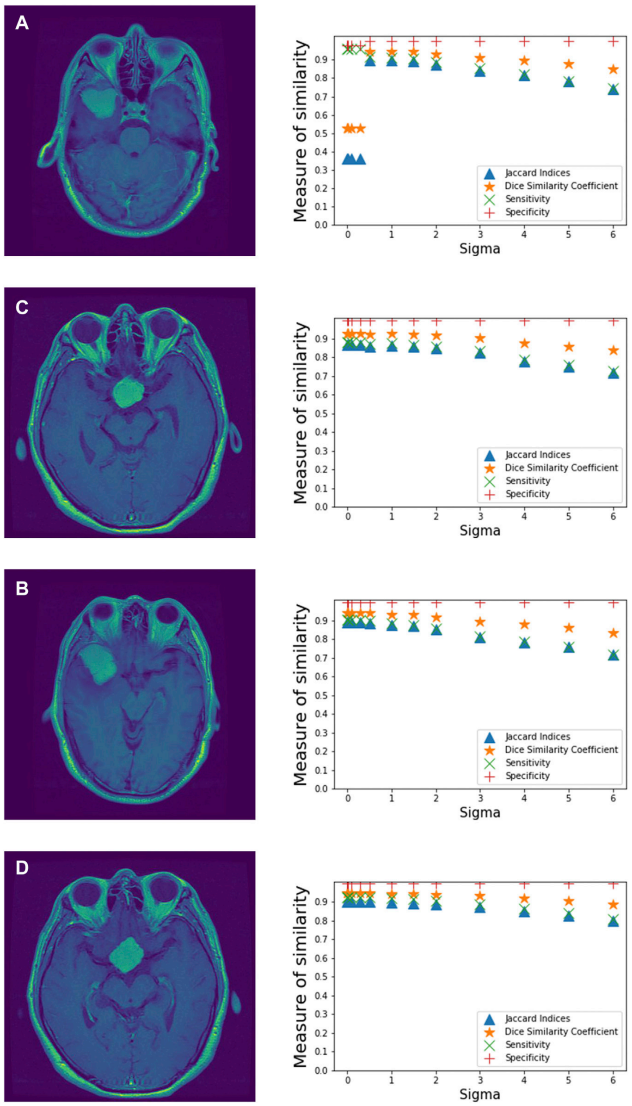


FIGURE 9 | Influence of sigma on the performance of the minimum spanning tree-based method. **(A–D)** Images and the corresponding graphs showing measures of similarity (vertical axis) of the segmented images using the minimum spanning tree method in relation to the tuned sigma (horizontal axis). The sigma values tested include: 0, 0.01, 0.1, 0.3, 0.5, 1.0, 1.5, 2.0, 3.0, 4.0, 5.0, and 6.0.

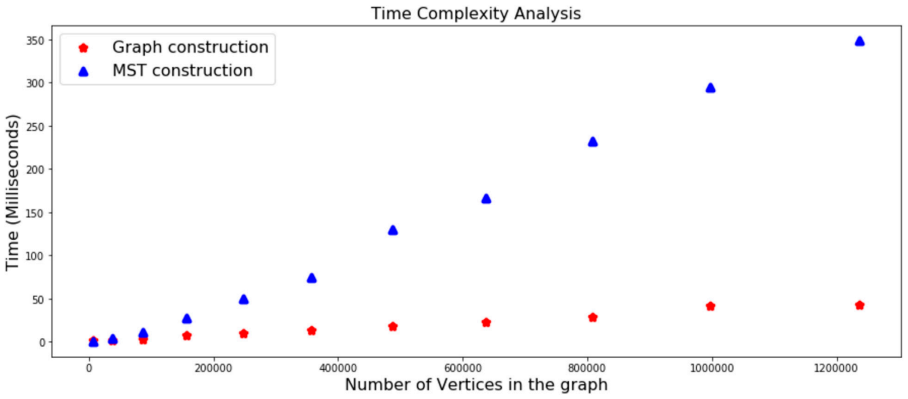


FIGURE 10 | Time complexity analysis. The time spent on graph construction is less than the time spent on minimum spanning tree (MST) construction. The computation time increases based on the size of the input image.

TABLE 1 | The performance of the minimum spanning tree approach based on data set number two, presented using Jaccard Indices, Dice Similarity Coefficients, Sensitivity, and Specificity.

Slice	Jaccard indices	Dice similarity coefficients	Sensitivity	Specificity
01	0.998 880	0.999 439	0.998 931	0.999 949
02	0.999 362	0.999 681	0.999 422	0.999 930
03	0.999 067	0.999 533	0.999 104	0.999 960
04	0.999 185	0.999 592	0.999 199	0.999 987
05	0.998 512	0.999 255	0.998 527	0.999 988
06	0.998 091	0.999 044	0.998 108	0.999 989

and the background, but one may need more interactive steps to segment the tumor. If the region of interest is strongly connected to the background, meaning that there is no boundary between them, the method will fail.

Further, we present results for the tumor being at a complicated location in **Figure 7**. We visualize the results in each step showing how the tumor was being detached from other brain materials and non-brain materials. We apply a sigma of 0.9 in the preprocessing step and then construct the MST. We run the interactive part and isolate the tumor by continuously splitting the MST. The approach segments the tumor efficiently.

In each of the segmented results, we provide a summary for the performance analysis of the MST. We use Jaccard indices, Dice similarity coefficients, sensitivity, and specificity to evaluate the performance of the MST-based approach. The results are summarized in **Tables 1**, **2**. In **Table 2** we compare the results obtained by using the MST-based method to the results obtained by Kasar et al. (2021) using the same data set. Also, **Figures 8**, **9** present results by testing different values of sigma that can give a good performance of the approach. The results indicate that for images whose ROI is well separated from the background, a sigma value close to zero can give good results. For images whose tumors are in complex locations, tuning parameters in the preprocessing step can be challenging.

In the analysis, we only used a Gaussian filter. However, other filters can be tested to check if they can give better results, depending on the quality of the image and the complexity of the tumor's location. The implementation was done by writing scripts in the python programming language. **Figure 10** gives the time complexity analysis for graph and MST construction. The link for the code used is available in the data availability section, as well as the links for the data used in this paper. As pointed out in the gland challenge of image processing, the availability of large public datasets is highly desirable Dufaux (2021) especially with their gold standard for testing different algorithms. Most freely available image data sets do not contain ground truth segmentation.

TABLE 2 | The performance of the minimum spanning tree approach compared to UNET and SEGNET based on data set number one, presented using average in each measure of similarity.

Method	Jaccard indices	Dice similarity coefficients	Sensitivity	Precision	Specificity
MST based	0.744 286	0.846 947	0.819 194		0.998 502
UNET	—	0.760 000		0.900 000 0	—
SEGNET	—	0.670 000		0.900 000 0	—

DATA AVAILABILITY STATEMENT

The links for the datasets analyzed for this study can be found in (Cheng, 2017) (first data set) and (Aubert-Broche et al., 2006; Cocosco et al., 1997) (second data set). The code used for producing the result is freely available here https://github.com/simeonmayala/minimum-spanning-tree-segmentation.

AUTHOR CONTRIBUTIONS

All authors conceptualized the study and made significant contributions to the manuscript. SM conducted the analysis to produce the results and put forward the proposed draft of the manuscript; IH, JH, and SA contributed in writing, editing, creation of figures, and provided knowledge support of the biological concepts. SG and MB supervised the work including writing, reviewing, and editing.

FUNDING

The project is funded by the University of Bergen, Bergen, Norway.

ACKNOWLEDGMENTS

The preprint version of (Kasar et al., 2021) was included for data comparison.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frsip.2022.816186/full#supplementary-material

REFERENCES

Aubert-Broche, B., Evans, A. C., and Collins, L. (2006). A New Improved Version of the Realistic Digital Brain Phantom. *NeuroImage* 32, 138–145. doi:10.1016/j.neuroimage.2006.03.052

Banerjee, S., Mitra, S., and Uma Shankar, B. (2016). Single Seed Delineation of Brain Tumor Using Multi-Thresholding. *Inf. Sci.* 330, 88–103. doi:10.1016/j.ins.2015.10.018

Cheng, J. (2017). *Brain Tumor Dataset*. doi:10.6084/m9.figshare.1512427.v5

Cheng, J., Huang, W., Cao, S., Yang, R., Yang, W., and Yun, Z. (2015). Enhanced Performance of Brain Tumor Classification via Tumor Region Augmentation and Partition. *PloS one* 10, e0140381. doi:10.1371/journal.pone.0140381

Cheng, J., Yang, W., Huang, M., Huang, W., Jiang, J., Zhou, Y., et al. (2016). Retrieval of Brain Tumors by Adaptive Spatial Pooling and fisher Vector Representation. *PloS one* 11, e0157112. doi:10.1371/journal.pone.0157112

Cheriton, D., and Tarjan, R. E. (1976). Finding Minimum Spanning Trees. *SIAM J. Comput.* 5, 724–742. doi:10.1137/0205051

Ciesielski, K. C., and Udupa, J. K. (2011). A Framework for Comparing Different Image Segmentation Methods and its Use in Studying Equivalences between Level Set and Fuzzy Connectedness Frameworks. *Computer Vis. Image Understanding* 115, 721–734. doi:10.1016/j.cviu.2011.01.003

Cocosco, C. A., Kollokian, V., Kwan, R. K.-S., Pike, G. B., and Evans, A. C. (1997). "Brainweb: Online Interface to a 3d Mri Simulated Brain Database," in *NeuroImage (Citeseer)*.

Dale, A. M., Fischl, B., and Sereno, M. I. (1999). Cortical Surface-Based Analysis: I. Segmentation and Surface Reconstruction. *Neuroimage* 9, 179–194. doi:10.1006/nimg.1998.0395

Diestel, R. (2000). *Graph Theory*. electronic edition. new york: Springer-Verlag, 6–12.

Dufaux, F. (2021). Grand Challenges in Image Processing. *Front. Signal Process.* 1, 3.

Felzenszwalb, P. F., and Huttenlocher, D. P. (2004). Efficient Graph-Based Image Segmentation. *Int. J. Comput. Vis.* 59, 167–181. doi:10.1023/b:visi.0000022288.19776.77

Hahn, H. K., and Peitgen, H.-O. (2000). "The Skull Stripping Problem in Mri Solved by a Single 3d Watershed Transform," in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Springer), 134–143. doi:10.1007/978-3-540-40899-4_14

Hua, R., Huo, Q., Gao, Y., Sui, H., Zhang, B., Sun, Y., et al. (2020). Segmenting Brain Tumor Using Cascaded V-Nets in Multimodal Mr Images. *Front. Comput. Neurosci.* 14, 9. doi:10.3389/fncom.2020.00009

Huang, H., Yang, G., Zhang, W., Xu, X., Yang, W., Jiang, W., et al. (2021). A Deep Multi-Task Learning Framework for Brain Tumor Segmentation. *Front. Oncol.* 11. doi:10.3389/fonc.2021.690244

Iglesias, J. E., Liu, C.-Y., Thompson, P. M., and Tu, Z. (2011). Robust Brain Extraction across Datasets and Comparison with Publicly Available Methods. *IEEE Trans. Med. Imaging* 30, 1617–1634. doi:10.1109/tmi.2011.2138152

Kalavathi, P., and Prasath, V. S. (2016). Methods on Skull Stripping of Mri Head Scan Images—A Review. *J. digital Imaging* 29, 365–379. doi:10.1007/s10278-015-9847-8

Kang, B. (2021). "Exploring Graph-Based Neural Networks for Automatic Brain Tumor Segmentation," in From Data to Models and Back: 9th International Symposium, DataMod 2020, Virtual Event, October 20, 2020 (Springer Nature), 18. Revised Selected Papers.12611

Kasar, P. E., Jadhav, S. M., and Kansal, V. (2021). *Mri Modality-Based Brain Tumor Segmentation Using Deep Neural Networks*. doi:10.21203/rs.3.rs-496162/v1

Long, X., and Sun, J. (2020). Image Segmentation Based on the Minimum Spanning Tree with a Novel Weight. *Optik* 221, 165308. doi:10.1016/j.ijleo.2020.165308

Lundh, F. (1999). An Introduction to Tkinter. Available at:www.pythonware.com/library/tkinter/introduction/index.htm

Morris, O., Lee, M. d. J., and Constantinides, A. (1986). Graph Theory for Image Analysis: an Approach Based on the Shortest Spanning Tree. *IEE Proc. F (Communications, Radar Signal Processing) (Iet)* 133, 146–152. doi:10.1049/ip-f-1.1986.0025

Nandi, A. K. (2021). Apropos of Signal Processing. *Front. Signal Process*. doi:10.3389/frsip.2021.686341

Pettie, S., and Ramachandran, V. (2002). An Optimal Minimum Spanning Tree Algorithm. *J. ACM (Jacm)* 49, 16–34. doi:10.1145/505241.505243

Roy, S., and Maji, P. (2015). "A Simple Skull Stripping Algorithm for Brain Mri," in 2015 Eighth International Conference on Advances in Pattern Recognition (ICAPR), 1–6. doi:10.1109/ICAPR.2015.7050671

Saglam, A., and Baykan, N. A. (2017). Sequential Image Segmentation Based on Minimum Spanning Tree Representation. *Pattern Recognition Lett.* 87, 155–162. doi:10.1016/j.patrec.2016.06.001

Ségonne, F., Dale, A. M., Busa, E., Glessner, M., Salat, D., Hahn, H. K., et al. (2004). A Hybrid Approach to the Skull Stripping Problem in Mri. *Neuroimage* 22, 1060–1075. doi:10.1016/j.neuroimage.2004.03.032

Smith, S. M. (2000). *Bet: Brain Extraction Tool. FMRIB TR00SMS2b, Oxford Centre for Functional Magnetic Resonance Imaging of the Brain)*. Headington, UK: Department of Clinical Neurology, Oxford University, John Radcliffe Hospital.

Umesh, P. (2012). Image Processing in python. *CSI Commun.* 23.

Urquhart, R. (1982). Graph Theoretical Clustering Based on Limited Neighbourhood Sets. *Pattern recognition* 15, 173–187. doi:10.1016/0031-3203(82)90069-3

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., et al. (2020). SciPy 1.0: Fundamental Algorithms for

Scientific Computing in Python. *Nat. Methods* 17, 261–272. doi:10.1038/s41592-019-0686-2

Wang, L., Wang, S., Chen, R., Qu, X., Chen, Y., Huang, S., et al. (2019). Nested Dilation Networks for Brain Tumor Segmentation Based on Magnetic Resonance Imaging. *Front. Neurosci.* 13, 285. doi:10.3389/fnins.2019.00285

Xu, Y., and Uberbacher, E. C. (1997). 2d Image Segmentation Using Minimum Spanning Trees. *Image Vis. Comput.* 15, 47–57. doi:10.1016/s0262-8856(96)01105-5

Zahn, C. T. (1971). Graph-theoretical Methods for Detecting and Describing Gestalt Clusters. *IEEE Trans. Comput.* 100, 68–86. doi:10.1109/t-c.1971.223083

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2022 Mayala, Herdlevær, Haugsøen, Anandan, Gavasso and Brun. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

Paper C

GUBS: Graph-based Unsupervised Brain Segmentation in MRI images.

Simeon Mayala, Ida Herdlevær, Jonas Bull Haugsøen, Shamundeeswari Anandan, Nello Blaiser, Sonia Gavasso, and Morten Brun

MDPI Journal of imaging (2022).

Journal of *Imaging*

MDPI

Article

GUBS: Graph-Based Unsupervised Brain Segmentation in MRI Images

Simeon Mayala [1,*], Ida Herdlevær [2,3], Jonas Bull Haugsøen [2,3], Shamundeeswari Anandan [2,3], Nello Blaser [4], Sonia Gavasso [2,3] and Morten Brun [1]

[1] Department of Mathematics, University of Bergen, 5020 Bergen, Norway
[2] Department of Clinical Medicine, University of Bergen, 5021 Bergen, Norway
[3] Neuro-SysMed, Department of Neurology, Haukeland University Hospital, 5053 Bergen, Norway
[4] Department of Informatics, University of Bergen, 5020 Bergen, Norway
* Correspondence: simeon.mayala@uib.no

Abstract: Brain segmentation in magnetic resonance imaging (MRI) images is the process of isolating the brain from non-brain tissues to simplify the further analysis, such as detecting pathology or calculating volumes. This paper proposes a Graph-based Unsupervised Brain Segmentation (GUBS) that processes 3D MRI images and segments them into brain, non-brain tissues, and backgrounds. GUBS first constructs an adjacency graph from a preprocessed MRI image, weights it by the difference between voxel intensities, and computes its minimum spanning tree (MST). It then uses domain knowledge about the different regions of MRIs to sample representative points from the brain, non-brain, and background regions of the MRI image. The adjacency graph nodes corresponding to sampled points in each region are identified and used as the terminal nodes for paths connecting the regions in the MST. GUBS then computes a subgraph of the MST by first removing the longest edge of the path connecting the terminal nodes in the brain and other regions, followed by removing the longest edge of the path connecting non-brain and background regions. This process results in three labeled, connected components, whose labels are used to segment the brain, non-brain tissues, and the background. GUBS was tested by segmenting 3D T1 weighted MRI images from three publicly available data sets. GUBS shows comparable results to the state-of-the-art methods in terms of performance. However, many competing methods rely on having labeled data available for training. Labeling is a time-intensive and costly process, and a big advantage of GUBS is that it does not require labels.

Keywords: brain tissues; non-brain tissues; segmentation; minimum spanning tree

check for updates

Citation: Mayala, S.; Herdlevær, I.; Haugsøen, J.B.; Anandan, S.; Blaser, N.; Gavasso, S.; Brun, M. GUBS: Graph-Based Unsupervised Brain Segmentation in MRI Images. *J. Imaging* **2022**, *8*, 262. https://doi.org/10.3390/jimaging8100262

Academic Editor: Caroline Petitjean

Received: 17 August 2022
Accepted: 22 September 2022
Published: 27 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The brain is a complex organ that makes the central nervous system together with the spinal cord. It is divided into forebrain (sensory processing, higher reasoning), midbrain (motor movement, audio/visual processing), and hindbrain (autonomic functions such as sleep and respiratory rhythms). Over the past years, non-invasive imaging techniques have gained momentum, in assessing brain injury and studying brain anatomy. In particular, magnetic resonance technology is widely used in the diagnosis of brain diseases such as brain tumors, multiple sclerosis, hematomas and to find the cause of conditions such as epilepsy and headaches [1]. The technology produces magnetic resonance imaging (MRI) data, which can be processed to produce 3D volumetric data with the intensity of voxels varying according to the properties of different tissues. MRI images are most commonly presented as a stack of two-dimensional slices. Analysis of such high quality complex MRI data is a tedious process. Recent advances with computer aided-tools have overcome the major pitfalls of manual MRI data analysis. Brain MRI segmentation is an important step in image processing as it highly influences the outcome of the entire analysis, which is

crucial in the case of surgical planning, delineating lesions and image-guided interventions. To segment any target structure in the brain, it is common to perform a preprocessing step to isolate the brain from non-brain tissues such as the skull, dura and scalp [2,3].

Methods for segmenting the brain from non-brain tissues can be classified as manual, semi-automated, and fully automated [4,5]. Manual brain extraction gives high precision but is labor-intensive [6]. Semi-automated methods involve a certain degree of user intervention, whereas automated methods do not depend on any human interaction. Most automated methods for brain extraction can be classified in categories, such as mathematical morphology-based, intensity-based, deformable surface-based, atlas-based, and hybrid methods [2,7,8]. Machine learning techniques, including neural networks are also widely used for skull stripping [9,10]. We review different skull stripping techniques before describing GUBS.

Louis Lemieux et al. [11] proposed a fully automated method for segmenting the whole brain in T1- weighted volume. It is fast and based on foreground thresholding and morphological operations. It performs 3D connected component analysis at each level. Furthermore, Brain Extraction Tool (BET) is an automated method for segmenting Magnetic Resonance head images to separate brain and non-brain tissues. It uses a deformable model that evolves to fit the brain's surface by applying a set of locally adaptive model forces. The method is robust, fast and it does not require any pre-registration or preprocessing [12].

FreeSurfer is an open-source software that implements various image processing tools for both structural and functional MRI data sets. Skull stripping is one of the tasks that FreeSurfer provides for processing MRI images. It performs automatic skull stripping from intensity normalized images through a deformation of a tessellated ellipsoidal template into the shape of the inner surface of the skull [13]. David W. Shattuck and Richard M. Leahy [14] present an MRI analysis tool that produces cortical surface representations with spherical topology from MR images of the human brain. One of the tools that include skull stripping is a brain surface extractor (BSE). It breaks connections between the brain and non-brain tissues by using a morphological erosion operation, and then it identifies the brain using a connected component operation followed by a dilation operation to undo the effects of the erosion [2]. The final step is closing small holes that may occur in the brain surface.

The work of [15] presents a skull-stripping algorithm based on a hybrid approach (HWA). It combines watershed algorithms and deformable surface models. The method uses a localized voxel in T1-weighted white matter to create a global minimum in the white matter. The global minimum is created before applying the watershed algorithm with a pre-flooding height. The Robust Brain Extraction (ROBEX) is another method for performing skull stripping. It is a learning-based brain extraction system that combines discriminative and generative models. It is trained to detect brain boundaries using the random forest approach and ensures that the result is plausible. To obtain the final segmentation, the imperfect shape presented by the model is refined by using a graph cut [16]. The work presented in [17] proposed an automatic skull stripping method based on deformable models and histogram analysis named Simplex Mesh and Histogram Analysis Skull Stripping (SMHASS) method. It defines the starting point for deformation using a rough segmentation based on thresholds and morphological operators. The deformable model is based on a simplex mesh, whereas its deformation is controlled by local gray levels of the image and the information obtained on the gray level modeling of the rough-segmentation.

A simple skull stripping algorithm (S3) is proposed in [18]. The method is based on brain anatomy and image intensity characteristics. It is a knowledge-based algorithm and works by using adaptive intensity thresholding and then morphological operations. Oeslle Lucena et al. [9] proposed a Convolutional Neuron Network (CNN) for brain extraction in MRI trained with "silver standard" masks. The method generates silver standard masks which are used as inputs by using the Simultaneous Truth and Performance Level Estimation (STAPLE) algorithm and then implementing a tri-planar method using parallel 2D U-Net-based CNNs, named as CONSNet. Anam Fatima et al. [19] proposed a skull

stripping method and evaluated it on a 2D slice-based multi-view U-Net (MVU-Net) architecture. It performs as well as a 3D model while using fewer computational resources. In [20], a Single-Input Multi-Output U-Net (SIMOU-Net) was developed for segmenting foetal brain. Different from the original U-Net, the SIMOU-Net has a deeper architecture and takes account of the features extracted from each side output. Furthermore, Ref. [6] proposed a 3D-UNet for skull stripping to address the entire brain extraction problem satisfactory for diverse datasets. The work in [21] proposed a graph-based method for skull stripping, which uses intensity thresholding on the input image to obtain a preliminary mask. Finally, the method removes narrow connections using graph cuts followed by post-processing.

In this paper, we propose a graph-based approach that represents an MRI image using a graph, before segmenting the brain. The method uses the minimum spanning tree (MST) of the constructed adjacency graph to separate the brain, non-brain tissues, and the background. The approach involves the following steps; preprocessing of the MRI image, sampling points within and outside the brain, MST construction from the graph, isolating brain, non-brain tissues, and the background, followed by post-processing. The general idea of using a graph-based method for skull stripping relates to the ideas of [21]. We define an image into adjacency graph in the same way, but our approach differs significantly regarding the segmentation criteria. We perform skull stripping using the MST of the graph, in which each vertex has a minimal number of connections instead of using the whole graph. Different from our approach, the method presented by [21] depends on an estimate of the region bounded by white matter obtained by region growth. We define edge weights differently, and our approach does not depend on region growth. The method presented by [21] uses an initial mask obtained by thresholding and uses a graph cut to disconnect the narrow connections. GUBS depends on the MST constructed from the adjacency graph and separates the brain, non-brain tissues, and the background by disconnecting paths connecting these regions.

2. Materials and Methods

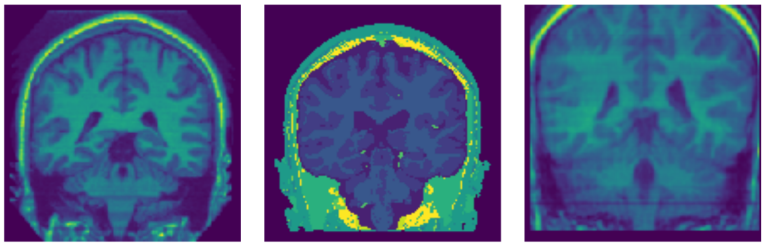
2.1. Materials

In this paper, we analyzed three different data sets. The first data set is from the Open Access Series of Imaging Studies (OASIS). It consist of T1W images from 77 subjects with an isotropic voxel resolution of 1.0 mm and a shape of $176 \times 208 \times 176$ [22]. The ground truth segmentation provided with this data set was created using a custom method based on registration to an atlas and then revised by human experts. Twenty of the subjects in this dataset suffered from Alzheimer's disease, a degenerative disease characterized by the loss of brain tissue [9,16].

The second data set consists of 20 simulated T1W MRI images mimicing healthy brains collected from the BrainWeb (BW) database [23,24]. They are provided as *.MINC* format anatomical models consisting of a set of 3D tissue membership volumes, one for each tissue class. Each label at a voxel in the anatomical model represents the tissue that contributes the most to that voxel. They have dimensions of $362 \times 434 \times 362$ and have a 0.5 mm isotropic voxel size.

The third data set consists of T1W MRI images from 18 subjects provided to the Internet Brain Segmentation Repository (IBSR) by the Center for Morphometric Analysis at Massachusetts General Hospital [25]. The images are stored in the NIfTI format. The shape of the images is $256 \times 256 \times 128$ and the voxel resolution is 0.94 mm $\times$ 0.94 mm $\times$ 1.5 mm [26]. The database contains a manual segmentation of the gray matter (GM), white matter (WM) and cerebral spinal fluid (CSF), as well as skull stripped images.

Figure 1 shows coronal sections of brains sampled from a 3D MRI of a representative subject from each of the three data sets.



(a) (b) (c)

Figure 1. Sampled Slices: (**a**) Sampled slice coronal section from a 3D MRI volume OASIS data set, (**b**) Sampled slice coronal section from a 3D MRI volume BW data set, (**c**) Sampled slice coronal section from a 3D MRI volume IBSR data set. Notice the differences and the quality of data sets.

2.2. Methods

In this section, a graph-based method for brain segmentation using MST is presented. An adjacency graph is constructed from a preprocessed 3D MRI and then its MST is applied for separating the brain, non-brain tissues, and the background. The summarized steps for brain segmentation are presented in Figure 2.

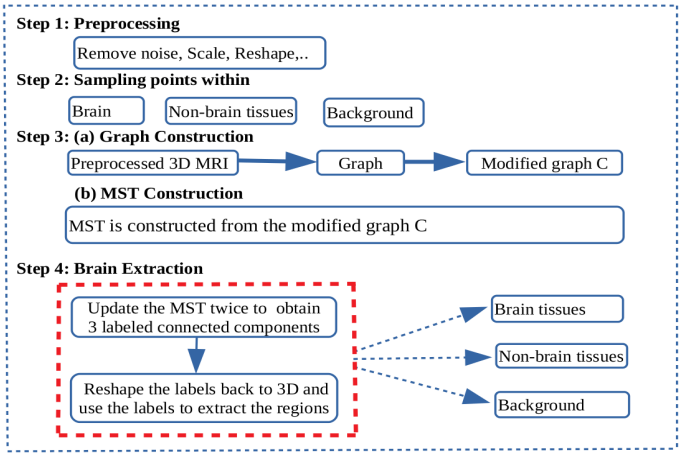


Figure 2. Schematic diagram: Flow diagram showing the steps for brain extraction. Step 1: Preprocessing to remove noise, scale values in the range of 0 and 1 and reshape the 3D magnetic resonance imaging (MRI) volume. Step 2: Sampling points within the brain, non-brain tissues, and the background. Step 3: An adjacency graph weighted by absolute intensity differences is constructed from the preprocessed 3D MRI volume. Then, nodes in the adjacency graph corresponding to the sampled points in step 2 are collapsed in their respective regions to form a graph C. From the modified graph C, a minimum spanning tree is constructed. Step 4: Brain segmentation. Nodes in C representing each of the regions of interest are the terminal nodes for the paths to be disconnected to separate the regions. First, the minimum spanning tree (MST) is modified by removing the edge with highest edge weight in the path connecting the representative nodes to separate the brain and non-brain subtrees. Again the MST is modified by removing the edge with highest edge weight in the path connecting the representative nodes to separate the non-brain and background subtrees. New labels are assigned and reshaped back to the shape of the 3D MRI.

Let A be an MRI image with dimension $M \times N \times L$ such that $A_{(i,j,k)}$ gives the intensity value at position (i,j,k) for $i = 0,1,\ldots,M-1$; $j = 0,1,\ldots,N-1$ and $k = 0,1,\ldots,L-1$. Voxels at position (i,j,k) and (i',j',k') are called adjacent if $(i-i')^2+(j-j')^2+(k-k')^2 = 1$. We can mathematically define

$$A = [A_{(i,j,k)}] \in \mathbb{R}^{M \times N \times L} \tag{1}$$

and each point changes along the coordinate axes with respect to the position of the adjacent voxels.

2.2.1. Preprocessing

We read all the image data sets using a *NiBabel* package that can read different common medical images file formats [27] and then retrieve a 3D image using a *nilearn* package [28]. Finally, every MRI image is convert into *.npy* using a *numpy* package [29]. Each image was resized to the size of $128 \times 128 \times 128$. The intensities were scaled in the range of 0 to 1 using

$$A^{scaled}_{(i,j,k)} = \frac{A_{(i,j,k)} - A^{\min}_{(j,k)}}{A^{\max}_{(j,k)} - A^{\min}_{(j,k)}}, \tag{2}$$

where $A^{\min}_{(j,k)} = \min_{i \in \{0,\ldots,M-1\}} \{A_{(i,j,k)}\}$ and $A^{\max}_{(j,k)} = \max_{i \in \{0,\ldots,M-1\}} \{A_{(i,j,k)}\}$. We adapt the formula and its implementation from scikit-learn [28]. Since the implementation requires only number of samples and number of features, the 3D MRI are reshaped into 2D before scaling and then reshape back to 3D after scaling. Whenever $A^{\max}_{(j,k)}$ is approximately or equal to $A^{\min}_{(j,k)}$ the situation is taken care by setting scales of near constant features to 1 to avoid division by very small number or zero values.

We remove noise and make sure that there is no spot in the background (outside the head) by removing small objects. Some data sets may need additional preprocessing steps to accelerate the separation of the regions. Since the brain is connected to the rest of the head by dark, thin segments [11], a small threshold may be applied to disconnect more the brain from the non-brain tissues. This is done by setting all values less than the threshold to 0. For IBSR and OASIS data sets, the values 0.25 and 0.32 were used in the preprocessing step, respectively. For data sets with good contrast between the brain and non-brain tissues such as BW, thresholding is unnecessary.

2.2.2. Edge Surface Detection

The edge surfaces in 3D images are defined as the structural boundaries of objects in the image. However, the true edge-surfaces of 3D images are continuous surfaces rather than discrete 3D edges like points which are detected by edge detectors [30]. These are points sampled at the surface that form the boundary between the object and the background. In this context the surface of the object is represented by intensity changes in the data volume [31]. The changes will be detected between edges.

The edges are detected by highlighting the local variation between the adjacent voxels. Example of the two adjacent voxels (i,j,k) and (i',j',k') located on the opposite side of an edge. We use the finite difference method (particularly forward difference) to compute the change between two adjacent voxels from the binary image. We define B to be the binary image of A. The change is computed by

$$\varepsilon = B_{(i',j',k')} - B_{(i,j,k)}. \tag{3}$$

For a 3D binary image the edge is detected when the changes between two neighboring voxels are

$$\varepsilon = \begin{cases} +1 & if\ B_{(i',j',k')} = 1\ and\ B_{(i,j,k)} = 0 \\ -1 & if\ B_{(i',j',k')} = 0\ and\ B_{(i,j,k)} = 1 \end{cases} \tag{4}$$

So, when ε is +1 or -1 the forward difference method asserts the presence of an edge. The edge surfaces of the 3D image consist of all points satisfying the changes in Equation (4) along a given axis. Then, the discrete 3D edge-like points represents the set of the edge points detected from 3D binary image B by satisfying the condition ε for the neighboring voxels.

In the implementation we perform two steps to obtain the discrete 3D edge-like points. Image A is binarized to obtain image B by either setting voxels to 1 if the gray value is greater than zero or setting to 0 elsewhere. We fill all possible holes in the binary image B by using a function for binary closing from Scipy package [32]. Then, we use a *diff* function from numpy package [29] to compute the discrete difference of the binary image by using the forward difference formula. Then, the points satisfying condition ε are discrete 3D edge-like points distributed at the surface of the head. These points are very useful in the next subsection when sampling points within the brain and non-brain tissues.

2.2.3. Sampling Points within the Brain, Non-Brain Tissues and Background

Let T be the set of points in the MRI image A, i.e.,

$$T = \{(i, j, k) : 0 \leq i \leq M-1, 0 \leq j \leq N-1, 0 \leq k \leq L-1\}. \tag{5}$$

Next, we will sample points T_B, T_{NB} and T_{BG} in the brain, non-brain tissues and background, respectively. We define, $T_B \subset T$, $T_{NB} \subset T$ and $T_{BG} \subset T$ such that T_B, T_{NB} and T_{BG} are mutually exclusive but not exhaustive. That is,

$$T_B \cap T_{NB} = T_{NB} \cap T_{BG} = T_B \cap T_{BG} = \emptyset \tag{6}$$

and

$$T_B \cup T_{NB} \cup T_{BG} \subsetneq T. \tag{7}$$

We do the following things to obtain points in T_B: From the discrete 3D edge-like points distributed at the surface of the head we take points above the selection line which is obtained by visual inspection (see Figure 3). Then, pull these points from the surface of the head to the brain by a certain distance $\delta_1 \in \mathbb{N}$. Then, compute the mean point of the pulled points. For each pulled point a distance is computed from the mean point and get the maximum distance. Then, the maximum value times 0.75 is the a distance threshold value for removing the points with the distance above the threshold value. This last step is performed to restrict the pulled points to lie within the brain and remove points close to the skull.

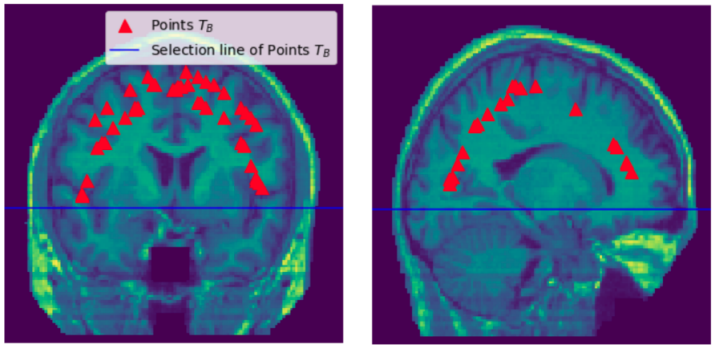


Figure 3. Sampling points T_B within the brain: (**a**) Sampled slice coronal section showing the selection line and points sampled within the brain, (**b**) Sampled slice sagittal section section showing the selection line and points sampled within the brain.

To obtain points T_{NB} in the non-brain tissues we use all the discrete 3D edge-like points distributed at the surface of the head and pull them inside towards the non-brain tissue by $\delta_2 \in \mathbb{N}$. The pulled points are rechecked if there is any point with the intensity value of zero it is removed. See Figures 4 and 5.

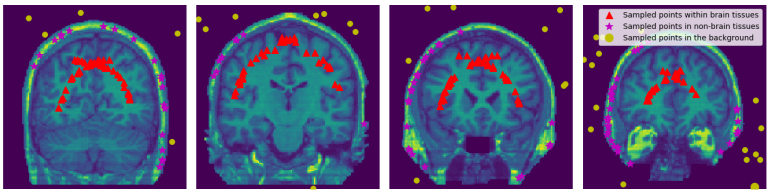


Figure 4. Sampled Points: 2D visualization of the representative coronal section from MRI image of a single subject (OASIS data) showing the sampled points within the brain, within the non-brain, and in the background region. δ_1 and δ_2 values were set to 15 and 3, respectively.

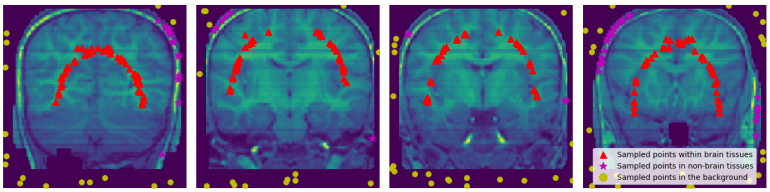


Figure 5. Sampled Points: 2D visualization of the representative coronal section from MRI image of a single subject (IBSR data) showing the sampled points within the brain region, within non-brain tissues, and in the background. δ_1 and δ_2 values were set to 15 and 3, respectively. Notice the removed part of the skull and the brain from some slices.

To obtain points T_{BG} sampled in the background we use the binary image after closing all the possible holes (this is the same binary image described in the last paragraph of Section 2.2.2).

First, we compute all points in the binary image whose points have an intensity values of 0. We also find points at the six surfaces of the 3D cube of the binary image whose points have an intensity values of 0. We combine these points to increase the chance of getting representative points from all the sides. We then sample 20,000 of these points uniformly at random to get the T_{BG} points.

2.2.4. Graph

Let (V, E) be the voxel adjacency graph constructed from A such that V is the set of nodes and E represents a set of weighted edges. Each node represents a voxel at location (i, j, k) in A, and each edge connects two adjacent voxels. The edge's weight is the absolute value of the difference between the intensity values of the voxels it connects.

2.2.5. Collapsing Nodes

Given a graph G and a subset H of the nodes in G we construct a graph where all the nodes in H have been collapsed to a single node. In order to ease the construction of this graph we represent the single collapsed node by a node $h \in H$.

The graph G_H^h where all nodes in H have been collapsed to a single node h in H is constructed as follows:

1. For every edge e in G, if both end nodes of e appear in H the edge e is discarded.
2. For every remaining edge e, if a node v in H appears in e, the edge e is modified by replacing v by h.
3. Remove all the nodes in H except the node h from the modified graph to obtain the graph G_H^h.

2.2.6. Segmentation Criteria

The central problem addressed by GUBS is to segment an image using samples of nodes from the regions of interest. From the voxels adjacency graph, we construct a new graph by successively collapsing the nodes sampled from each of the regions. The main idea of GUBS is to use the minimum spanning tree of this new graph instead of the minimum spanning tree of the voxels adjacency graph.

Given an MRI image of the brain, we want to separate the voxel adjacency graph into three regions representing the brain, non-brain tissues, and background.

Let $V_B \subset V$, $V_{NB} \subset V$ and $V_{BG} \subset V$ be sets of nodes in a graph G corresponding to sets of coordinate points T_B, T_{NB} and T_{BG} sampled from brain, non-brain tissues, and background, respectively. We construct the graphs G_B, G_{NB} and G_{BG} so that the nodes V_B, V_{NB} and V_{BG} are in G_B, G_{NB} and G_{BG}, respectively.

Let $v_B \in V_B$, $v_{NB} \in V_{NB}$ and $v_{BG} \in V_{BG}$ be randomly sampled nodes. Let $A = G_{V_B}^{v_B}$, $B = A_{V_{NB}}^{v_{NB}}$, $C = B_{V_{BG}}^{v_{BG}}$. In the modified graph C each of the sets V_B, V_{NB} and V_{BG} have been collapsed to a single node.

The minimum spanning tree (MST) of the graph C is the spanning tree with the least total edge weight among all possible spanning trees of the adjacency graph [33].

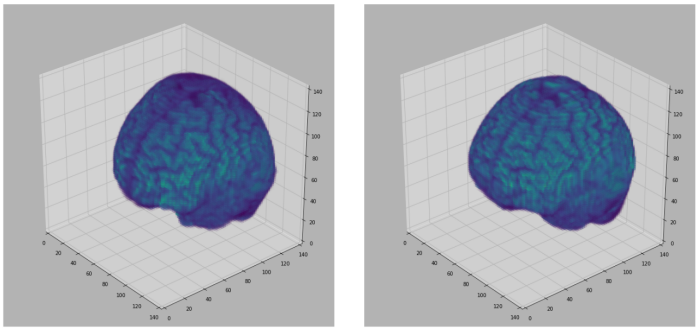
A path P in MST is a sequence of nodes in which each pair of consecutive nodes are connected by an edge. Note that since MST is a tree every pair of nodes is connected by a unique path.

The MST is constructed from the modified graph C and the components representing brain, non-brain tissues and the background are extracted as follows: First, we modify the MST by removing the edge with highest edge weight in the path connecting the nodes v_B and v_{NB} to separate the brain (G_B) and non-brain (G_{NB}) subtrees. Next, we separate the non-brain region and the background by removing the edge with highest edge weight in the path connecting the nodes v_{NB} and v_{BG}. This gives us the connected components of the tree with three different labels. Finally, the labels are reshaped back to the shape of the input MRI image.

3. Results

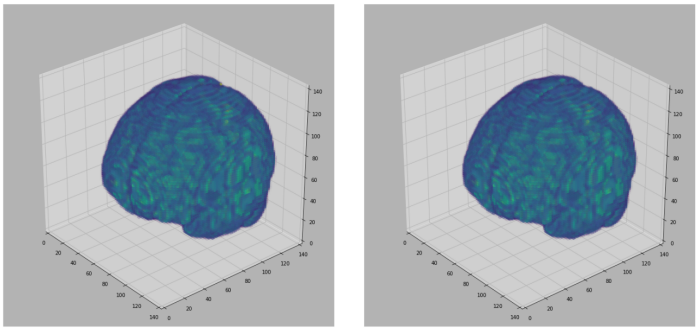
For visualization purposes, we perform a padding of background slices on the MRI image. The visualization of the results in this paper are performed using Matplotlib [34] and Seaborn [35]. The segmented 3D brain from OASIS, BW and IBSR data sets are visualized in Figures 6–8, respectively.

Figures 6–8 represent visual comparison of the extracted 3D brain using the GUBS approach to the gold standard segmented brain. The general quality of the predicted brain is sufficient, except for small variation in finer details. The small variation can be attributed to different reasons such as the quality of the data (see Figure 1), differences when preparing the gold standard segmentation, the patients conditions, such as disease or aging, and possibly parameter tuning for certain data.



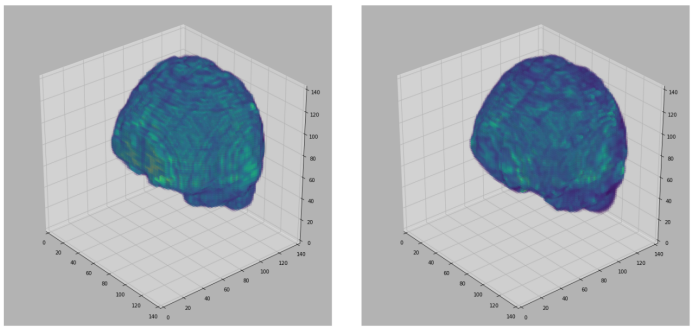
(a) (b)

Figure 6. Segmented Brain (OASIS data): One representative subject representing (**a**) 3D brain segmented using GUBS approach (predicted), (**b**) 3D brain (ground truth). The masks were segmented using a custom method based on registration to an atlas, and then revised by human experts.



(a) (b)

Figure 7. Segmented Brain (BW data): One representative subject representing (**a**) 3D brain segmented using GUBS approach (predicted), (**b**) 3D brain (ground truth). The ground truth was obtained from the labels representing cerebrospinal fluid (CSF), gray matter (GM), and white matter (WM).



(a) (b)

Figure 8. Segmented Brain (IBSR data): One representative subject representing (**a**) 3D brain segmented using GUBS approach (predicted), (**b**) 3D brain (ground truth). The ground truth was obtained by manual-guided expert segmentation.

Figures 9 and 10 present examples of a visual comparison of 2D slices segmented from two different subjects from IBSR and OASIS data sets, respectively. Notice that for Figure 9, the original sagittal sections show that some parts of the 3D head were cut. Figure 10 also present representative sagittal sections from a single subject. Since the proposed approach depends on sampling points within different regions, tuning the parameters for extracting the brain from the two compared MRI volumes (Figures 9 and 10) can differ significantly.

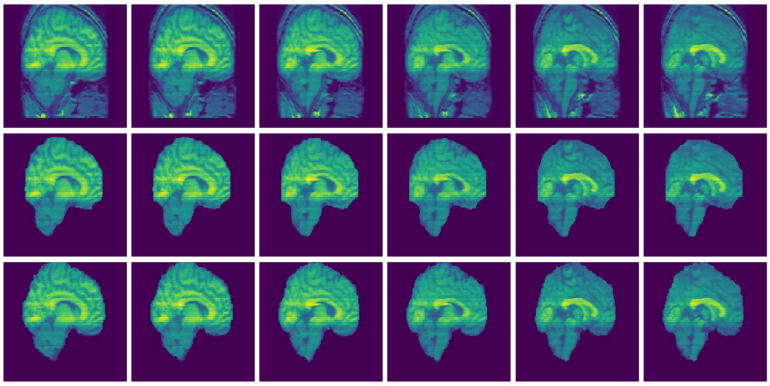


Figure 9. Selected MRI slices (IBSR data set): Sagittal MRI plane segmented brain. Row one: Input images, Row two: Predicted brain and Row three: Ground truth brain.

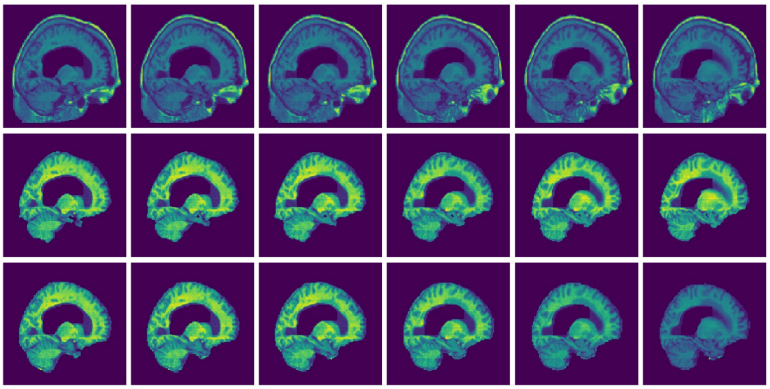


Figure 10. Selected MRI slices (OASIS data set): Sagittal MRI plane segmented brain. Row one: Input images, Row two: Predicted brain and Row three: Ground truth brain.

3.1. Performance Analysis

In this subsection, we present a quantitative analysis using Jaccard Index (JI), Dice Similarity Coefficient (DSC), sensitivity, specificity, accuracy and precision to assess the performance of GUBS. We evaluate the performance by computing the listed measures of similarity between the predicted and the ground truth of the MRI image of the binary labels.

The voxels in the binary labels that are correctly classified as brain are represented as true positive (TP). The voxels that are incorrectly classified as brain are represented as false positive (FP). The voxels that are correctly classified as non-brain tissues are represented by true negative (TN). The voxels that are in the brain region but incorrectly classified

as non-brain tissues are represented by false negative (FN). The Jaccard Index and Dice Similarity Coefficient are computed by

$$JI = \frac{TP}{TP + FP + FN}, \; DSC = \frac{2TP}{2TP + FP + FN}. \tag{8}$$

The sensitivity and specificity which show the percentage of brain and non-brain voxels are computed by

$$Sensitivity = \frac{TP}{TP + FN}, \; Specificity = \frac{TN}{TN + FP}. \tag{9}$$

The precision gives the ratio of the correctly positive identified labels against all the labels, whereas accuracy gives the ratio of the correctly identified.

$$Precion = \frac{TP}{TP + FP}, \; Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{10}$$

The values for the measures of similarity are presented in the range of 0 to 1. A measure of similarity with 0 value shows that there is no overlap between the predicted brain and the ground truth. The measure of similarity with 1 indicates that there is a perfect overlap between ground truth and the predicted brain segmentation.

The quantitative evaluation of GUBS compared to the state of the art method is presented in Tables 1–3 for OASIS, BW and IBSR data sets, respectively. The general performance of GUBS is satisfactory. For some data sets GUBS does not outperform other state of the art methods, but the results obtained by GUBS is consistent with previous results. Looking at the performance from the individual data sets GUBS does not perform well on sensitivity for OASIS and IBSR, but it performs well on specificity from all the three data sets.

Table 1. Performance analysis of GUBS compared to STAPLE and CONSNet [9] methods based on the OASIS data set by presenting the average in each measure of similarity.

	JI (mean ± sd)	**DSC (mean ± sd)**	**Sensitivity (mean ± sd)**	**Specificity (mean ± sd)**
STAPLE	-	**0.960960 ± 0.0070**	0.989830 ± 0.0060	0.951880 ± 0.0200
CONSNet	-	0.955480 ± 0.0100	**0.990550 ± 0.0060**	0.939800 ± 0.0280
GUBS	0.872633 ± 0.0148	0.931918 ± 0.0084	0.937179 ± 0.0256	**0.974223 ± 0.0101**

Table 2. Performance analysis of GUBS compared to BSE, HWA and SMHASS [17] methods based on the BW data by presenting the average in each measure of similarity.

	JI (mean ± sd)	**DSC (mean ± sd)**	**Sensitivity (mean ± sd)**	**Specificity (mean ± sd)**
BSE	0.875000 ± 0.0490	0.932000 ± 0.0310	0.991000 ± 0.0040	0.979000 ± 0.0120
HWA	0.685000 ± 0.0170	0.813000 ± 0.0120	**1.000000 ± 0.0010**	0.928000 ± 0.0050
SMHASS	0.904000 ± 0.0110	0.950000 ± 0.0060	0.990000 ± 0.0030	0.985000 ± 0.0020
GUBS	**0.982396 ± 0.0271**	**0.990927 ± 0.0141**	0.984012 ± 0.0268	**0.999356 ± 0.0005**

Table 3. Performance analysis of GUBS compared to HWA, SMHASS [17] and multi-view U-Net (MVU-Net) [19] methods based on the IBSR data set by presenting the average in each measure of similarity.

	JI (mean ± sd)	**DSC (mean ± sd)**	**Sensitivity (mean ± sd)**	**Specificity (mean ± sd)**
HWA	0.814000 ± 0.0360	0.897000 ± 0.0220	**1.000000 ± 0.0000**	0.966000 ± 0.0120
SMHASS	**0.905000 ± 0.0300**	**0.950000 ± 0.0170**	0.992000 ± 0.0100	0.985000 ± 0.0090
MVU-Net	-	0.908100	0.941400	**0.989400**
GUBS	0.859300 ±0.0176	0.924229 ± 0.0102	0.918936 ± 0.0334	0.980869 ± 0.0104

Consistency Analysis of the GUBS across Different Data Sets

We have segmented three different data sets from OASIS, BW, and IBSR. They are significantly different in terms of quality and quantity. The OASIS data sets were collected from participants of different age groups, healthy and unhelthy subjects suffering from dementia. The data sets from BW are normal simulated brains whereas the IBSR data sets were collected from healthy subjects. To better understand the performance of the GUBS for differences in brain shapes across ages, possibly diseased and non-diseased brains, we combined all the results to explore the relationship between different measures of similarity. Figure 11 presents plots in matrix format showing the relationship between different measures of similarity on the three data sets. The diagonal subplots show the distributions for a specific measure of similarity in the individual data sets.

From Figure 11, the distribution in the diagonal show that the accuracy, DSC, and sensitivity from the OASIS data set have a lower deviation from the mean value compared to BW and IBSR data sets. It also indicates that specificity and precision from BW data have a lower deviation from the mean compared to the specificity and precision from OASIS and IBSR. Furthermore, Figure 11 shows that BW has the highest mean values for all measures of similarity compared to OASIS and IBSR data sets. Measures of similarity from IBSR show a higher deviation from the mean values compared to the measures from other data sets.

The pair plots show that the DSC are positively correlated with the accuracies obtained from all the data sets. The sensitivity obtained from BW and IBSR is positively correlated with accuracy and DSC. Specificity and precision obtained from BW show that they are not correlated with other measures of similarity whereas specificity and precision obtained from IBSR and OASIS are negatively correlated with sensitivity obtained from these data sets. For all data sets the obtained precision is positively correlated with the specificity.

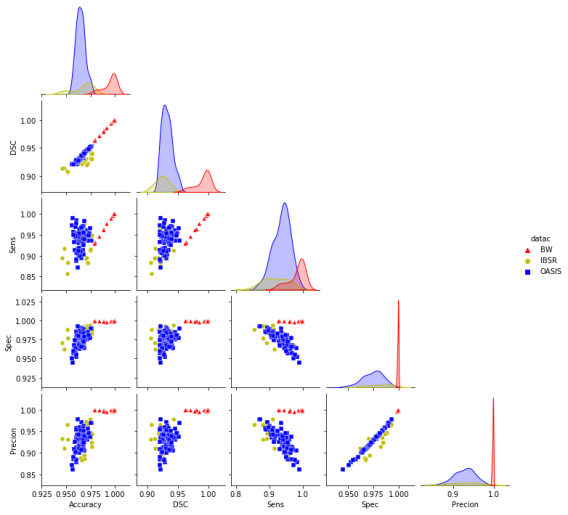


Figure 11. Pair plots for the measures of similarity (Combined data sets): Pair plots showing the pairwise relationship between different measures of similarity for the results obtained using GUBS across the combined data sets. DSC = Dice Similarity coefficients, Sens = Sensitivity, Spec = Specificity.

Figure 12d shows a separation between the measures with and without outliers after combining the results from the three data sets. JI shows many outliers compared to accuracy and DSC. Based on the results presented in Tables 1–3 it is possible that the noted outliers come from BW results. Furthermore, we note that sensitivity, specificity, and precision

do not have outliers even though specificity shows high performance for all the data sets compared to sensitivity and precision.

Looking at the size of the box plots we note that the measures of accuracy, DSCs, and specificity in their distribution show that the lower and upper quartile are close to each other compared to other measures.

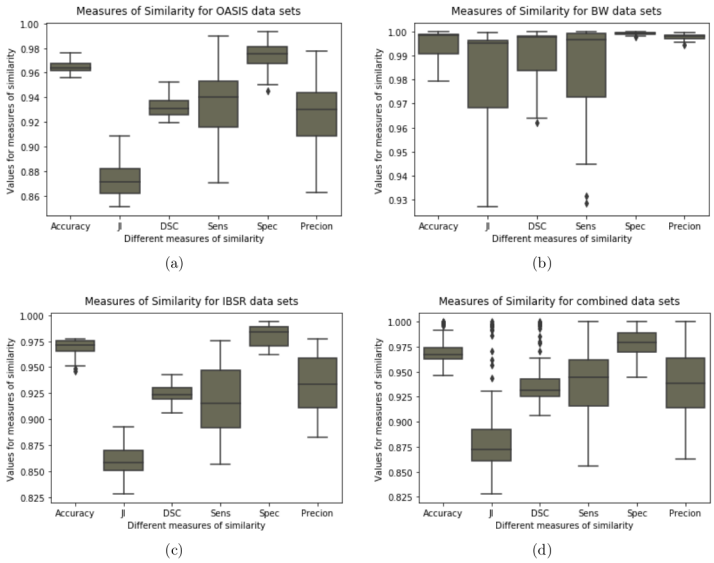


Figure 12. Boxplot for the measures of similarity (Combined Data sets): Boxplot showing variability for the measures of similarity for the results obtained using GUBS method across the combined data sets. JI = Jaccard Indices, DSC = Dice Similarity coefficients, Sens = Sensitivity, Spec = Specificity.

3.2. *Parameter Selection*

Due to high variation within and across image data sets, it is challenging to get a single set of parameters that works to produce the best possible segmentation results for different image data sets. In this work, parameter tuning was initially done on trial and error bases to obtain a value that could achieve good results. For each data set, the GUBS is run by testing different parameter values and choosing the parameters which lead to the separation of the tree into the separated components. Figures 13 and 14 show experimentation for different threshold parameters with same nodes sample size of 20,000, where Figures 15 and 16 show experimentation for different nodes sample sizes and same threshold. Note that BW data (see Figure 16) does not need threshold.

(a) **Image** (b) θ_1=0.0 (c) θ_2=0.1 (d) θ_3=0.2 (e) θ_4=0.27 (f) θ_5 = 0.3

Figure 13. A representative coronal section from 3D MRI showing the separation of components for different thresholds: MRI from a single subject (IBSR data set) experimented using different thresholds and nodes sample size is 20,000. GUBS is run on 3D and for visualization we take 2D at the same location for all experiments.

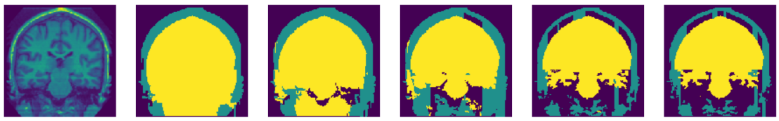


Figure 14. A representative coronal section from 3D MRI showing the separation of components for different thresholds: MRI from a single subject (OASIS data set) experimented using different thresholds and the nodes sample size is 20,000. GUBS is run on 3D and for visualization we take 2D at the same location for all experiments.

Figure 15. A representative coronal section from 3D MRI showing the separation of components for different sample size: MRI from a single subject (IBSR data set) experimented using threshold $T = 0.27$ and different size of the sampled nodes. GUBS is run on 3D and for visualization we take 2D at the same location for all experiments.

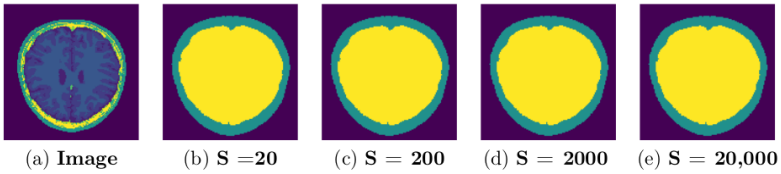


Figure 16. A representative axial section from 3D MRI showing the separation of components for different sample size: MRI from a single subject (BW data set) experimented without threshold, different size of the sampled nodes. GUBS is run on 3D and for visualization we take 2D at the same location for all experiments.

The selection line, δ_1 and δ_2 can highly vary based on the size of the images. The selection of parameters for the selection line depends on the visual inspection because the 3D MRI undergo different preprocessing steps after acquisition of the signals. For OASIS, IBSR and BW data sets the selection line values used were 60, 65 and 35, respectively. For all the three data sets $\delta_1 = 15$. For IBSR and OASIS $\delta_2 = 1$, for BW $\delta_2 = 3$.

Figures 9 and 10 show experimental result of one subject from IBSR and OASIS, respectively. The challenging part is to sample within the non-brain tissues because some parts of the brain and non-brain tissues have been removed (see row one in Figure 9). Some points sampled from the non-brain tissues are likely to be taken from the brain when δ_2 is applied. For Figures 9 and 10, $\delta_1 = 15$ and $\delta_2 = 1$.

3.3. Experimental Timing

The time complexity analysis for implementing GUBS is divided into three parts. These include adjacency graph construction, MST construction, and the time is taken for disconnecting the MST into different connected components. The adjacency graph size is defined by the number of voxels in the MRI image, corresponding to vertices in the adjacency graph. The experimental timing for the adjacency graph and MST construction is efficient and presented in [36]. For the adjacency graph constructed from MRI image $134 \times 134 \times 134$ (with 2,406,104 nodes in the adjacency graph), we update the MST twice to disconnect a path connecting the brain and the non-brain tissues as well as disconnect

the path connecting non-brain tissues and the background. The experimental time ranges between 12 and 25 s for disconnecting a path. The time spent on separating the tree into different connected components depends on the length of the path between the terminal nodes. The implementation was done by writing scripts in the python programming language, and it was run on a PC processor (Core i7-8650UCPU @ 1.90GHz×8).

4. Discussion

We extended the segmentation criteria used in the paper [36] by collapsing subgraphs of the voxels adjacency graph before constructing the MST and presented the GUBS method for segmenting the brain from MRI images. GUBS works by representing MRI volume into an adjacency graph, followed by collapsing representative nodes sampled from the brain, no-brain, and background regions, respectively. Then, the MST is constructed from the modified graph. The collapsed nodes are used as terminal nodes for disconnecting the paths in the MST to separate the brain, non-brain tissues, and background.

The approach was tested by segmenting the brain from three different MRI data sets. The results are compared to the ground truth to assess the performance of GUBS. The experimental results show that GUBS successfully segments brain with high performance from different data sets regardless of the differences of these data sets. Moreover, the results obtained using GUBS are compared to the results obtained using different state of the art methods. GUBS gives competitive results in terms of performance. However, unlike different state of the art methods which require labeled images for training, GUBS does not require labels. In most cases the labeled images are obtained manually by highly qualified individuals. This task is labour intensive and time consuming. That is one of strength of GUBS.

The evaluation of the results obtained using the GUBS approach from the three data sets indicates that the quality of the data highly influences the results. The results obtained from the BrainWeb data set indicate that GUBS outperforms different state of the art methods, whereas results obtained from the OASIS data set provide competitive results compared to the ones obtained previously using neural network approaches. In some cases, the specificity obtained by the GUBS approach outperforms other methods. Similar to other methods, GUBS get good results from IBSR data set. Since the approach depends on sampling points within different regions, the quality of the data might have influenced the IBSR results because some parts of the MRI images were removed. It is likely that nodes sampled in the wrong region will be collapsed together with nodes in a wrong region. Thus, the GUBS approach will be limited in the setting of incomplete 3D brain MRIs. An extension of GUBS to a semi-automated method, in which experts can sample points with high control before running the approach would alleviate this problem.

5. Conclusions

We developed a graph-based method that uses a minimum spanning tree (MST) to segment 3D brain from MRI images. The proposed method was tested by segmenting three different data sets. It is efficient and competes the state-of-the-art methods in terms of performance. The proposed method is simple to adapt and apply on different data sets.

Author Contributions: Conceptualization, S.M. and M.B.; Formal analysis, S.M.; Methodology, S.M. and M.B.; Software, S.M. and M.B.; Supervision, N.B., S.G. and M.B.; Visualization, S.M.; Writing—original draft, S.M.; Writing—review and editing, I.H., J.B.H., S.A., N.B., S.G. and M.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research work is funded by the University of Bergen, Norway.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The first data set is from the Open Access Series of Imaging Studies (OASIS) [22], www.oasis-brains.org (accessed on 15 January 2022). The second data set is from

the BrainWeb (BW) database [23,24], https://brainweb.bic.mni.mcgill.ca/ (accessed on 15 January 2022). The third data set is from the Internet Brain Segmentation Repository (IBSR) [25], https://www.nitrc.org/projects/ibsr (accessed on 15 January 2022). Codes for producing results are available here https://github.com/simeonmayala/GUBS-Brain-segmentation.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Despotović, I.; Goossens, B.; Philips, W. MRI segmentation of the human brain: Challenges, methods, and applications. *Comput. Math. Methods Med.* **2015**, *2015*, 450341. [CrossRef] [PubMed]
2. Kalavathi, P.; Prasath, V. Methods on skull stripping of MRI head scan images—A review. *J. Digit. Imaging* **2016**, *29*, 365–379. [CrossRef] [PubMed]
3. Ramon, C.; Garguilo, P.; Fridgeirsson, E.A.; Haueisen, J. Changes in scalp potentials and spatial smoothing effects of inclusion of dura layer in human head models for EEG simulations. *Front. Neuroeng.* **2014**, *7*, 32. [CrossRef] [PubMed]
4. Fatima, A.; Shahid, A.R.; Raza, B.; Madni, T.M.; Janjua, U.I. State-of-the-art traditional to the machine-and deep-learning-based skull stripping techniques, models, and algorithms. *J. Digit. Imaging* **2020**, *33*, 1443–1464. [CrossRef]
5. Li, J.; Erdt, M.; Janoos, F.; Chang, T.C.; Egger, J. Medical image segmentation in oral-maxillofacial surgery. In *Computer-Aided Oral and Maxillofacial Surgery*; Academic Press: Cambridge, MA, USA, 2021; pp. 1–27.
6. Hwang, H.; Rehman, H.Z.U.; Lee, S. 3D U-Net for skull stripping in brain MRI. *Appl. Sci.* **2019**, *9*, 569. [CrossRef]
7. Kapur, T.; Grimson, W.E.L.; Wells, W.M., III; Kikinis, R. Segmentation of brain tissue from magnetic resonance images. *Med. Image Anal.* **1996**, *1*, 109–127. [CrossRef]
8. Gui, L.; Lisowski, R.; Faundez, T.; Hüppi, P.S.; Lazeyras, F.; Kocher, M. Morphology-driven automatic segmentation of MR images of the neonatal brain. *Med. Image Anal.* **2012**, *16*, 1565–1579. [CrossRef]
9. Lucena, O.; Souza, R.; Rittner, L.; Frayne, R.; Lotufo, R. Convolutional neural networks for skull-stripping in brain MR imaging using silver standard masks. *Artif. Intell. Med.* **2019**, *98*, 48–58. [CrossRef]
10. Rehman, H.Z.U.; Hwang, H.; Lee, S. Conventional and deep learning methods for skull stripping in brain MRI. *Appl. Sci.* **2020**, *10*, 1773. [CrossRef]
11. Lemieux, L.; Hagemann, G.; Krakow, K.; Woermann, F.G. Fast, accurate, and reproducible automatic segmentation of the brain in T1-weighted volume MRI data. *Magn. Reson. Med.* **1999**, *42*, 127–135. [CrossRef]
12. Smith, S.M. BET: Brain Extraction Tool. In *FMRIB TR00SMS2b (Oxford Centre for Functional Magnetic Resonance Imaging of the Brain)*; Department of Clinical Neurology, Oxford University, John Radcliffe Hospital: Headington, UK, 2000; pp. 1–25
13. Dale, A.M.; Fischl, B.; Sereno, M.I. Cortical surface-based analysis: I. Segmentation and surface reconstruction. *Neuroimage* **1999**, *9*, 179–194. [CrossRef] [PubMed]
14. Shattuck, D.W.; Leahy, R.M. BrainSuite: An automated cortical surface identification tool. *Med. Image Anal.* **2002**, *6*, 129–142. [CrossRef]
15. Ségonne, F.; Dale, A.M.; Busa, E.; Glessner, M.; Salat, D.; Hahn, H.K.; Fischl, B. A hybrid approach to the skull stripping problem in MRI. *Neuroimage* **2004**, *22*, 1060–1075. [CrossRef] [PubMed]
16. Iglesias, J.E.; Liu, C.Y.; Thompson, P.M.; Tu, Z. Robust brain extraction across datasets and comparison with publicly available methods. *IEEE Trans. Med. Imaging* **2011**, *30*, 1617–1634. [CrossRef]
17. Galdames, F.J.; Jaillet, F.; Perez, C.A. An accurate skull stripping method based on simplex meshes and histogram analysis for magnetic resonance images. *J. Neurosci. Methods* **2012**, *206*, 103–119. [CrossRef]
18. Roy, S.; Maji, P. A simple skull stripping algorithm for brain MRI. In Proceedings of the 2015 Eighth International Conference on Advances in Pattern Recognition (ICAPR), Kolkata, India, 4–7 January 2015; pp. 1–6.
19. Fatima, A.; Madni, T.M.; Anwar, F.; Janjua, U.I.; Sultana, N. Automated 2D Slice-Based Skull Stripping Multi-View Ensemble Model on NFBS and IBSR Datasets. *J. Digit. Imaging* **2022**, *35*, 374–384. [CrossRef]
20. Rampun, A.; Jarvis, D.; Griffiths, P.D.; Zwiggelaar, R.; Scotney, B.W.; Armitage, P.A. Single-Input Multi-Output U-Net for Automated 2D Foetal Brain Segmentation of MR Images. *J. Imaging* **2021**, *7*, 200. [CrossRef]
21. Sadananthan, S.A.; Zheng, W.; Chee, M.W.; Zagorodnov, V. Skull stripping using graph cuts. *NeuroImage* **2010**, *49*, 225–239. [CrossRef]
22. Marcus, D.S.; Wang, T.H.; Parker, J.; Csernansky, J.G.; Morris, J.C.; Buckner, R.L. Open Access Series of Imaging Studies (OASIS): Cross-sectional MRI data in young, middle aged, nondemented, and demented older adults. *J. Cogn. Neurosci.* **2007**, *19*, 1498–1507. [CrossRef]
23. Aubert-Broche, B.; Evans, A.C.; Collins, L. A new improved version of the realistic digital brain phantom. *NeuroImage* **2006**, *32*, 138–145. [CrossRef]
24. Cocosco, C.A.; Kollokian, V.; Kwan, R.K.S.; Pike, G.B.; Evans, A.C. Brainweb: Online interface to a 3D MRI simulated brain database. *NeuroImage* **1997**, *5*, 425.
25. Rohlfing, T. Image similarity and tissue overlaps as surrogates for image registration accuracy: Widely used but unreliable. *IEEE Trans. Med. Imaging* **2011**, *31*, 153–163. [CrossRef] [PubMed]

26. Kleesiek, J.; Urban, G.; Hubert, A.; Schwarz, D.; Maier-Hein, K.; Bendszus, M.; Biller, A. Deep MRI brain extraction: A 3D convolutional neural network for skull stripping. *NeuroImage* **2016**, *129*, 460–469. [CrossRef] [PubMed]
27. Brett, M.; Markiewicz, C.J.; Hanke, M.; Côté, M.A.; Cipollini, B.; McCarthy, P.; Jarecka, D.; Cheng, C.P.; Halchenko, Y.O.; Cottaar, M.; et al. nipy/nibabel: 3.2.1. 2020. Available online: https://doi.org/10.5281/zenodo.4295521 (accessed on 1 February 2022). [CrossRef]
28. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
29. Harris, C.R.; Millman, K.J.; van der Walt, S.J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N.J.; et al. Array programming with NumPy. *Nature* **2020**, *585*, 357–362. doi: 10.1038/s41586-020-2649-2. [CrossRef]
30. Heng, P.; Wang, L.; Wong, T.; Leung, K.; Cheng, J.C.Y. Edge surface extraction from 3D images. In Proceedings of the Medical Imaging 2001: Image Processing, San Diego, CA, USA, 17–22 February 2001; Volume 4322, pp. 407–416.
31. Bomans, M.; Hohne, K.H.; Tiede, U.; Riemer, M. 3-D segmentation of MR images of the head for 3-D display. *IEEE Trans. Med. Imaging* **1990**, *9*, 177–183. [CrossRef]
32. Virtanen, P.; Gommers, R.; Oliphant, T.E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods* **2020**, *17*, 261–272. [CrossRef]
33. Morris, O.; Lee, M.d.J.; Constantinides, A. Graph theory for image analysis: An approach based on the shortest spanning tree. *IEE Proc. (Commun. Radar Signal Process.)* **1986**, *133*, 146–152. [CrossRef]
34. Hunter, J.D. Matplotlib: A 2D graphics environment. *Comput. Sci. Eng.* **2007**, *9*, 90–95. doi: 10.1109/MCSE.2007.55. [CrossRef]
35. Waskom, M.; Botvinnik, O.; O'Kane, D.; Hobson, P.; Lukauskas, S.; Gemperline, D.C.; Augspurger, T.; Halchenko, Y.; Cole, J.B.; Warmenhoven, J.; et al. mwaskom/seaborn: V0.8.1 (September 2017). Available online: https://doi.org/10.5281/zenodo.883859 (accessed on 1 February 2022). [CrossRef]
36. Mayala, S.; Herdlevær, I.V.; Bull Haugsøen, J.; Anandan, S.; Gavasso, S.; Brun, M. Brain Tumor Segmentation Based on Minimum Spanning Tree. *Front. Signal Process.* **2022**, *2*, 816186. [CrossRef]

Graphic design: Communication Division, UiB / Print: Skipnes Kommunikasjon AS

uib.no

ISBN: 9788230850572 (print)
9788230843239 (PDF)