Segmentation, Feature Extraction &
Autoimmune Clustering for
Foreground-background Image
Retrieval

by

Maedeh Beheshti

School of Information and Communication Technology
Griffith Sciences, Griffith University
Queensland, Australia

Submitted in fulfilment
of the requirements of the degree of
Doctor of Philosophy

September 2016
ABSTRACT

In our digital era, many attempts in remote sensing, fashion, crime prevention, publishing, medicine, architecture and bio-medicine have resulted in a large number of image data sets. Traditional methods to search and retrieve from these data sets are gradually being replaced by state-of-the-art and modern techniques such as content based image retrieval. Retrieving images through extracting contents as a feature and a similarity measure is one of the most challenging applications of computer vision. Due to the increasing number of images with different varieties and types, the traditional content based image retrieval systems are unable to properly exploit the content information of images for retrieval. Thus, extracting relevant features of images and finding a measure of image similarity that returns appropriate relationships is challenging. Content Based Image Retrieval (CBIR) is one of the open problems which still needs much more research effort to completely replace traditional retrieval systems. Feature extraction based on colour, texture, shape and etc. which has been done locally or globally for an image is one of the main parts of CBIR. Image segmentation, which extracts objects from the background and partitioning an image into several regions, helps facilitate feature extraction based on shapes or region of interest (ROI). It is usually performed based on some rules. For example, pixels with similar features and attributes are clustered in one region. Image segmentation has been one of the most important unsolved problems in computer vision for many years. Recently, there have been great efforts in producing better segmentation algorithms. In the first and second part of this thesis, recent approaches of image segmentation to extract image features are studied. A detailed study on two recently proposed graph-based segmentation methods, namely, graph-cut models (deterministic) and unified graphical model (probabilistic) has been conducted. In the experimental result section, the strength and weakness of graph-based methods will be discussed, as well as their runtime performance and accuracy which will be evaluated. The last chapter of this research is developing an appropriate machine learning system on the proposed CBIR framework. A biology inspired clustering system named artificial immune system (AIS) has been proposed in this regard for image retrieval. AIS use the idea from natural immune systems to solve engineering problems and present principles for learning and pattern
recognition. This approach improves learning abilities, scalability, robustness and automatic scale estimation. Finally, the focus of the research in this thesis is on developing a robust and adaptive CBIR system for scenery and computer-aided diagnosis in biomedical imaging which combines computational strategies inspired by biology, together with existent image processing approaches. Choosing two different groups of scenery and medical images provide us an appropriate benchmark of evaluations and experimental results on a two broad areas of images with varieties.
DECLARATION

This work has not previously been submitted for a degree or diploma in any university. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

(Signed) Maedeh Beheshti

23rd September, 2016
ACKNOWLEDGMENTS

I would like to express my deepest appreciation to my principal supervisor, Dr. Jolon Faichney, for his continuous support during my PhD. His patience, motivation, immense knowledge, and guidance helped me throughout the research and writing of this thesis. Without Dr. Jolon’s guidance, advice, support and assistance, I am sure this study would have not been quite as incisive.

I am grateful for the initial and ongoing assistance of my associate supervisor, Dr Xin-Wen Wu, and would also like to thank my external supervisors, Prof. Kambiz Badie and Dr. Maryam Tayefeh Mahmoudi, both of whom gave me patient guidance and advice throughout my study.

Griffith University, but, in particular, the School of Information and Communication Technology have always been a truly wonderful “home” in which to study, work and spend so much time. I express my most sincere thanks and appreciation to all the academics, general, technical and administrative staff and numerous postgraduate students over the years. Their friendship, advice, assistance and of course, critical, technical and administrative support was a central and very important aspect of my PhD study.

I am very thankful to Dr. Maryam Rahnemoonfar, director of the Bina computer vision lab at Texas A&M University-Corpus Christi, of which I had the opportunity to be a Visiting Scholar. I would like to thank everyone at the School of Computer Science, with my very special thanks to Dr. Scott King, for their warm welcome to the Pixel laboratory. Their technical support helped me complete an important component of my thesis. I could share ideas, and the comments I received were very valuable.

Last but not least, I would also like to thank my family, especially my parents, for their support to pursue my passion for learning. They always encouraged me to work hard and never give up.
LIST OF PUBLICATIONS

Journals:


Conferences:


# Table of Contents

ABSTRACT .................................................................................................................. Error! Bookmark not defined.

DECLARATION ............................................................................................................. Error! Bookmark not defined.

ACKNOWLEDGMENTS ............................................................................................... IV

LIST OF PUBLICATIONS ........................................................................................... Error! Bookmark not defined.

TABLE OF CONTENTS .............................................................................................. Error! Bookmark not defined.

LIST OF TABLES ......................................................................................................... VIII

LIST OF FIGURES ........................................................................................................ X

1 INTRODUCTION ....................................................................................................... 1

1.1 Motivation and significance .................................................................................. 2

1.2 Research scope ..................................................................................................... 2

1.3 Summary of research methodology and contribution ......................................... 3

1.3.1 Research methodology .................................................................................... 3

1.3.2 Research methodology diagram ...................................................................... 4

1.4 Thesis outline ....................................................................................................... 5

2 BACKGROUND ....................................................................................................... 7

2.1 Introduction .......................................................................................................... 7

2.2 Graph-cut methods for image segmentation ....................................................... 7

2.2.1 Maximum-flow/minimum-cut algorithms ....................................................... 7

2.2.2 Energy minimization in computer vision ....................................................... 12

2.2.3 Energy minimization for labelling problems ............................................... 12

2.2.4 Binary and multi-label graph-cut .................................................................. 14

2.2.5 Optimal multi-region objects segmentation .............................................. 16

2.3 Graphical models and image segmentation ....................................................... 18

2.3.1 Conditional random field as an example ...................................................... 21

2.4 Artificial neural network ..................................................................................... 27

2.5 Spatial fuzzy clustering for image segmentation ................................................. 29

2.6 Fuzzy level-set for image segmentation ............................................................. 30

2.7 Summary ............................................................................................................. 30

3 GRAPH-CUT SEGMENTATION BASED ON BAYES MODEL .......................... 32

3.1 Introduction .......................................................................................................... 32

3.2 Bio-cell image segmentation using Bayes graph-cut model ............................. 33

3.3 Experimental results and analysis ...................................................................... 38

3.4 Conclusion .......................................................................................................... 42

4 ENSEMBLE-BASED IMAGE SEGMENTATION FOR FLUORESCENCE MICROSCOPY IMAGES .......................................................... 43

4.1 Introduction .......................................................................................................... 43

4.2 Proposed ensemble-based framework .................................................................. 46
4.2.1 Experimental results ........................................................................................................48
4.3 Discussion and conclusion ....................................................................................................66
5 GRAPH-BASED IMAGE RETRIEVAL USING FUZZY LINKING HISTOGRAM AND
ARTIFICIAL IMMUNE SYSTEM .....................................................................................68
5.1 Introduction ..........................................................................................................................68
5.2 Proposed components for CBIR .........................................................................................69
  5.2.1 Conditional random field for region-based image segmentation ................................69
  5.2.2 Immune-based clustering .............................................................................................71
5.3 Feature extraction ................................................................................................................75
  5.3.1 Fuzzy linking histogram ...............................................................................................75
5.4 Total framework of the proposed CBIR ..............................................................................77
5.5 Experimental results and analysis ......................................................................................79
  5.5.1 Image segmentation based on graph-cut algorithms and probabilistic graphical
       models .......................................................................................................................79
  5.5.2 Content-based image retrieval .......................................................................................82
5.6 Conclusion ............................................................................................................................97
6 CONCLUSION .......................................................................................................................98
  6.1 Contributions ......................................................................................................................98
  6.2 Further research ..................................................................................................................99
BIBLIOGRAPHY .....................................................................................................................102
APPENDIX A ..........................................................................................................................110
List of Tables

Table 2.1. Label cost assignment to a pixel.......................................................... 13
Table 2.2. Node potentials....................................................................................... 22
Table 2.3. Edge potentials ...................................................................................... 22
Table 2.4. Probability of each configuration for four students............................. 23
Table 3.1. Segmentation accuracy results of BBBC008V1 for different methods ...... 41
Table 4.1. Naïve statistics computed from error matrix ........................................ 50
Table 4.2. Kappa statistics computed from error matrix ......................................... 51
Table 4.3. Average of naïve and kappa accuracy and error per each algorithm and ensemble – dataset 1-W(1) ........................................................................... 52
Table 4.4. Average of naïve and kappa accuracy and error per each algorithm and ensemble – dataset 1-W(2) ........................................................................... 52
Table 4.5. Average of naive user and naive producer accuracy per each algorithm and ensemble – dataset 1-W(1) ........................................................................... 52
Table 4.6. Average of naive user and naive producer accuracy per each algorithm and ensemble - dataset 1-W(2) ........................................................................... 53
Table 4.7. Average of naïve and kappa accuracy and error per each algorithm and ensemble – dataset 2 - C (1) ........................................................................... 59
Table 4.8. Average of naïve and kappa accuracy and error per each algorithm and ensemble – dataset 2 - C (3) ........................................................................... 59
Table 4.9. Average of naive user and naive producer accuracy per each algorithm and ensemble - dataset 2- C (1) ........................................................................... 60
Table 4.10. Average of naive user and naive producer accuracy per each algorithm and ensemble - dataset 2- C (3) ................................................................. 60
Table 5.1. Segmentation time for two graph-based models ........................................ 82
Table 5.2. Average retrieval precision for top 10 retrievals in 10th iteration ............... 95
Table 5.3. Compartmental results of evaluation of image segmentation ................. 96
List of Figures

Figure 1.1. Research methodology ................................................................. 4

Figure 2.1. (a) Edge in the original graph and (b) Edge in the residual network .......... 8
Figure 2.2. (a) Initial network and (b) Residual network ................................................. 9
Figure 2.3. (a) Iteration 1 (S 2 T) and (b) Iteration 2 (S 2 4 T) ....................................... 9
Figure 2.4. (a) Iteration 3 (S 3 4 T) and (b) Iteration 4 (S 1 T) ..................................... 10
Figure 2.5. Final flow graph ...................................................................................... 10
Figure 2.6. (a) The abdomen image (b) The labeled image of abdomen ......................... 12
Figure 2.7. A graph with two terminals ....................................................................... 14
Figure 2.8. (a) Shows current labelling containing three $\alpha, \beta$ and $\gamma$ parts – (b) and (c) Show an $\alpha, \beta$ and $\gamma, \beta$ swap respectively ........................................................................................................... 16
Figure 2.9. Inter-surface connection and intra-column links in a 3D shape ................. 17
Figure 2.10. An example of two mutually exclusive regions ...................................... 17
Figure 2.11. A structure for unified graphical model combining CRF model and BN model by region nodes .................................................................................................................. 20
Figure 2.12. Order of seating for four students ......................................................... 21
Figure 2.13. A sample of cheating for 200 questions for four students seated in different rooms .................................................................................................................................. 24
Figure 2.14. A sample of cheating for 200 questions for four students seated in the same room .................................................................................................................................. 25
Figure 2.15. An over-segmented image with related super-pixels which show the nodes in CRF and the links between them are the connections between the nodes ............ 26
Figure 2.16. Feed-forward NNs with two hidden layers ............................................. 27
Figure 3.1. Graph-cut model ...................................................................................... 34
Figure 3.2. Gaussian Bayes model ............................................................................. 35
Figure 3.3. (a) An image of human disorder cells (b) Its ground truth ....................... 38
Figure 3.4. (a-b) The segmentation result of multi-label graph-cut method .............. 38
Figure 3.5. (a) Original image (b) Ground truth (c) Segmentation result of the proposed method ............................................................ 39
Figure 3.6. (a-c) First row shows the segmentation result of the binary (object/background) graph-cut method- Second row shows the segmentation result of the binary graph-cut method based on Bayes theorem................................. 40
Figure 4.1. Overview of our proposed framework based on robust cell segmentation.. 46
Figure 4.2. Original images of the two different data sets. (a-b) Synthetic microscopy images from SIMCEP (w1 and w2) (c-d) BBBC008 (c1 and c2).............................. 49
Figure 4.3. Segmentation results of different methods on a randomly selected image from dataset 1. From left to right: original image, ground truth, Bayes graph-cut, binary graph-cut, fuzzy LSM, SFCM and Ensemble method ........................................ 52
Figure 4.4. Different boundary recognition in two models of segmentation (a) Ground Truth (b) Weighted ensemble (c) Segmentation Binary graph-cut segmentation........ 53
Figure 4.5. Box plots of (a) Overall Naïve accuracy (w1), (b) Overall Naïve accuracy (w2), (c) Overall Kappa accuracy (w1), (d) Overall Kappa accuracy (w2) dataset1 .... 56
Figure 4.6. Box plots of (a) Overall Naïve error (w1), (b) Overall Naïve error (w2), (c) Overall Kappa error (w1), (d) Overall Kappa error (w2) for dataset1 .................. 57
Figure 4.7. Average Naïve and Kappa user accuracies for foreground and background for four segmentation methods and ensemble (dataset1) ........................................... 58
Figure 4.8. Average Naïve and Kappa producer accuracies for foreground and background for four segmentation methods and ensemble (dataset1) ...................... 58
Figure 4.9. Segmentation results of different methods on a randomly picked image from dataset 2 ......................................................................................... 59
Figure 4.10. Box plots of (a) Overall Naïve accuracy (c1), (b) Overall Naïve accuracy (c3), (c) Overall Kappa accuracy (c1), (d) Overall Kappa accuracy (c3) for dataset2... 63

Figure 4.11. Box plots of (a) Overall Naïve error (c1), (b) Overall Naïve error (c3), (c) Overall Kappa error (c1), (d) Overall Kappa error (c3) for dataset2

Error! Bookmark not defined.

Figure 4.12. Average Naïve and Kappa user accuracies for foreground and background for four segmentation methods and ensemble (dataset2) Error! Bookmark not defined.

Figure 4.13. Average Naïve and Kappa producer accuracies for foreground and background for four segmentation methods and ensemble (dataset2) Error! Bookmark not defined.

Figure 4.14. Runtime in seconds for Bayes graph-cut, binary graph-cut, fuzzy LSM and SFCM segmentation methods and weighted ensemble... Error! Bookmark not defined.

Figure 5.1. An over-segmented image with related superpixels which show the nodes in CRF and the links between them are the connections between the nodes .......................... 70

Figure 5.2. A binding between an external antigen (new image) and an image of the cluster with high affinity ................................................................. 74

Figure 5.3. (a) Membership function of the input variable L* (b) Membership function of the input variable a* (c) Membership function of the input variable b* (d) Membership function of the output variable of fuzzy system ............................... 76

Figure 5.4. Fuzzy linking histogram and an image output .................................... 78

Figure 5.5. A total framework of CBIR system ..................................................... 79

Figure 5.6. Comparison between unified graphical model and graph-cut model for the horse dataset ............................................................ 81

Figure 5.7. Comparison between unified graphical model and graph-cut model for the grey-scale horse dataset ........................................................................ 81
Figure 5.8. Output of N-cut algorithm on Weizmann dataset ........................................ 82
Figure 5.9. An RGB image from Weizmann dataset and Ground truth ....................... 83
Figure 5.10. Results of our CBIR system through CRF image segmentation model with feature vector containing fuzzy linking histogram (top row) and features without fuzzy linking histogram (bottom row).......................................................................................................................... 84
Figure 5.11. The mean average precision of the three different methods for ten iterations on Microsoft and Weizmann datasets ................................................................................................................. 94
Figure 5.12. The recall retrieval performance of combining different texture, colour and fuzzy linking histogram features on Microsoft and Weizmann datasets ................. 95
Figure 6.1. An ontology-based system ............................................................................... 101
INTRODUCTION

Content Based Image Retrieval is a research area which attracts great attention of many researchers in each part of the problem such as feature extraction, similarity assessment, and image analysis [1, 2]. In order to have a robust CBIR system, image feature extraction and selection is a very important task. In addition, machine learning algorithms like clustering and classification need to be developed for CBIR systems. Image analysis and feature extraction is also one of the challenging issues that needs to be solved. The first part of this study mainly focuses on the feature extraction part of a CBIR system. As real world problems suffer from complexity and uncertainty, they are often studied using a probabilistic framework. Probabilistic graphical models are powerful techniques in machine learning as they could take into account the uncertainty presence in knowledge and information.

Partitioning an image into some meaningful parts or regions (sets of pixels) that share certain visual characteristics based on measurements taken from an image like grey level (level of brightness), colour, texture, depth or motion, is the main purpose of image segmentation. Image segmentation plays a crucial role in many applications such as object identification for object-based measurements (size and shape), content-based image retrieval, and medical imaging. In spite of the many existing algorithms and techniques for image segmentation, our research focuses only on graph partitioning methods. Image segmentation algorithms in this category can be broadly divided into two types: 1- deterministic 2- probabilistic [3]. Algorithms such as max/min cut and normalized cut belong to the deterministic approach, and algorithms such as Markov random field [4] and Bayesian network belong to the probabilistic graphical approach. Energy minimization [5] and probabilistic inference are the main tasks, respectively of the graph cut algorithms [6-9] and probabilistic models. Because of dealing with many variables, the energy function of graph-cut algorithms [10] tends to be very complex and often NP-hard, especially for non-convex problem [11-13].
1.1 Motivation and significance

Appropriate feature extraction could greatly facilitate further image analysis. Finding powerful feature extraction techniques for scenery and medical image analysis, which is appropriate for the type of images, considered as an important task in CBIR. In other words, different kinds of images related to different categories contain specific features that separate them from other categories. For example, medical images are varied in format. In medical studies, MRI is suitable for imaging soft tissue, but does not reveal the bone structure, and this can only be achieved by using Computerised Tomography (CT) which is better for imaging bone. A relevant similarity measure through applying machine learning algorithms in order to retrieve similar images to query image is another important task in CBIR. A biology inspired clustering system using the idea from natural immune systems is a solution to solve similarity measure problem. A biology inspired clustering system indicates scalability and effectiveness for learning, robustness to noise and pattern recognition.

1.2 Research scope

In this study, we will focus our research on intelligent image feature extraction, by using state-of-the-art segmentation methods. In the proposed CBIR we will use both Global and Local methods [14, 15] for feature extraction. We propose colour histograms, Gabor filters and Fuzzy linking histogram [15, 16] for global feature extraction. By using knowledge and experiences from object recognition and energy minimization algorithms we aim to improve those algorithms and develop new CBIR to be able to generalize to various scientific image analysis and retrieval problems. Some segmentation techniques and other image analysis techniques have shown themselves useful in a wide array of domains such as classification of tissues in computed tomography or content-based image retrieval in medical databases.

We will investigate how these new intelligent feature extraction methods and image representations can be used in our research. The specific research questions defined for this research are as below:

1. What is the most effective image retrieval method that can be proposed to develop a content-based image retrieval system?
• How can segmentation and feature extraction methods (globally or locally) be employed in CBIR to increase the image retrieval rate and decrease the cost of incorrect retrieval?

2. How can cell imaging be optimized with graph-based concepts to provide better performance and fewer false results than state-of-the-art techniques in efficient segmentation, and flexible retrieval of medical images?

1.3 Summary of research methodology and contribution

1.3.1 Research methodology

The overall framework used in the thesis is discussed in this section. This section contains answers to the research questions mentioned in section 1.2. It provides a description of the methods employed, and the procedures used in the investigation.

Image segmentation and feature extraction phase: The first part of this study was dedicated for investigating the best methods of feature extraction. As illustrated in previous sections, in order to tackle the segmentation problem and extract objects from the background, two groups of graph-based algorithms, deterministic and probabilistic, have been explored. In this regard, other methods of low-level feature extraction for colour, texture and shape will also be considered as local or global image features. In Chapter 3 we propose a graph-cut image segmentation based on Bayes model and evaluate the proposed method on bio-cell images. Chapter 4 addresses the segmentation problem by proposing an ensemble-based image segmentation model. In order to evaluate the efficacy of the proposed method a huge number of statistical measure performed on bio-cell images. The first part of chapter 5 also shows the results of applying probabilistic approach of image segmentation on scenery images.

Image retrieval phase: The purpose of this phase is to construct a model of content-based image retrieval. The main objective is to come up with learning methods that could automatically extract related features from different types of images and make a decision about related image categories which have to be retrieved. The second part of this research is specific to the investigation of some machine learning and clustering techniques. Although there are many proposed methods of clustering in this area, we have chosen bio-inspired clustering techniques which have been proposed based on the human immune system. The next step of this phase is identifying limitations in the
methods. In this regard, the requirements and limitations of graph-based segmentation methods are considered to identify the research gap. Chapter 5 represents an implemented CBIR system based on fuzzy linking histogram features extracted from foreground of images and artificial immune system as a similarity measure.

**Test and evaluation:** The image retrieval phase will produce a framework that is suitable for content-based image retrieval. In the test and evaluation phase, the proposed CBIR is tested and evaluated using different kinds of image data sets through different features extracted from foreground. The results are analysed and compared with two different segmentation methods.

### 1.3.2 Research methodology diagram

Figure 1.1 describes our research methodology in a diagram as below.

![Research methodology diagram](image)

*Figure 1.1. Research methodology*
All algorithms in our experiments in all phases of research methodology are implemented in Matlab software using an Intel core i5 3320M, 2.6 GHz CPU with 8GB RAM.

1.4 Thesis outline

This thesis is organised as follows:

- **Chapter 1** describes research methodology in response to the research questions provided in Chapter 1. Hypotheses are also discussed in this Chapter in response to the research questions.

- **Chapter 2** provides a background to, and a literature review of, the research topic.

- **Chapter 3** In order to improve the conventional deterministic segmentation models, this Chapter proposes a novel graph-cut cell image segmentation algorithm based on Bayes theorem. There are two segmentation phases in this method. The first phase is an interactive process to specify a preliminary set of regional pixels and the background based on the interactive graph-cut model. In the second phase, final segmentation is calculated based on the idea of Bayes theorem, combining prior information with data. Our idea can be considered an integration of graph-cut methods and Bayes theorem for cell image segmentation. Experimental results show that the proposed model performs better in comparison with several existing methods.

- **Chapter 4** In this Chapter an ensemble based method is proposed for the segmentation of cell cancer microscopy images. The ensemble is constructed using Bayes graph-cut algorithm, binary graph-cut algorithm, spatial fuzzy C-means, and fuzzy Level-Set algorithm, which are chosen for their accuracy and efficiency in the segmentation area. We investigate the performance of each method separately and finally compare the results with the ensemble method. Experiments are conducted over two datasets with different cell
types. At 95% confidence level, the ensemble based method represents the best among all the implemented algorithms.

- **Chapter 5** describes an integrated framework consists of a graph-based segmentation algorithm, a fuzzy linking histogram feature extraction and immune-based clustering system method. To improve the retrieval performance and reduce the semantic gap, our system incorporates a combination of different colour and texture features with fuzzy linking histogram that uses a graph-theoretic approach to improve retrieval of images in accordance with the bio-inspired clustering algorithm. The performance of the proposed framework is evaluated using a large number of images from two datasets with different scenes and complexity. A quantitative comparison is also performed with several existing approaches.

- **Chapter 6** describes the major contribution of this thesis, and explains future directions of this research.
2 BACKGROUND

2.1 Introduction

Image segmentation is the task of labelling each pixel in an image in order to locate objects and their boundaries. Image segmentation is applicable in many areas: content-based image retrieval, machine vision, medical imaging, object detection, and object recognition. Many algorithms have been proposed for image segmentation and they can be classified into different categories. The two main categories for image segmentation are probabilistic approaches and deterministic ones. Probabilistic methods contain two main categories: 1- Graphical Model (GM) based segmentation; 2- Bayesian Segmentation without using GM [3]. Graphical Model (GM) based segmentation tries to solve image segmentation problems with graphical models such as Markov Random Field (MRF), Conditional Random Field (CRF) and Bayesian Network (BN). Bayesian Segmentation attempts to solve image segmentation problems with Bayesian statistics like discriminative methods and generative methods. Some deterministic segmentation methods are k-means-clustering, watershed segmentation and graph-partitioning methods [3]. In this thesis, we focus on the probabilistic graphical models and graph-cut methods.

2.2 Graph-cut methods for image segmentation

2.2.1 Maximum-flow/minimum-cut algorithms

Maximum flow algorithms offer foundations to many global optimization methods. There are many methods for implementing the maximum flow algorithms, such as Goldberg-Tarjan’s [17] “push-relabel” methods and algorithms based on Ford-Fulkerson [18]. In this section, we will describe the Ford-Fulkerson algorithm due to its effectiveness and popularity in many graph-cut methods.
Figures 2.2 to 2.5 show the maximum flow/minimum cut procedure for finding the maximum flow from S (source) node to T (terminal) node [17]. The weights shown in the initial weighted network, Figure 2.2 (a), indicate the capacity of the edges, and the current maximum flow from one node to the other node. Figure 2.2 (b) represents the equal residual network for the capacity network. For example in Figure 2.2 (a) the maximum flow from node (2) to node (4) is 4. There are many paths from S to T in Figure 2.2 (a) and each path has a bottleneck edge. The edge in a path which has a minimum capacity is a bottleneck edge. The purpose of max/min cut is to determine the maximum amount of flow which can be sent from one source node S to the terminal node T. For example, in Figure 2.2 (a) although the capacity of the (S1) edge is 7, the maximum flow of 5 can be sent from S to T in a (S1T) path. Therefore, the (1T) edge is a bottleneck here. Each iteration tries to saturate at least one path in the graph. This means that after each saturation, all the capacity of the edges constructed for that path has been used. Therefore the maximum amount of flow is expected to be passed through a saturated path. In each iteration, we try to find a path from S to T. If there is an edge in the original graph with capacity 3 as depicted in Figure 2.1 (a), its equal edge in the residual network is according to Figure 2.1 (b). For each edge there is a flow/capacity number such as 3/0 in residual graph. Algorithm 2.1 shows how capacity in the residual network is calculated.

![Figure 2.1](image)

Figure 2.1. (a) Edge in the original graph and (b) Edge in the residual network

Beginning from iteration 1, which has been shown in Figure 2.2, all the procedure will be finished in iteration 4. In each iteration, a path will be saturated. Path S2T has been selected for the first time in Figure 2.2. In this path the capacity between 2 and T nodes, which is 3 units, shows a bottleneck.
Figure 2.2. (a) Initial network and (b) Residual network

Figure 2.3(a) shows how the capacity for residual network is calculated according to Algorithm 2.1. For the forward arrows in the S2T path in the residual network, (0+3) and (8-3) show 3/5 calculation for the S2 forward arrow. Also (0+3) and (3-3) show 3/0 calculation for 2T forward arrows.

Figure 2.3. (a) Iteration 1 (S 2 T) and (b) Iteration 2 (S 2 4 T)

The flow current calculation is the same for the backward arrows. The capacity for the backward path is calculated as \( \text{Capacity After Flow} = \text{Capacity Before Flow} - \text{Flow} \). Figure 2.3 (a) represents the path found between S and T as S2T with max flow, 3. In the next iteration for finding the saturated path S24T, Figure 2.3 (b) shows a path with a bottle neck 4 and a maximum achieved flow 4. Figure 2.4 (a and b) show S34T
and S1T saturations which resulted in 2 and 5 maximum flows. Finally the maximum flow is calculated as 3+4+2+5=14. Regarding Figure 2.5, there are two ways from S to reach one of the 1 or 2 nodes, but there is no way for them to reach T node. All the nodes to T node have been blocked and their capacity has been used completely. At this point, we say that all the paths from S to T have been saturated. In this regard, S and T will be separated by removing some saturated edges, 1T, 2T, 24 and S3 from the main graph. The separated nodes are two groups, \{S, 1, 2\} and \{3, 4, T\}.

Figure 2.4. (a) Iteration 3 (S 3 4 T) and (b) Iteration 4 (S1 T)

Figure 2.5. Final flow graph
Algorithm 2.1. Ford-Fulkerson

Ford-Fulkerson Algorithm $G(V, E, s, t, f_w, c_a)$

$f_w(u, v) = 0$ for each pair $(u, v)$ in $V^*V$ \hspace{1cm} // Initialization term

Construct residual network $G_R$

Residual network $G_R(V, E_R) = G(V, E)$

$\forall (u, v) \in E, (u, v) \in E_R$ and $(v, u) \in E_R$ \hspace{1cm} // Initialization term

$f_{WR}(u, v) = 0$ and $f_{WR}(v, u) = 0$ \hspace{1cm} // Initialization term

$c_{AR}(u, v) = c_a(u, v) - f_{WR}(u, v)$ \hspace{1cm} // Initialization term

$c_{AR}(v, u) = c_a(v, u) - f_{WR}(v, u)$ \hspace{1cm} // Initialization term

While there is an $s$-$t$ path $p$ in the residual network $G_R$

& $c_{AR}(u, v) > 0$ for all edges $(u, v)$

    do \hspace{1cm} // increase flow by augmenting flow along $p$

        $c_{AR}(p)$ $\leftarrow$ $\min c_{AR}(u, v)$

        $f_{WR}(u, v)$ $\leftarrow$ $f_{WR}(u, v) + c_{AR}(p)$

        $c_{AR}(u, v)$ $\leftarrow$ $c_{AR}(u, v) - f_{WR}(u, v)$

        $f_{WR}(v, u)$ $\leftarrow$ $f_{WR}(v, u) - c_{AR}(p)$

        $c_{AR}(v, u)$ $\leftarrow$ $c_{AR}(v, u) - f_{WR}(v, u)$

        $f_w$ $\leftarrow$ $f_w + c_{AR}(p)$

    return $f$

End

In algorithm 2.1, $f_w$ and $c_a$ variables show flow and capacity of a path respectively. Variables that appear with $R$ are specified for residual graph. Also $V, E$ represent nodes and edges in the graph respectively.

Locality and parallelization are two important factors in flow propagation in max/min algorithms. Locality refers to fitting the massive and huge graph vertices into computer memory, and parallelization refers to carrying out the steps in maximum flow algorithms asynchronously [17, 19]. Absence of each of them will result in an impractical maximum flow algorithm. In addition, the running time of any maximum flow is an important issue which has to be considered. On one hand, finding the shortest path for sending flow is the goal [17, 19]; on the other hand, looking for the shortest path is expensive and time consuming.
2.2.2 Energy minimization in computer vision

An energy function in computer vision is actually a mathematical function that needs to be maximised or minimized [20]. Prior to describing the energy minimization for graph–cut algorithms, energy minimization in labelling problems is described in order to introduce the concept behind the energy minimization in graph-cut models.

2.2.3 Energy minimization for labelling problems

The main purpose in labelling problems is to assign a label from a set of labels to data in an observation data set. This is performed based on a criterion that specifies the cost of assigning a label to data. In an image, all the pixels in the image constitute the set of data (observation) which needs to be labelled. This observation in an image is not limited only to pixels of an image. It can be salient points in an image, depth measurements from a range scanner, or intensity measurements from CT/MRI. In Figure 2.6 (a,b) the main image and the labeling image have been shown. The purpose of labeling in this figure is to specify an object from the background. Then two labels are enough to make this classification. In Figure 2.6 (b) it has been shown how the main object received a different label (white) from the background (black).

![image](image)

(a) (b)

Figure 2.6. (a) The abdomen image (b) The labeled image of abdomen

Although the accuracy of segmentation in this Figure is not perfect, it can be understood how object has been separated from background by two labels.
Suppose \( P, p \) depict the observations and a pixel node, respectively, and \( L \) indicates the label set. \( h: P \rightarrow L \) is a discrete labeling which assigns to each pixel \( p \in P \) a label from \( L \). For example if we have \( P = \{ p, q \} \) and \( L = \{ l_1, l_2, l_3 \} \) then labeling \( h = \{ l_2, l_3 \} \) means \( h_p = l_2 \) and \( h_q = l_3 \). Overall, \( |L|^{|P|} \) possible configurations are available for a labelling problem and which one of them is preferable to the others is based on the problem. For example, if \( P \) indexes 200 * 200 pixels of an image, and there exists two labels \( \{ \text{objects, background} \} \) in the label set, then the possible labelling for this case is \( 2^{40000} \). An energy function based on the labelling terms \( h \) is defined according to Eq. 2.1.

\[
\cdots + h(2 \text{mret}) + h(1 \text{mret}) = h \cdot E
\]  

(2.1)

In this way, the energy function has been broken down into terms that make it possible to add extra terms to the equation. If each term is related to the cost of assigning a label to a data or pixel, minimizing energy function results in a compromise among all the labelling criteria in the energy function. An example of how this works is as follows. Assume a set of three pixels \( P = \{ p, q, r \} \) and a set of two labels \( \{ lb1, lb2 \} \) and \( D_p(l) \) as an indicator of cost of assigning a label \( lb \) to one pixel \( P \). In this regard, we can define an energy function according to Eq. 2.2.

\[
E(f) = D_p(h_p) + D_q(h_q) + D_r(h_r)
\]  

(2.2)

Table 2.1 shows the cost of assigning each label to a pixel. The minimum energy cost 7 will be achieved by \( h = \{ lb1, lb1, lb2 \} \) labelling.

<table>
<thead>
<tr>
<th>L ( \setminus ) P</th>
<th>P</th>
<th>Q</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>lb1</td>
<td>1</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>lb2</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

The time needed for minimizing the above binary energy function is \( \psi(|P||L|) \) time due to the ability of minimizing each term independently.
2.2.4 Binary and multi-label graph-cut

Binary graph-cut is a method proposed by Boykov and Jolly [13] for segmentation when they published their ideas about finding the s-t cut ($S^*$) of minimal total cost with 2 labels (binary labels) in a graph. In order to partition an image into two segments, object and background, many methods such as Snakes [21], active contours [21], geodesic active contours [22], shortest path techniques [23, 24] and other algorithms like region growing, and split and merge have been proposed. In binary graph-cut, an image could only be segmented into two parts, object and background, with soft constraints on boundary and region. In the binary graph-cut model, an undirected graph $G (V,E)$ (see Figure 2.7) is formed in which $V$ and $E$ represent graph nodes (equal to image pixels $P$) and graph edges, respectively. In addition, two extra terminal nodes, $S$ (source) and $T$ (sink), which represent the object and background labels, are also defined. All the edges are divided into two classes: n-links and t-links.

![Figure 2.7. A graph with two terminals – Arrows show the flow from source (S) to sink (T) – dashed curve represents the min-cut point [11, 13]. – Thickness of the links represent the amount of flows that has been sent from the source to the destination](image)

Each node has two terminal links with terminal nodes, one for $S$ and one for $T$. These links are known as T-links. N-links or neighbourhood links are the links between pixels and they construct an arbitrary neighbourhood system. Each edge has a dedicated cost or weight and an s-t cut ($S^*$) is a subset of edges with minimum cost partitioning the graph (image) into two parts, source and terminal. In this regard, the purpose of segmentation is to minimize the energy function in Eq. 2.3 which is the sum of regional (cost of t-links) and boundary terms (cost of n-links).
\[ E(A) = \delta \cdot \text{Region}(A) + \text{Boundary}(A) \]  

(2.3)

A defines a segmentation and \( \delta \geq 0 \) is a coefficient which emphasizes the regional term. This energy minimization through s-t cut is binary because there are only two terminals. The segmentation is performed based on max-flow algorithm [17] which has been discussed in section 2.2.1.

To handle more than two labels, i.e. \(|L| \geq 3\), multi-label energy minimization through multi-terminal min-cut or move-making (local search) algorithms like alpha-beta swap algorithm and alpha expansion were proposed [12]. Alpha-beta swap algorithm and alpha expansion are two kinds of very large search neighbourhood techniques (VLSN). Through VLSN algorithms and appropriate move space (S), an exploration of an exponential number of alternative labeling will be facilitated in polynomial time. Considering \( \alpha \beta \)-swap here, given current labeling \( f \) from a pair of labels \( \{\alpha, \beta\} \in L \), the purpose is finding a labeling \( f^* \in \{\alpha, \beta\} \) that minimizes the energy function in Eq. 2.4 [12].

\[
\text{Energy}(f) = \text{Energy}_{\text{smooth}}(f)[\sum_{(p,q) \in N} V_{p,q}(f_p, f_q)] + \\
\text{Energy}_{\text{data}}(f)[\sum_{p \in P} D_p(f_p)]
\]

In Eq. 2.4, \( N \) is the set of interacting pairs of pixels such as \( p \) and \( q \). \( D_p(f_p) \) is the cost of assigning a label, such as \( L \) to a pixel \( p \), and \( V_{p,q}(f_p, f_q) \) is a smooth term between two pixels \( p \) and \( q \) that shows coherence and consistency between labels [11, 12]. The smooth term smooths out inconsistency and fluctuations caused by noisy data. In Eq. 2.4, the first part expresses the extent to which \( f \) is not piecewise smooth while the second part measures the disagreement between \( f \) and the observed data [12].

According to Figure 2.8 each variable labeled either \( \alpha \) or \( \beta \) is able to keep its current label (See Figure 2.8.a) or swap to the other label. Figure 2.8.b represents the swap between \( \alpha, \beta \) labels, and variables with label \( \beta \) which have been expanded and variables with label \( \alpha \) which have been decreased. In addition, Figure 2.8.c represents the swap between \( \beta, \gamma \) labels where variables with label \( \beta \) have been decreased and variables with label \( \gamma \) have been increased. Each time only two variables from the move
space will be selected for swapping. If there are $K_1$ variables with a current label in \{\alpha, \beta\}, there will be $2^{K_1}$ possible $\alpha\beta$-swap moves available.

Figure 2.8. (a) Shows current labelling containing three $\alpha$, $\beta$ and $\gamma$ parts – (b) and (c) Show an $\alpha$, $\beta$ and $\gamma$, $\beta$ swap respectively

2.2.5 Optimal multi-region objects segmentation

Recently, global optimization for multi-surfaces [25, 26] inspired by surface segmentation in volumetric images [27] has become a topic of interest. Because of the importance of this innovation, the global optimization for multi-surfaces which is a new version of surface segmentation in volumetric images will be described in this section.

Surface segmentation attended to the 3D segmentation of 3D object. In [27], graph-cut method used for segmentation.

The optimal multi-region model in [25] is similar to the main idea in [27] but addresses some of its deficiencies. In the optimal multi-region model, the authors try to address some existing difficulties in volumetric surface segmentation such as the unfolding problem or the topological limits on shapes by introducing totally different inter-layer arcs. Their model is a multi-region model with one layer per region. The basic graph-cut idea is inspired by a multi-label approach from Boykov et al. [12, 28]. It defines $P$ as a set of image pixels, $L$ as a set of region indices, and $x$ as a set of binary variables $x \in B^{L \times P}$. $x^k_p$ shows a binary index over pixels $p \in P$ and regions $k \in L$. Besides the usual data term and smoothing term in Eq. 2.4, there is an inter-region interaction term in the energy equation in Eq. 2.5. $x^k_p = 1$ means pixel $p$ is included in region $k$. $x^k_p = 0$ means pixel $p$ belongs to the background.
\[ \text{Energy}(f) = \text{Energy}_{\text{smooth}}(f) \left[ \sum_{(p,q) \in N} V_{p,q}(f_p, f_q) \right] + \]

\[ \text{Energy}_{\text{data}}(f) \left[ \sum_{p \in P} D_p(f_p) \right] + \sum_{m,n \in L \atop m \neq n} W^{mn}(x^m, x^n) \]

\[ W^{mn}(x^m, x^n) = \sum_{p,q \in N^{mn}} W^m_{pq}(x^m_p, x^n_q) \]

is a geometric term for making interaction between two regions \( m \) and \( n \). \( N^{mn} \) is an inter-region connectivity which shows all pairs of pixels \( (p, p') \) in which region \( m \) has geometric interaction with region \( n \). Figure 2.9 shows a schematic view of Eq. 2.5 and how two surfaces are connected to each other.

There are three types of geometric constraints that are applied to the constructed graph: containment, exclusion and attraction. Containment constraint requires that region B has to be included in region A. Exclusion constraint requires that A and B cannot overlap at any pixel. Attraction requires that region A should not be grown too far away from region B. Figure 2.10 shows the constraint energy terms between two regions.
This multi-region model has been successfully applied to medical image and normal scene segmentation. It can also be applied to 2D to N-D images. But similar to multi-surface method, its drawback is also in the difficulty of defining some constraints between layers which are necessary in accurate segmentation. All graph-cut based segmentation algorithms suffer from some difficulties in the optimization procedure. These difficulties in some cases stem from the maximum flow/minimum cut algorithms which are the main optimization techniques behind the graph-cut algorithms. These problems are 1- Control of locality: which allows many steps of algorithm to be made in memory constraints. 2- Parallelization: This allows multiple steps of algorithm carried out asynchronously.

2.3 Graphical models and image segmentation

Probability theory provides a fundamental basis for many scientific problems in artificial intelligence, machine learning and image processing areas. In a probabilistic Graphical Model (PGM) a graph comprises nodes (vertices) connected by links (edges or arcs) [29].

In a PGM, each node represents a random variable (or group of random variables) and the links express probabilistic relationships between these variables. In contrast to graph-cut models which utilize an energy function, probabilistic graphical models perform probabilistic inference for segmentation [30]. Probabilistic graphical models provide an easy visualization of the structure of a probabilistic model [3, 19]. In this regard, the whole graph expresses the way in which the joint distribution over all of the random variables can be decomposed into a product of factors.

There are some advantages for probabilistic graphical models, such as:

- By inspection of the graph we can obtain insight into the properties of the model like conditional independence properties
- Complex computations, required to perform inference and learning in sophisticated models, can be expressed in terms of graphical manipulations and mathematical expressions
- The graph as a whole shows the way in which the joint distribution over all of the random variables can be decomposed into a product of factors
There are different types of probabilistic graphical models (PGM) such as Markov Random Field (MRF), Conditional Random Field (CRF) and Bayesian Network (BN). Bayesian Network as a class of probabilistic graphical models was first proposed by Pearl [31] in 1985. Bayesian Networks are also known as directed graphical models (existed links in graphs have a particular directionality indicated by arrows). Directed graphs are useful for showing causal relationships between random variables. In this way it tries to give a more comprehensive representation of the joint probability distribution.

In order to be more familiar with PGM concepts, we describe some notations relating to PGMs. Suppose $\mathcal{H} = (\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_n)$ is a vector of random variables and let $x = (x_1, x_2, \ldots, x_n)$ be a possible value setting for these variables. $P(\mathcal{H} = x)$ or $P(x)$ are used to show the joint probability when all variables in $\mathcal{H}$ are discrete. Conditional probability of value $x_i$ of variable $H_i$ given $H_j = x_j$ is defined as $P(H_i = x_i | H_j = x_j)$.

Let $M, N$ and $Z$ represent three disjoint variables. $M$ is conditionally independent of $N$ given $Z$ , $I(M,N|Z)$ if $P(M|Z,N) = P(M|Z)$. Bayesian Networks encode a factorization for a joint probability distribution. It is formulated as Eq. 2.6.

$$P(x) = P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | pa(x_i))$$  \hspace{1cm} (2.6)

Where $pa(x_i)$ shows any parents of $x_i$. $x = \{x_i\}$ is a set of random variables.

Markov Random Fields are known as undirected graphical models (existed links in graph do not carry arrows and have no directional significance). Undirected graphs are suitable for expressing soft constraints between random variables.

In graph-cut approaches prior knowledge about image data are usually not considered. In order to overcome this weakness, Zhang et al. [3, 19] proposed a combinatorial probabilistic framework of a directed (e.g. Bayesian Network) and undirected (e.g. Conditional Random Field (CRF) or Markov Random Model (MRF)) graphical models. It is inspired by the human’s abilities to segment an image with additional constraints such as contour smoothness, connectivity, object shape, etc.

Many methods have been proposed for image segmentation based on MRF, CRF, Bayesian Network (BN) and their combination, such as the MRF hybrid model of Hinton et al. [32], the BN and MRF model of Murino et al. [33] and Liu et al. [34]. In the recent method by Zhang and Ji [3], the causal and non-causal relationships among all the image entities are considered. Figure 2.11 aims to facilitate region-based (upper
cloud) and edge-based (lower cloud) segmentation in an image by unifying CRF and BN. This approach consists of three parts, CRF, BN and a combination of these two models. The CRF model (undirected) is constructed based on a superpixel model. Superpixels are specified by an over-segmentation pre-processing task on the image which extracts some homogeneous parts (regions). Each part is a single region node in CRF.

![Image of graphical model combining CRF and BN]

Figure 2.11. A structure for unified graphical model combining CRF model and BN model by region nodes [3]

The model is multi-layered, which supposes the CRF is a labelling system which assigns a label based on the local feature vector extracted from image (x) for a superpixel (Y) and BN as an inference system of the type of the edge nodes, given various measurements and constraints. The CRF part plays the region-based image segmentation role, and the BN part has an edge-base segmentation role in this combination. Because of the inability of CRF to represent a causal model, a Bayesian Network is deployed to model the causal relationship. The BN part provides a representation of the relationships between the superpixels (Y), the edge segments (e) and the vertices (v). In this model, BN part is constructed based on a map of edges (e) and (v). The vertex v represents a place which three or more edges intersect. Figure 2.11 shows the model for 4 superpixels. Vertices (v) are a place where three or more edges intersect.
intersect and M denotes measurements of features of edges (e) and vertices (v). Chapter 5 shows some comparative studies to illustrate the performance of the two approaches (graph-cut and probabilistic models) on images with complex background scenes. Due to the importance of using CRF model as a main probabilistic algorithm in this research, we describe it more as an example in the next section.

2.3.1 Conditional random field as an example

In this section, an example of a simple undirected probabilistic graphical model with the concept of CRF is introduced [35-37]. To predict the percentage of cheating in four students, John, Jacky, Julie and Jimmy, all the students are seated in one room in the order shown in Figure 2.12:

![Figure 2.12. Order of seating for four students](image)

All students can see the answers of the others and everybody thinks that the neighboring student studied more, and therefore has the correct answers. However, only Jacky and Jimmy studied well for their exam. Due to this presumption, all of their answers are closely related to each other. Although John and Julie are far from each other, their answers are dependent due to Jacky. In an undirected graphical model (UGM) two variables are dependent on each other if there is a path between them. So, all of the four variables in the example are dependent on each other. But there is an exception here in the concept of conditionally independent variables. For example even though Jimmy and Jacky are dependent, they will be independent if we know the value of Julie.

Now a pairwise UGM model from this example concerning dependencies and independencies is constructed. According to Eq. 2.7, the joint probability of a particular assignment to all of the variables \( x_n \) will be shown as a normalized product of a collection of non-negative factors (function).

\[
p(x_1, x_2, ..., x_N) = \frac{1}{Z} \prod_{i=1}^{N} \psi_n(x_n) \prod_{e=1}^{E} \psi_e(x_{ek}, x_{el})
\]  

(2.7)

\( \psi_n \) is a factor or function for node n, and \( \psi_e \) is a factor for edge e. In this example, the nodes are students and the edges are the only link between two students who are seated next to each other. There is no edge between variables with one or more nodes distance.
\( \psi_n \) is a node potential function which gives a non-negative weight to each possible value of the random variable \( x_n \). For example, \( \psi_3(W_{\text{ans}}) = 65 \), and \( \psi_3(C_{\text{ans}}) = 35 \) means that the third node’s (Julie) answers tend to be more incorrect than correct.

Similarly, \( \psi_e \) is an edge potential which gives a non-negative weight to each possible value of the random variable of edges. Edge is a combination of two node variables \( x_{ek} \) and \( x_{el} \). For example, the students who are seated next to each other receive a higher edge score than those who are far from each other. \( Z \) is a normalization constant which gives the above distribution a summation of one over all possible joint configurations of the variables. The definition of \( Z \) is as Eq. 2.8.

\[
Z = \sum x_1 \sum x_2 \ldots \sum x_N \prod_{n=1}^N \psi_n(x_n) \prod_{e=1}^E \psi_e(x_{ek}, x_{el}) \tag{2.8}
\]

In this example of UGM, there are 4 nodes corresponding to 4 students and each node has 2 states, correct and incorrect. Table 2.2 shows how node potentials have been assigned to each node.

<table>
<thead>
<tr>
<th>Student</th>
<th>Correct Answer</th>
<th>Wrong Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>.25</td>
<td>.75</td>
</tr>
<tr>
<td>Jacky</td>
<td>.9</td>
<td>.1</td>
</tr>
<tr>
<td>Julie</td>
<td>.25</td>
<td>.75</td>
</tr>
<tr>
<td>Jimmy</td>
<td>.9</td>
<td>.1</td>
</tr>
</tbody>
</table>

Table 2.3 shows how edge potentials have been assigned to each edge. The potential amount for having the same state for any two neighbour nodes is two times more than having different state.

Table 2.3. Edge potentials

<table>
<thead>
<tr>
<th>n1\n2</th>
<th>Correct Answer</th>
<th>Wrong Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct Answer</td>
<td>2/6</td>
<td>1/6</td>
</tr>
<tr>
<td>Wrong Answer</td>
<td>1/6</td>
<td>2/6</td>
</tr>
</tbody>
</table>
Table 2.4 shows probability distribution by UGM aligned with all possible configurations of Correct/Incorrect (wrong) answers for each student. Np and Ep are summarizations of node potential and edge potential respectively. According to Eq. 2.7 the Pro-cul is calculated by multiplying all node and edge potentials in a row.

Table 2.4. Probability of each configuration for four students

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>C_ans</td>
<td>C_ans</td>
<td>C_ans</td>
<td>C_ans</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>9</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>648</td>
<td>0.17</td>
</tr>
<tr>
<td>W_ans</td>
<td>C_ans</td>
<td>C_ans</td>
<td>C_ans</td>
<td>3</td>
<td>9</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>972</td>
<td>0.26</td>
</tr>
<tr>
<td>C_ans</td>
<td>W_ans</td>
<td>C_ans</td>
<td>C_ans</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td>0.00</td>
</tr>
<tr>
<td>W_ans</td>
<td>W_ans</td>
<td>C_ans</td>
<td>C_ans</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>108</td>
<td>0.03</td>
</tr>
<tr>
<td>C_ans</td>
<td>C_ans</td>
<td>W_ans</td>
<td>C_ans</td>
<td>1</td>
<td>9</td>
<td>3</td>
<td>9</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>486</td>
<td>0.13</td>
</tr>
<tr>
<td>W_ans</td>
<td>C_ans</td>
<td>W_ans</td>
<td>C_ans</td>
<td>3</td>
<td>9</td>
<td>3</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>729</td>
<td>0.19</td>
</tr>
<tr>
<td>C_ans</td>
<td>W_ans</td>
<td>W_ans</td>
<td>C_ans</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>9</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>54</td>
<td>0.01</td>
</tr>
<tr>
<td>W_ans</td>
<td>W_ans</td>
<td>W_ans</td>
<td>C_ans</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>9</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>324</td>
<td>0.09</td>
</tr>
<tr>
<td>C_ans</td>
<td>C_ans</td>
<td>C_ans</td>
<td>W_ans</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>36</td>
<td>0.01</td>
</tr>
<tr>
<td>W_ans</td>
<td>C_ans</td>
<td>C_ans</td>
<td>W_ans</td>
<td>3</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>54</td>
<td>0.01</td>
</tr>
<tr>
<td>C_ans</td>
<td>W_ans</td>
<td>C_ans</td>
<td>W_ans</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>W_ans</td>
<td>W_ans</td>
<td>C_ans</td>
<td>W_ans</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>6</td>
<td>0.00</td>
</tr>
<tr>
<td>C_ans</td>
<td>C_ans</td>
<td>W_ans</td>
<td>W_ans</td>
<td>1</td>
<td>9</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>108</td>
<td>0.03</td>
</tr>
<tr>
<td>W_ans</td>
<td>C_ans</td>
<td>W_ans</td>
<td>W_ans</td>
<td>3</td>
<td>9</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>162</td>
<td>0.04</td>
</tr>
<tr>
<td>C_ans</td>
<td>W_ans</td>
<td>W_ans</td>
<td>W_ans</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>12</td>
<td>0.00</td>
</tr>
<tr>
<td>W_ans</td>
<td>W_ans</td>
<td>W_ans</td>
<td>W_ans</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>72</td>
<td>0.02</td>
</tr>
</tbody>
</table>

Also, the last column, Probability distribution, is the result of (Pro-cul)/Z. Z, as appears in Eq. 2.8 is obtained by the summation of the Pro-cul column. Noted that in order to simplifying the probability calculation in Table 2.4, all the numbers in each row in Table 2.2 and Table 2.3 have been multiplied with a number, such as 4 or 6 to give an integer, and each row can be multiplied with a different numbers.

Figure 2.13 represents a different condition in which all students were seated in different rooms. It shows the real score which each student can achieve without cheating. Each column shows a student and each row depicts one question. Blue colour
represents a correct answer and red colour is the wrong answer. For example, as shown in Figure 2.13, Jacky and Jimmy, who studied well, achieved the best results for 200 questions. In contrast, John and Julie who did not study well did not achieve good results.

Figure 2.13. A sample of cheating for 200 questions for four students seated in different rooms

Figure 2.14 shows a sampling of 200 questions from the configurations in Table 2.4. As depicted in this Figure, John and Julie achieved extra correct answers by cheating, and also Jacky and Jimmy lost more correct answers because of cheating. In order to achieve the sampling of Figure 2.14, one random number between [0, 1] was chosen. For example, if 0.65 was selected. Referring to Table 2.4, 0.17 was achieved for the first configuration from the probability column; then 0.17 + 0.26 = 0.43 for the second configuration; 0.17 + 0.26 + 0 = 0.43 for the third configuration; 0.17 + 0.26 + 0 + 0.03 = 0.46 for the fourth configuration; 0.17 + 0.26 + 0 + 0.03 + 0.13 = 0.59 for the fifth configuration; and finally 0.17 + 0.26 + 0 + 0.03 + 0.13 + 0.19 = 0.78 for the sixth configuration. As 0.78 exceeded the threshold of 0.65, we stopped the calculation. The achieved configuration, shown in the sixth row of Table 2.4. is, w_ans, c_ans, w_ans,
c_ans. If this process for the calculations is carried out for additional configurations up to 200 questions, the result will be shown in Figure 2.14.

Figure 2.14. A sample of cheating for 200 questions for four students seated in the same room

Regarding the above example which introduced us simply to the concept of conditional random field with 4 variable nodes, one extension of this algorithm has been proposed for image segmentation [3]. Algorithm 2.2 shows how CRF is applied for image segmentation. The concept behind the algorithm is exactly the same as the example of the students above. Figure 2.15 shows how an oversegmentation method resulted in some superpixels. In this Figure, there are some superpixels which are mapped to CRF variables, and the extracted features like colour and texture are the value of any superpixel. Each superpixel (specified with white colour) in the original image has been mapped with one region node (black colour) in the CRF part. The grey node x represents the whole image and y_i, y_j represent the labels of regions. The links in the CRF model obtained from the pairwise relationship. It means if there is a link between two adjacent superpixels, a link also will be appeared between two CRF nodes in the model.
Figure 2.15. An over-segmented image with related super-pixels which show the nodes in CRF and the links between them are the connections between the nodes

Algorithm 2.2. Image segmentation with an unified graphical model algorithm [3]

- Over-segment the image & extract super-pixels using Watershed Segmentation Method or Normalized-cut method (No. of super-pixels can be 15 or more)
- Node measurement using feature vector extracted from each superpixel as below:
  - For Colour images: using the average CIELAB colour and their standard deviations as the local features $x_i$ (for each superpixel) –here the length of the feature vector is 6
  - Because each node has to be labeled, a 3-layer perceptron with an structure of 6 nodes in the input layer, 35 nodes in the hidden layer and 1 node in the output layer is used
  - For Grey-Scale images: using the average intensity and 12 Gabor textures as the local features $x_i$ (for each superpixel) –here the length of the feature vector is 13– the Gabor textures are calculated by filtering the grey-scale image with a set of Gabor filter banks. The average magnitude of the filtered image in each superpixel region is used as the Gabor feature. Gabor filter banks with 3 scales and 4 orientations are used.
  - Because each node has to be labeled, a 3-layer perceptron with a structure of 13 nodes in the input layer, 25 nodes in the hidden layer and 1 node in the output layer is used
  - Output of classifier is +1 or -1. It is a classifier based on a multilayer perceptron (MLP)
CRF Model: \( \frac{1}{Z} \prod_{i \in V} \phi(y_i, x_i) \prod_{j \in N_i} \exp(y_i y_j \lambda^T g_{ij}(x)) \), as illustrated in the above example the first part is the node potential and second part is the edge potential.

\( \phi(y_i, x_i) = \frac{1}{1 + \exp(-y_i \text{net}(x_i))} \) (Net\( (x_i) \) is calculated by Neural Network)

\( g_{ij}(x) = [1, |x_i - x_j|^T] \), \( x_i \) and \( x_j \) are feature vectors of node \( i \) and \( j \) respectively.

Maximum likelihood estimation (MLE) is used to find parameter \( \lambda \).

### 2.4 Artificial neural network

ANNs are important methods in supervised learning due to its properties such as adaptability, ability to generalize and learning capability [38]. A feed-forward neural network (FFNN) is a well-known method in many applications. The feed-forward neural network is known as Multi-Layer Perceptron (MLP). Figure 2.16 shows an example of a MLP. MLPs are widely used for classification. Feed-forward NNs has an input layer and an output layer beside one or more hidden layers. In this model, the information flow from input to output goes through hidden layers and is affected by weights and activation functions. The training of an ANN involves adjusting the interconnection weights to achieve lower error. BP algorithm is usually used to train Feed-forward NNs. The NN used in Algorithm 2.2 for node labelling is the same as Figure 2.16.

![Feed-forward NNs with two hidden layers](image)

Figure 2.16. Feed-forward NNs with two hidden layers
Neural network as a biologically inspired algorithm provides a programming environment facilitating a computer to learn from observational data.

Deep learning refers to the robust techniques that facilitate neural networks to learn. Deep convolutional networks are the most widely used types of deep network. Neural networks and deep learning as the state-of-the-art machine learning approaches provide the best solutions to many problems in image recognition, speech recognition, and natural language processing.

Deep learning especially convolutional networks has proven very successful for a variety of recognition tasks in various applications of computer vision, such as image classification, object detection, image retrieval, semantic segmentation, and human pose estimation, which are the main tasks of image understanding [39-42].

Detecting overlapped pedestrians in an image is one of the most important applications of deep learning [43]. In this model of recognition a deep learning implemented to estimate the visibility statuses of multiple pedestrians and recognize overlapped pedestrians via a mutual visibility deep model. The mutual visibility deep model designed mainly for 2-pedestrians parts.

A scene classification framework performed by CNN on the SUNRGBD and NYU Depth Datasets proposed by Wang et al.in [44]. A combination of CNN features extracted from three different modalities, RGB, HHA and SA of RGB-D data in addition to global features are computed to construct the proposed deep learning.

In medical image segmentation [45], a single CNN has been trained to perform three different tasks of segmentation: 1- segmentation of six tissues in MR brain images 2- segmentation for the pectoral muscle in MR breast images, and 3- segmentation of the coronary arteries in cardiac CTA.

Identification of the imaging modality, visualized anatomical structures and the tissue classes were another important tasks of learning for the proposed system [46].

Isin et al. proposed another deep learning model in medical imaging for MRI-based brain tumor segmentation. The large amounts of MRI-based image data could be efficiently processed and evaluated by the proposed deep learning model [47].

Large scale object recognition with Convolutional Neural Networks (CNNs) is also the greatest successes of Deep Learning [48, 49].
In all proposed models of deep learning mostly based on CNN, the results show better performance than the other traditional methods. The main power of CNN stems from learning data representations directly from data in a hierarchical layer based structure.

2.5 Spatial fuzzy clustering for image segmentation

Fuzzy c-means (FCM) clustering algorithm, as a method of unsupervised clustering, has been mostly used in different areas of image and data clustering such as: image segmentation, cell imaging and geology. In 1973, the FCM algorithm was proposed by Dunn and later, in 1981, the algorithm was modified by Bezdek [50]. The FCM algorithm aims to classify an image based on a similar feature space. The goal of the algorithm is minimizing $P$ in Eq. 2.9.

$$P = \sum_{i=1}^{N} \sum_{j=1}^{V} M_{ij} u \|y_j - c_i\|^2$$  \hspace{1cm} (2.9)

Where $M_{ij}$ represents the membership of pixel $y_j$ in the $i$th cluster, $c_i$ is the $i$th cluster center, $\|\cdot\|$ is a norm metric and parameter $u$ is a constant to control the fuzziness of the result.

The conventional FCM algorithm does not take any advantage of the pixel correlations. Neighbourhood pixels in an image have a higher correlation in features than the pixels that are not in similar vicinity. Spatial relationship of image pixels is an important feature for image segmentation that could be achieved from pixel correlations. Chuang et al. [51] proposed a spatial FCM which incorporates a spatial function into the membership function as Eq. 2.10.

$$M'_{ij} = \frac{M_{ij}^p \rho^k}{\Sigma_{l=1}^{t} M_{ij}^p \rho^k}$$  \hspace{1cm} (2.10)

$p_{ij} = \Sigma_{k \in NB(y_j)} M_{ik}$ is a spatial function to control spatial information. $NB(y_j)$ shows a square window with the center on $y_j$ pixel in the spatial domain.
2.6 Fuzzy level-set for image segmentation

Compared with FCM models which utilize pixel classification for image segmentation, level-set methods exploit dynamic variation boundaries. Level-set methods utilize a combination of active contours and a time dependent PDE function $\chi(t, x, y)$ for image segmentation [52].

In 1988, Osher and Sethian were the pioneers who introduced the level set method for following fronts propagating with curvature-dependent speed [53, 54]. In this thesis we use a combinational framework of fuzzy c-means and level-set method [52]. In this framework, the results of fuzzy c-means are utilized for automating initialization and controlling parameters of level-set model. It benefits from spatial fuzzy c-means to enhance determining contour of interest in medical images. The fuzzy level set method applied for different applications such as, video/image processing, graphics and medical imaging [52, 55].

2.7 Summary

In this chapter, we investigated two categories of image segmentation algorithms: 1- graph-cut models, which are based on max-min algorithms, and 2- probabilistic graphical models. Although each category performed well for image segmentation, they have some advantages and disadvantages.

In graph-cut models, finding the shortest path for sending flow is the goal (the most beneficial advantage of this model) but looking for the shortest path is expensive and time consuming. These difficulties in some cases stem from the maximum flow/minimum cut algorithms which are the main optimization techniques behind the graph-cut algorithms. These problems are 1- Control of locality: which allows many steps of algorithm to be made in memory constraints. 2- Parallelization: This allows multiple steps of algorithm carried out asynchronously.

There are also some benefits for graphical models:
- By inspection of the graph we can obtain insight into the properties of the model like conditional independence properties
Complex computations, required to perform inference and learning in sophisticated models, can be expressed in terms of graphical manipulations and mathematical expressions.

The graph as a whole shows the way in which the joint distribution over all of the random variables can be decomposed into a product of factors.

The disadvantages of graphical models are as below:

This model has a fixed complexity that will conclude in either under-fit or overfit data. Designer must decide about everything like interactions, non-linearities, etc. Any incorrect decision may result in highly biased models and invalid hypothesis tests. This is difficult for correlated variables and problems with many features. Data must be transformed to a discretised, square root or log transforms to match the parametric form. The next chapters show how we applied both models of graph-based image segmentations on medical and natural images. The evaluation results represent promising achievements from our proposed segmentation approaches.
3

3.1 Introduction

Fluorescence microscopy, such as two-photon imaging, provides accurate imaging of living tissues by increasing excitation light penetration and reducing the background emission signal. In combination with fluorescence staining, fluorescence microscopy results in useful information about cells in image cytometry. However, fluorescence image analysis is difficult due to factors such as optical setup, properties of the detectors and the photophysics inherent to the diverse fluorescent probes [56]. In this regard, visual assessment of fluorescence microscopy images is time consuming. Analysis of fluorescence microscopy images is facilitated and accelerated by fast automated detection methods that can be achieved by computer analysis of images produced by a digital camera attached to a fluorescence microscope [57, 58]. Cell image segmentation is the first step of all automatic image analysis in high throughput cell screening applications. However in high-throughput tasks, the manual or semi-automatic detection of cells over a cluttered background is known to be tedious due to the massive amount of image data. Therefore, automatic approaches are required which efficiently address the problems of different cell types and image artefacts such as intensity inhomogeneity and noise. In order to automatically partition an image into two segments, object and background, many methods have been used such as random walkers [59], active contours [21], geodesic active contours [22], shortest path techniques [23, 24], region growing and split and merge [60]. Although there are many existing algorithms and techniques for image segmentation, in this thesis we focus only on graph-cut segmentation methods. The most important reason is the ability of graph-cut algorithms to search the exponential space of solutions of a problem for finding the most optimum one in a polynomial time.
As the graph-cut methods are labelling approaches, the main purpose in labelling problems is to assign a label from a set of labels to data in an observation data set. This is performed based on a criterion that specifies the cost of assigning a label to data [5, 13, 37]. In an image, all the pixels in the image constitute the set of data (observation) which needs to be labelled. This observation in an image is not limited only to pixels of an image, it can be salient points in an image, depth measurements from a range scanner, or intensity measurements from CT/MRI.

Different methods of graph-cut models such as binary and multi-label [12], inspire the concept of labelling as mentioned above. Image segmentation algorithms in the graph-based category can be broadly divided into two types: deterministic and probabilistic [3]. The graph-cut approach [61] is categorized as a deterministic approach of segmentation. Energy minimization [5] and inference are the main tasks, respectively, of the graph-cut algorithms [7-9, 11] and probabilistic models. Graph-cut methods are based on maximum flow algorithms used for energy minimization. There are many methods for implementing the maximum flow algorithms, such as push-relabel methods [17] and algorithms based on Ford-Fulkerson [18]. Algorithms such as max/min cut and normalized cut belong to the deterministic approach, and algorithms such as Markov Random Field [4] and Bayesian Network belong to the probabilistic graphical approach. The max/min cut approach will be used in every graph-cut model in the following parts using the Ford-Fulkerson algorithm.

The main contribution of this work is proposing a Bayes model of interactive graph-cut that takes all advantages of a Bayes-Gaussian model [62] and graph-cut approach. Also, we show how the proposed approach is effective in bio-cell image segmentation compared with the conventional existence approaches.

3.2 Bio-cell image segmentation using Bayes graph-cut model

The Graph-cut method was proposed by Boykov and Jolly [13] for energy-based image segmentation. Its main purpose is about interactively finding the $s - t$ cut of minimal total cost with 2 labels (binary labels) in a graph. In interactive graph-cut approach an image could only be segmented into two parts, object and background, with soft
constraints on boundary and region [63]. In their model, an undirected graph \( G(V, E) \) (Figure 3.1) is defined in which \( V \) and \( E \) represent graph nodes (equal to image pixels \( P \)) and graph edges, respectively. Two extra terminal nodes, \( S \) (source) and \( T \) (sink), which represent the object and background labels, are also defined. Based on the two terminal nodes, all the edges will be divided into two classes: \( N \)-links and \( T \)-links.

Each node has two terminal links with terminal nodes, one for \( S \) and one for \( T \). These links are known as \( T \)-links. \( N \)-links or neighborhood links are the links between pixels and they construct an arbitrary neighborhood system. Each edge has a dedicated cost or weight and an \( s-t \) cut is a subset of edges with minimum cost partitioning the graph (image) into two parts, source and terminal. In this regard, the purpose of segmentation is to minimize the energy function in Eq. 3.1 which is the sum of regional (cost of \( T \)-links) and boundary terms (cost of \( N \)-links).

\[
E(M) = \alpha \text{Region}(M) + \text{Boundary}(M)
\]  

(3.1)

\( M \) defines a segmentation and \( \alpha \geq 0 \) is a coefficient which emphasizes the regional term. In the conventional graph-cut model the regional term is calculated by a histogram model. According to the compatibility of every pixel’s intensity by region histogram or background histogram, the position of each pixel for being part of region or background will be defined. Assume \( P \) as a set of pixels in an image and \( M = (M_1, ..., M_p, ..., M_{|P|}) \) as a binary vector for assigning two binary labels “object” and “background” to each pixel \( p. M_p \) can be either object or background.

![Figure 3.1. Graph-cut model](image)
Regarding the graph-cut model, a segmentation energy function that needs to be decreased in cost is defined as Eq. 3.1. In this thesis, instead of trying to fit the intensity of pixel \( p \) into a predefined model of intensity, such as a histogram, in Eq. 3.2 and Eq. 3.3, a Bayes framework [64] has been proposed.

\[
R_p("obj") = -\ln P(I_p|"obj") \\
R_p("bkg") = -\ln P(I_p|"Bkg")
\]  

Eq. 3.2

Eq. 3.3

\( I_p \) represents the intensity of pixel \( p \). \( P(I_p|"obj") \) and \( P(I_p|"Bkg") \) depict intensity distribution of seed points in a histogram for object and background. The proposed approach attempted to specify Region(\( M \)) in Eq. 3.1 according to Bayes model. In this regard, due to having only two regions, “object” and “background”, two events can be assigned to each pixel as follows: \( e1 \) in the presence of object; and \( e2 \) in the presence of background. In order to decide which event is probable, one of the two probabilities, \( e1 \) or \( e2 \), can be chosen. Then one of two decisions will be achieved: 1) The object is present and thus should be chosen by the segmentation procedure (\( D1 \)); 2) The background is present and thus should be chosen by the segmentation procedure (\( D2 \)).

![Figure 3.2. Gaussian Bayes model](image)

In fact the Bayes procedure is utilized to decide which event is correct. Finally, the following four conditional probabilities are implemented for the combinations of decisions in the hypotheses test: (also shown in Figure 3.2)

\( P(D1|e1) \) or \( P(obj|obj) \) is the probability of declaration that the pixel is recognized as an object when it is actually an object.
\( P(D2|e1) \) or \( P(bkg|obj) \) is the probability of declaration that the pixel is recognized as a background when it is actually an object.
\( P(D1|e2) \) or \( P(obj|bkg) \) is the probability of declaration that the pixel is recognized as an object when it is a background.
\( P(D2|e2) \) or \( P(bkg|bkg) \) is the probability of declaration that the pixel is recognized as a background when it is a background.

In the statistical terminology, the second probability, which is the probability of misclassifying the object pixels as background, \( P(D2|e1) \), when it is actually object, is called Type I risk. On the other hand, \( P(D1|e2) \), which can be considered the probability of misclassifying background as object pixels when it is actually background, is called type II risk. The consequence of each combination of hypothesis and decision is quantified with an associated loss. Thus the region calculation will be evaluated by two Gaussian functions. Therefore the total risk for region \( (M) \) recognition is calculated as Eq. 3.4.

\[
\text{Region (M)} = P(obj.) P(D2|e1) + P(bkg.) P(D1|e2)
\] (3.4)

And finally boundary \( (M) \) will be computed according to Eq. 3.5 and Eq. 3.6.

\[
\text{Bound}(M) = \sum_{(p,p')} \text{Bound}_{p,p'} \cdot \delta_{m_p\neq m_{p'}}
\] (3.5)

\[
\text{Bound}_{p,p'} = \exp \left( -\frac{(I_p-I_{p'})^2}{2\sigma^2} \right) \cdot \frac{1}{\text{dist}(p,p')}
\] (3.6)

Let \( P = \{p \in P\} \) denote the set of data elements (pixels or voxels) and \( I_p: P \rightarrow IR \) denote a given intensity of image. Now \( D1 \) and \( D2 \) are redefined as follows; all pixels of the image that lead the segmentation procedure to choose decision \( D1 \) fall in the “object”, whereas all pixels of the image that result in decision \( D2 \) fall in the “background”. The two hypotheses, \( D1 \) and \( D2 \), are considered in association with pdfs \( P(I_p|D1) \) and \( P(I_p|D2) \), respectively. Now the region cost based on the Bayes risk for segmenting an image to object or background can be written as Eq. 3.7.
\[ R = \sum_{p \in \text{bkg}} P(I_p | p \in \text{obj}) P(p \in \text{obj}) \] (3.7)

\[ + \sum_{p \in \text{obj}} P(I_p | p \in \text{bkg}) P(p \in \text{bkg}) \]

Taking a logarithm, Eq. 3.7 can be rewritten as Eq. 3.8.

\[ R = \sum_{p \in \text{bkg}} \ln(P(I_p | p \in \text{obj}) P(p \in \text{obj})) \] (3.8)

\[ + \sum_{p \in \text{obj}} \ln(P(I_p | p \in \text{bkg}) P(p \in \text{bkg})) \]

Assuming that Gaussian mixture model conditions are satisfied by the image, therefore \( P(I_p | p \in \text{obj}) \) and \( P(I_p | p \in \text{bkg}) \) follow the Gaussian distribution [62]. Now Eq. 3.8 can be reformulated as Eq. 3.9.

\[ R = \sum_{p \in \text{bkg}} \ln(N(I_p | \mu_{bkg}(p), \sigma^2_{bkg}) P(p \in \text{obj})) \] (3.9)

\[ + \sum_{p \in \text{obj}} \ln(N(I_p | \mu_{obj}(p), \sigma^2_{obj}) P(p \in \text{bkg})) \]

where \( \mu_{bkg}, \mu_{obj}(p), \sigma^2_{bkg}, \) and \( \sigma^2_{obj} \) denote the mean and the covariance matrix of the object and background respectively, and \( N(I_p | \mu_i(p), \sigma^2_i) \) is defined as Eq. 3.10.

\[ N(I_p | \mu_i(p), \sigma^2_i) = \frac{1}{\sigma^2_i \sqrt{2\pi}} \exp \left( -\frac{1}{2} \frac{(I_p - \mu_i(p))^2}{\sigma^2_i} \right) \] (3.10)

\[ i = \{ \text{obj}, \text{bkg} \} \]
3.3 Experimental results and analysis

![Figure 3.3](image1.png)  
(a) An image of human disorder cells (b) Its ground truth [26]

![Figure 3.4](image2.png)  
(a-b) The segmentation result of multi-label graph-cut method

This section shows experimental results of applying Bayes proposed graph-cut segmentation method to the Broad Bioimage Benchmark Collection (BBBC) data sets [65]. These microscopy data sets are free to download, and in some cases ground truth exists. The data set includes human HT29 colon cancer cells images with the size of 512*512 pixels for an image. These fluorescence images are the main data, which
facilitate any spatial and temporal measurement of fluorescent molecules existing in a tissue, cell, or the whole body of an animal [66].

We tested this model on 5 images of BBBC data set in 1 and 3 channels for foreground/background segmentation. This section started with an example of multi-label graph-cut to clarify the improvement of the proposed model. Figure 3.3.(a-b) present an image of a Human Disorder cell from the BBBC data set and the image’s ground truth, respectively. The cell cluster within the red circle in Figure 3.3(b) are in close proximity and the ground truth clearly separates them. Figures 3.4.(a-b) show the segmentation results of the multi-label graph-cut method mentioned in section 1 (the original image has small noise). In Figure 3.4(a), the blue colour around each object represents how it has been separated from the others. The result of segmentation for separated objects has more than 80 percent accuracy, but the cell cluster is not accurately segmented. The method is successful in recognizing objects that are clearly separated but has problems with extracting object boundaries in congested areas.

Figure 3.5. (a) Original image (b) Ground truth (c) Segmentation result of the proposed method

Figure 3.4.(b) shows the same segmentation as a binary image, highlighting the lack of segmentation in the cell cluster. The ability of interactive binary graph-cut in separating congested cells more accurately motivated us to propose our model based on the binary graph-cut model. Figure 3.5 shows an example of an original image and the result of segmentation based on the proposed method. Figure 3.5 shows the proposed segmentation method on a clean image. Figure 3.6 shows the segmentation result of the binary (object/background) graph-cut method (first row) and the Bayes proposed one (second row) on a noisy image.
Figure 3.6. (a-c) First row shows the segmentation result of the binary (object/background) graph-cut method. Second row shows the segmentation result of the binary graph-cut method based on Bayes theorem (image noise has been increased from left to right for both rows).

Figure 3.5 and Figure 3.6 depict more accurate cell segmentation than multi-label graph-cut using binary graph-cut. The objects and boundaries are obtained with better accuracy in these Figures compared with Figure 3.4. Binary graph-cut segmentation method was successful in separated and congested areas. But there is still a difficulty in that it does not inherit the benefits of the Bayes method.

In Figure 3.5 and in the second row of the Figure 3.6, instead of trying to fit the intensity of pixel \( p \) into a predefined model of intensity, such as a histogram in (3.2) and (3.3), the proposed Bayes framework has been applied. In this experiment we started with a clean image to apply binary and Bayes graph-cut methods. In the case of clean image, the result of the proposed Bayes graph-cut was better than the conventional binary one. Then we combined the images with a small Gaussian noise (0.000001). The result of both methods was acceptable. Finally we increased the noise to 5% Gaussian noise.
As the Figure 3.6 and Table 3.1 show, the accuracy here for the proposed method, is better than the binary graph-cut model. The proposed method takes all the benefits of Bayes model that shows stronger results in noisy images and in congested cell areas than conventional binary graph-cut models. As it is mentioned before, the first step in the proposed model is specifying some pre-defined points by user.

We tested our model with different number of foreground and background seed points which interactively selected by a human user. First we started with 50 seed points for foreground and 30 seed points for background. Then we decreased the number of seed point up to 10 points for foreground and 5 for background. According to the Table 3.1 the results of our proposed model show that it is not dependent to the predefined points. This behaviour inherited of binary graph-cut model as represented in the Table 3.1. But inspiring Gaussian/Bayes model, the proposed model improved in accuracy and Error rate. In this section, the performance of our Bayes graph-cut model is compared with the watershed algorithm (WA) [67], conventional FCM [50, 68], Otsu thresholding (OT) [69] and Chan-Vese (CV) [70] segmentation models. Also, we take into account the conventional graph-cut method to evaluate the accuracy and precision of the results.

According to Table 3.1, graph-cut approaches (last 4 rows) performed better than the other methods. In addition the performance accuracy result for the proposed Bayes graph-cut is better than the binary model which is conventional model.

Table 3.1: Segmentation accuracy results of BBBC008V1 for different methods [57]

<table>
<thead>
<tr>
<th>Image of Human H29 Colon Cancer Cells of BBBC-8V1 Data Set</th>
<th>Noisy Image of Human H29 Colon Cancer Cells of BBBC-8V1 Data Set (5% Gaussian noise)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segmentation Method</td>
<td>PER(%)</td>
</tr>
<tr>
<td>CV</td>
<td>4.68</td>
</tr>
<tr>
<td>FCM</td>
<td>5.39</td>
</tr>
<tr>
<td>WA</td>
<td>5.91</td>
</tr>
<tr>
<td>OT</td>
<td>4.82</td>
</tr>
<tr>
<td>Binary Graph-Cut (10,5)</td>
<td>6.15</td>
</tr>
<tr>
<td>Binary Graph-Cut (50,30)</td>
<td>6.23</td>
</tr>
<tr>
<td>Proposed Bayes Graph-Cut (10,5)</td>
<td>2.26</td>
</tr>
<tr>
<td>Proposed Bayes Graph-Cut (50,30)</td>
<td><strong>2.04</strong></td>
</tr>
</tbody>
</table>
The segmentation accuracy is computed using the Dice index and Recognition Error Rate (RER) as given in Eq. 3.11 and Eq. 3.12. In the case of the foreground/background we have \( n = 1 \).

\[
\begin{align*}
\text{Dice}(R, S) & = \frac{1}{n} \sum_{i=1}^{n} \frac{2 |R \cap S|}{|R| + |S|} \\
\text{RER}(\%) & = \frac{T_{\text{cell}} + T_{\text{background}}}{n} \times 100
\end{align*}
\tag{3.11}
\tag{3.12}
\]

Here, \( R \) denotes the ground truth, \( S \) denotes the segmented image and \( n \) is the number of segmented areas. \( T_{\text{cell}} \) and \( T_{\text{background}} \) denote the number of pixels incorrectly classified as the cell and background, respectively. Also, the number of false negatives (FN) reported in Table 3.1. In terms of speed, the speed is \( O(mn^2 |C|) \) \[12, 28\] for graph-cut, where \( m \) and \( n \) are the number of edges and nodes, respectively, and \( |C| \) is the cost of the minimum cut for the graph-cut model.

### 3.4 Conclusion

We proposed a new graph-cut approach to bio-cell image segmentation based on Bayes model. We compared our proposed method with state-of-the-art algorithms and compared their performance on several images from one popular dataset of human disorders. Our results show that for bio-cell images with complicated or unclear cells, the proposed Bayes based method is able to take into account multiple prior constraints and exhibits superior performance. In future work, we plan to propose a combination work of probabilistic approach with deterministic graph-cut models. Also, we will apply our proposed method on other different bio-cell images and expand our experimental results on various data sets.
4 ENSEMBLE-BASED IMAGE SEGMENTATION FOR FLUORESCENCE MICROSCOPY IMAGES

4.1 Introduction

Fluorescence microscopy is a main component of biomedical studies, and cellular imaging is a method of determining the subcellular location of proteins [71, 72]. Fluorescence microscopy images are prepared by shining excitation light on the specimen to activate fluorescence [73, 74]. It provides an appropriate environment for researchers to understand the structure and architectural dynamics of the complex cellular and molecular living organisms, which is the main purpose of biological research in the postgenomic era.

The aim of biological imaging experiments is to accurately and automatically extract structural, spatial, and functional quantitative information about some biological phenomenon [73, 75]. Some of the critical problems in microscopic image analysis to extract useful information are restoration, registration, segmentation and others. In this thesis, we only focus on cell image segmentation.

Cell image segmentation is a significant part of many cytometric analyses [76]. In the cell segmentation process, nuclear segmentation is the first step and many simple operations like cell counting and cell-cycle assignment is often performed after this process. Automatic methods like machine learning with the ability to deal with different cell types and image artifacts are required because semi-automatic and manual
segmentation performed by medical professionals are exceedingly time-consuming, highly subjective, and irreproducible. There are many existing algorithms and techniques for cell image segmentation [52, 56, 75, 77-81]. Over the past few years, the use of machine learning methods to recognize all major patterns of subcellular locations has been convincingly presented through different feature sets and classifiers. Graph-cut [77-79], Bayes graph-cut [75, 80] and Level set segmentation [52] approaches of machine learning have been widely used for image cell segmentation with promising results [81]. In [56] Pecot et al. proposed a constrained graph-cut for 2D and 3D microscopy image segmentation based on choosing superpixels for constructing a graph instead of all pixels in an image. Beheshti et al. [75] proposed a Bayes model based on binary graph-cut that is able to perform foreground and background segmentation. Their method inspires the benefits of a Gaussian model in Bayes theory and is more powerful than traditional graph-cut when there is a noisy environment for microscopy cells. In [82] Ersoy et. al. proposed a level set method as a combination of level set and graph partitioning approaches. In [83] a multiple kernel local level set segmentation has been proposed. The model incorporates spatial constraints into data in order to achieve microscopy cell image segmentation. In [57, 70, 84, 85] a region-based level set method applied for image segmentation.

There are some advantages and disadvantages for graph-cut models [5, 18, 25] and level set approaches [86, 87]. Both of them are popular and accurate segmentation methods that are now used with appropriate accuracy. In graph-cut methods which are based on maximum flow/minimum cut approach, the purpose is finding the shortest path in the graph, but finding this shortest path is expensive [12, 13]. Also, low computational efficiency is the most important drawback of level set models. In order to tackle these kinds of problems, in this thesis we show how applying weighted ensemble methods on biomedical imaging provides high accuracy and efficient use of computational resources. An unsupervised ensemble-based microscopy image segmentation used in [88]. The authors proposed a Markov random field ensemble model for U2OS microscopy cell segmentation. Mohapatra et. al. [89] offered an ensemble classifier system for early diagnosis of lymphoblastic leukemia in blood with high accuracy. The achieved accuracy by ensemble in papers was promising.
To the best of our knowledge, the weighted ensemble methods [90-93] with the proposed structure in this thesis have not been used for fluorescent microscopy image segmentation.

The major contribution of this thesis is twofold.

- We propose a weighted ensemble framework for accurate and robust segmentation of cancer bio-cell images based on four state-of-the-art segmentation methods namely Bayes graph-cut algorithm, binary graph-cut algorithm, spatial fuzzy C-means, and fuzzy Level-Set algorithm. We apply the aforementioned segmentation algorithms on bio-cell images in order to provide an appropriate infrastructure for a weighted ensemble model. Then, the results of different models will be sent to the weighted ensemble algorithm to make a final decision based on the weighted majority. The ensemble based segmentation method takes advantages of all the member methods to improve the segmentation accuracy.

- Comprehensive evaluation and comparisons between four state-of-the-art methods and the proposed weighted ensemble method is also performed. We exploit Kappa and Naïve statistical measures in order to provide comprehensive evaluation of both overall and class wise (foreground and background) performances of the proposed framework.

Results revealed that the proposed weighted ensemble method is better than the compared state-of-the art methods both in terms of accuracy and robustness.

In this thesis, we used two datasets (simulated and real) to compare ensemble results with other methods. We show how the proposed approach is effective in cell image segmentation compared with the conventional existing approaches. This model tries to recognize cells or objects from background with high accuracy and also make a visible separation between each of the two connected cells.
4.2 Proposed ensemble-based framework

Figure 4.1. Overview of our proposed framework based on robust cell segmentation. The diagram from the left to right represent image segmentation through four Bayes graph-cut, binary graph-cut, fuzzy LSM, SFCM and finally weighted majority based ensemble.

The main idea of an ensemble method as a machine learning algorithm is a collective decision making. Classifiers are the most important infrastructure of an ensemble method and their vote prediction results in a decision making for a new data point. The diversities of clustering methods lead to the diversities of their predictions and accuracy. In literature [94], two voting mechanisms are available: (1) majority voting and (2) consensus voting. The consensus requires all classifiers reach a decision and a voting mechanism assigns the class label only if all the members agree. In the majority voting mechanism, a class label is assigned depending on the majority of the classifiers that has assigned that label. Majority voting is preferred in this study regarding the time and accuracy achieved by experimental results. Individual privacy of each classifier is preserved in this method and since only the importance of counting votes is the issue, decisions can be reached much more quickly with majority rule. Due to its constraining nature, consensus voting is found to be less efficient compared with majority voting to address time-sensitive issues.

Furthermore the accuracies obtained by an individual member of the ensemble are not the same so we have used a weighted majority framework. We exploit a weighted voting framework as well as its probabilistic set-up [93] for the weighted majority framework as follow.

Let us define a set of classes as $\phi = \{\psi_1, ..., \psi_c\}$ and the number of classifiers in the ensemble as $L$. Then the probability can be expressed as $Pr(\psi_k | s)$, $k = 1, ..., c$ where $s = [s_1, s_2, ..., s_L]^T$ is a label vector. Since the classifiers are independent in terms of their decision, Eq. 4.1 is defined as follow:
\[ Pr(\psi_k | s) = \frac{Pr(\psi_k)}{Pr(s)} \prod_{i=1}^{L} Pr(s_i | \psi_k) \] (4.1)

\( I_+^k \) denotes the set of indices of classifiers which suggested \( \psi_k \), and by \( I_-^k \) the set of indices of the classifiers which suggested another class label. The probability of interest becomes as Eq. 4.2.

\[ Pr(\psi_k | s) = \frac{Pr(\psi_k)}{Pr(s)} \prod_{i \in I_+^k} Pr(s_i = \psi_k | \psi_k) \prod_{i \in I_-^k} Pr(s_i = \tilde{\psi}_k | \psi_k) \] (4.2)

Let \( Pr(s_i = \psi_k | \psi_k) = p_i \) and \( Pr(s_i = \psi_j | \psi_k) = \frac{1-p_i}{c-1} \) for any \( k, j = 1, \ldots, c \neq k \).

Now the Eq. 4.2 becomes as Eq. 4.3 and Eq. 4.4 then Eq. 4.5.

\[ Pr(\psi_k | s) = \frac{Pr(\psi_k)}{Pr(s)} \prod_{i \in I_+^k} p_i \prod_{i \in I_-^k} \frac{1-p_i}{c-1} \] (4.3)

\[ Pr(\psi_k | s) = \frac{1}{Pr(s)} \prod_{i=1}^{L} \frac{1-p_i}{c-1} \frac{Pr(\psi_k)}{Pr(s)} \prod_{i \in I_+^k} \frac{p_i(c-1)}{1-p_i} \] (4.4)

\[ \log(Pr(\psi_k | s)) = \log \left( \prod_{i=1}^{L} \frac{1-p_i}{c-1} \frac{Pr(\psi_k)}{Pr(s)} \right) + \log(Pr(\psi_k)) + \sum_{i \in |I_+^k|} \log \left( \frac{p_i}{1-p_i} \right) + |I_+^k| \times \log (c-1) \] (4.5)

By dropping the first term, since it does not have any impact on the class decision making and expressing the classifier weight, and defining the classifier weight as, \( \psi_i = \log \left( \frac{p_i}{1-p_i} \right), \ 0 < p_i < 1 \), the above equation changes to Eq. 4.6.

\[ \log (Pr(\psi_k | s)) \propto \log (Pr(\psi_k)) + \sum_{i \in |I_+^k|} \psi_i + |I_+^k| \times \log (c-1) \] (4.6)

Our proposed methodology is explained in Figure 4.1. Initially we applied all four segmentation algorithms to the images: Bays-graphcut, binary-graphcut, fuzzyLSM and SFCM2D. Now we have segmented maps. Ensemble results are created using all four algorithms. It makes a decision on the basis of weighted majority voting. The weights are assigned to the members of the algorithm proportional to accuracies of the members.
Images are segmented using ensemble; then Naïve and Kappa accuracies are computed both overall and class wise.

4.2.1 Experimental results

This section shows experimental results of applying an ensemble segmentation method on human colon cancer microscopy images [95, 96] and synthetic images [97, 98]. We perform our experiments on a collection of 50 images of two datasets of Broad Bioimage Benchmark images. These dataset and ground truths are available in this address, http://www.broadinstitute.org/bbbc/.

4.2.1.1 Dataset description

1) Dataset 1 consists of a large number of HCS simulated images which were generated with the SIMCEP simulating software [97]. Each image is 696 x 520 pixels in 8-bit TIF format. Their nuclei and cell areas were matched to the average nuclei and cell areas from the BBBC006 Human U2OS cells image set. These simulated images have a given cell count with a 25% clustering probability and a CCD noise variance of 0.0001. Focus blur is also simulated by applying Gaussian filters to the images. We tested the ensemble model on 26 images of dataset 1 for in-focus images (w1) to denote Hoechst images (shown in Figure 4.2(a) and out-focus images (w2) to denote phalloidin synthetic images (shown in Figure 4.2(b)) for foreground segmentation.

2) Dataset 2 includes human HT29 colon cancer cells images with the size of 512*512 pixels for an image. These fluorescent images are the main data which facilitate any spatial and temporal measurement of fluorescent molecules, existing in a tissue, cell, or the whole body of human. This is composed of two different channels. For the first channel samples were stained with Hoechst in order to label DNA in the nucleus (shown in Figure 4.2(c)) and for the third channel Phalloidin used to stain the actin, which is present in the cytoplasm (shown in Figure 4.2(d)). These images show human HT29 colon cancer cells, a cell line that has been broadly employed for the study of many normal and neoplastic processes. We tested the ensemble model on 24 images of dataset 2 in 1 and 3 channels for fore-ground segmentation.
4.2.1.2 Evaluation and measures

The segmentation accuracies are computed using both overall Naïve and Kappa statistics [99]. Table 4.1 and Table 4.2 respectively represent the formula and description used for Naïve and Kappa statistical evaluation methods both in overall and class wise format. In overall Naïve accuracy, actual places on the ground truth are compared to the same place on the map. The Kappa analysis is a discrete multivariate technique offered for accuracy assessment. The Kappa calculation is based on the difference between how much agreement is actually present between the map of fluorescent images and their ground truth (observed agreement) compared to how much agreement would be expected to be present by chance alone (expected agreement). In order to represent accuracies of individual category, we also used producer’s and user’s accuracies instead of just using overall accuracy that only show the accuracy of overall segmentation. User’s accuracy corresponds to error of commission (inclusion) and Producer’s accuracy corresponds to error of omission (exclusion).
Table 4.1.
Naïve statistics computed from error matrix

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_{ij}$</td>
<td>Number of observations in row $i$ and column $j$</td>
<td>As observed</td>
</tr>
<tr>
<td>$n_{i+}$</td>
<td>Marginal sum of row $i$</td>
<td>$\sum_{j=1}^{k} n_{ij}$</td>
</tr>
<tr>
<td>$n_{+j}$</td>
<td>Marginal sum of columns $j$</td>
<td>$\sum_{i=1}^{h} n_{ij}$</td>
</tr>
<tr>
<td>$n$</td>
<td>Total number of observation</td>
<td>$\sum_{i=1}^{h} \sum_{j=1}^{k} n_{ij}$</td>
</tr>
<tr>
<td>$A_o$</td>
<td>Overall accuracy</td>
<td>$\frac{\sum n_{ii}}{n}$</td>
</tr>
<tr>
<td>$A_o$</td>
<td>Overall error</td>
<td>$1 - A_o$</td>
</tr>
<tr>
<td>$C_i$</td>
<td>User’s accuracy</td>
<td>$\frac{n_{ii}}{n_{i+}}$</td>
</tr>
<tr>
<td>$C_i$</td>
<td>Commission error</td>
<td>$1 - C_i$</td>
</tr>
<tr>
<td>$s_{i+}$</td>
<td>Standard error of user’s accuracy</td>
<td>$\sqrt{C_i(1-C_i)}/n_{i+}$</td>
</tr>
<tr>
<td>$O_j$</td>
<td>Producer’s accuracy</td>
<td>$\frac{n_{jj}}{n_{+j}}$</td>
</tr>
<tr>
<td>$O_j$</td>
<td>Omission error</td>
<td>$1 - O_j$</td>
</tr>
<tr>
<td>$s_{+j}$</td>
<td>Standard error of user’s accuracy</td>
<td>$\sqrt{O_j(1-O_j)}/n_{+j}$</td>
</tr>
</tbody>
</table>
Table 4.2.
Kappa statistics computed from error matrix

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \theta_1 )</td>
<td>Intermediate coefficient</td>
<td>( \theta_1 = \frac{1}{n} \sum_{i=1}^{k} n_{ii} )</td>
</tr>
<tr>
<td>( \theta_2 )</td>
<td>Intermediate coefficient</td>
<td>( \theta_2 = \frac{1}{n^2} \sum_{i=1}^{k} n_{ii} n_{i+} )</td>
</tr>
<tr>
<td>( \theta_3 )</td>
<td>Intermediate coefficient</td>
<td>( \theta_3 = \frac{1}{n^2} \sum_{i=1}^{k} n_{i+} n_{+i} )</td>
</tr>
<tr>
<td>( \theta_4 )</td>
<td>Intermediate coefficient</td>
<td>( \theta_4 = \frac{1}{n^3} \sum_{i=1}^{k} \sum_{j=1}^{k} n_{ij}(n_{ii} + n_{jj}) )</td>
</tr>
<tr>
<td>( \hat{K} )</td>
<td>Overall Kappa</td>
<td>( \frac{\theta_1 - \theta_2}{1 - \theta_2} )</td>
</tr>
<tr>
<td>( \sigma_{\hat{K}} )</td>
<td>Standard error of overall Kappa</td>
<td>( \sqrt{\frac{(\theta(1 - \theta_{12}))(1 - \theta_2)^2}{n}} + \frac{2(1 - \theta_{12})(2\theta_2 - \theta_3)}{(1 - \theta_2)^3} + \frac{(1 - \theta_2)^2(\theta_3 - 4\theta_2^2)}{(1 - \theta_2)^3} )</td>
</tr>
<tr>
<td>( \hat{K}_{ui} )</td>
<td>Kappa user’s accuracy</td>
<td>( \frac{n_{ii} n_{+} - n_{i+} n_{+i}}{n n_{i+} n_{+i}} )</td>
</tr>
<tr>
<td>( \hat{K}_{ij} )</td>
<td>Kappa producer’s accuracy</td>
<td>( \frac{n_{ij} n_{+} - n_{i+} n_{+j}}{n n_{i+} n_{+j}} )</td>
</tr>
<tr>
<td>( \sigma_{\hat{K}_{ui}} )</td>
<td>Standard error of Kappa user’s accuracy</td>
<td>( \sqrt{\frac{p_{ii} - p_{ii}}{n} \left(1 - p_{ii}\right)^2} + \frac{\left[(p_{ii} - p_{ii})(p_{ii} - p_{ii} - p_{ii}) + p_{ii}\right]}{(1 - p_{ii} - p_{ii} + p_{ii})} )</td>
</tr>
<tr>
<td>( \sigma_{\hat{K}_{ij}} )</td>
<td>Standard error of Kappa producer’s accuracy</td>
<td>( \sqrt{\frac{p_{ij} - p_{ij}}{n} \left(1 - p_{ij}\right)^2} + \frac{\left[(p_{ij} - p_{ij})(p_{ij} - p_{ij} - p_{ij}) + p_{ij}\right]}{(1 - p_{ij} - p_{ij} + p_{ij})} )</td>
</tr>
</tbody>
</table>
### 4.2.1.3 Dataset 1 results

Figure 4.3. Segmentation results of different methods on a randomly selected image from dataset 1. From left to right: original image, ground truth, Bayes graph-cut, binary graph-cut, fuzzy LSM, SFCM and Ensemble method.

<table>
<thead>
<tr>
<th>Segmentation Algorithms</th>
<th>Naive Overall Accuracy</th>
<th>Naive Overall Error</th>
<th>Kappa Accuracy</th>
<th>Kappa Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bayesgraphcut</td>
<td>95.84</td>
<td>4.15</td>
<td>73.13</td>
<td>0.20</td>
</tr>
<tr>
<td>Binarygraphcut</td>
<td>99.12</td>
<td>0.87</td>
<td>95.10</td>
<td>0.08</td>
</tr>
<tr>
<td>fuzzyLSM</td>
<td>99.06</td>
<td>0.93</td>
<td>94.64</td>
<td>0.09</td>
</tr>
<tr>
<td>SFCM</td>
<td>99.11</td>
<td>0.88</td>
<td>95.04</td>
<td>0.08</td>
</tr>
<tr>
<td>Weighted Ensemble</td>
<td>99.24</td>
<td>0.75</td>
<td>95.74</td>
<td>0.08</td>
</tr>
</tbody>
</table>

Table 4.3.
Average of naive and kappa accuracy and error per each algorithm and ensemble – dataset 1-W(1) (%)

<table>
<thead>
<tr>
<th>Segmentation Algorithms</th>
<th>Naive Overall Accuracy</th>
<th>Naive Overall Error</th>
<th>Kappa Accuracy</th>
<th>Kappa Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bayesgraphcut</td>
<td>87.70</td>
<td>12.29</td>
<td>70.43</td>
<td>0.12</td>
</tr>
<tr>
<td>Binarygraphcut</td>
<td>98.78</td>
<td>1.21</td>
<td>97.30</td>
<td>0.04</td>
</tr>
<tr>
<td>fuzzyLSM</td>
<td>99.04</td>
<td>0.95</td>
<td>97.88</td>
<td>0.03</td>
</tr>
<tr>
<td>SFCM</td>
<td>99.16</td>
<td>0.83</td>
<td>98.15</td>
<td>0.03</td>
</tr>
<tr>
<td>Weighted Ensemble</td>
<td>99.19</td>
<td>0.80</td>
<td>98.20</td>
<td>0.03</td>
</tr>
</tbody>
</table>

Table 4.4.
Average of naive and kappa accuracy and error per each algorithm and ensemble – dataset 1-W(2) (%)

<table>
<thead>
<tr>
<th>Segmentation Algorithms</th>
<th>Naive User Foreground</th>
<th>Naive User Background</th>
<th>Naive Producer Foreground</th>
<th>Naive Producer Background</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bayesgraphcut</td>
<td>65.27</td>
<td>99.25</td>
<td>90.96</td>
<td>96.25</td>
</tr>
<tr>
<td>Binarygraphcut</td>
<td>94.58</td>
<td>99.63</td>
<td>96.83</td>
<td>99.39</td>
</tr>
<tr>
<td>fuzzyLSM</td>
<td>92.35</td>
<td>99.80</td>
<td>98.17</td>
<td>99.15</td>
</tr>
<tr>
<td>SFCM</td>
<td>94.30</td>
<td>99.65</td>
<td>96.81</td>
<td>99.36</td>
</tr>
<tr>
<td>Weighted Ensemble</td>
<td>94.58</td>
<td>99.76</td>
<td>97.80</td>
<td>99.39</td>
</tr>
</tbody>
</table>

Table 4.5.
Average of naive user and naive producer accuracy per each algorithm and ensemble – dataset 1-W(1) (%)
Table 4.6.
Average of naive user and naive producer accuracy per each algorithm and ensemble - dataset 1-W(2) (%)

<table>
<thead>
<tr>
<th>Segmentation Algorithms</th>
<th>Naïve User</th>
<th></th>
<th>Naïve Producer</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Foreground</td>
<td>Background</td>
<td>Foreground</td>
<td>Background</td>
</tr>
<tr>
<td>Bayesgraphcut</td>
<td>66.14</td>
<td>99.16</td>
<td>97.65</td>
<td>84.81</td>
</tr>
<tr>
<td>Binarygraphcut</td>
<td>98.06</td>
<td>99.15</td>
<td>98.41</td>
<td>98.97</td>
</tr>
<tr>
<td>fuzzyLSM</td>
<td>98.56</td>
<td>99.40</td>
<td>98.87</td>
<td>99.13</td>
</tr>
<tr>
<td>SFCM</td>
<td>98.51</td>
<td>99.51</td>
<td>99.07</td>
<td>99.21</td>
</tr>
<tr>
<td>Weighted Ensemble</td>
<td>98.40</td>
<td>99.60</td>
<td>99.25</td>
<td>99.15</td>
</tr>
</tbody>
</table>

Figure 4.4. Different boundary recognition in two models of segmentation (a) Ground Truth (b) Weighted ensemble (c) Segmentation Binary graph-cut segmentation
In this section, the performance of our weighted ensemble method on dataset 1 is compared with the Bayes graph-cut, binary graph-cut, fuzzy LSM and SFCM (object/background) segmentation methods. We take into account different evaluation measures to calculate the accuracy and precision of the results on dataset 1 for two groups of in-(w1) and out-(w2) focus images. Figure 4.3 shows one of the original images and the ground truth from channel w1 along with the segmented images obtained using all the segmentation methods including ensemble from left to right. The results represent that ensemble method performance is higher than the other methods. Although the performance of binary graph-cut is close to the ensemble, the ensemble result is still better because of the smooth boundary of each recognized cell. The objects and boundaries are obtained with better accuracy with the weighted ensemble model (shown in Figure 4.4). Bayes graph-cut is clearly not performing well which can be seen in the Figure when we compare the result with the ground truth. However, it is difficult to infer based on the visual interpretation which algorithm is better. To have a better comparison in terms of accuracy and error we need to look for numeric comparison.

An average of Naïve and Kappa accuracy for each algorithm in addition to their error has been shown in Tables 4.3, 4.4 for channels w1 and w2 respectively. When we look at the Naïve overall accuracy, except Bayes graph-cut, every algorithm seems to perform similarly, but while looking into the Kappa overall accuracy we see that SFCM and weighted ensemble are higher than the other methods. In overall, it is easy to find that the accuracy of binary graph-cut, fuzzy LSM and SFCM methods are very similar to the weighted ensemble model. If we look at the overall Naïve errors for both focuses in w1 and w2, weighted ensemble has the less Naïve error. One common observation in both focuses is that weighted ensemble is performing better than other algorithms both in terms of overall Naïve and Kappa accuracies and errors.

An average of Naïve user and producer accuracy for each algorithm has been shown in Tables 4.5 and 4.6. Although binary graph-cut, fuzzy LSM and SFCM perform similar to the weighted ensemble method for foreground and background, the results of these Tables show better and acceptable accuracy for weighted ensemble method in foreground and background. From the Tables 4.5 and 4.6 we can observe that both user's and producer's accuracies for background are higher than that of the foreground in both the w1, w2 focuses and for all the segmentation algorithms. It means extracting the exact foreground from the image is relatively difficult for all algorithms.
According to the Tables 4.3, 4.4, 4.5, 4.6, for both w1 and w2, it is observed that Bayes graph-cut has the worst performance amongst all the segmentation algorithms. As those Tables demonstrate, robustness of ensemble in terms of higher accuracy and lower error is higher in both groups of w1 and w2. Also one interesting observation is that in channel w2 segmentation accuracies are higher than that of in channel w1.

Figure 4.5(a)-(d) and Figure 4.6(a)-(d) respectively depict a box plot of Naïve and Kappa accuracies and errors in both focuses of w1 and w2 for all the implemented algorithms per image. The results of ensemble for each image show how similar the results of accuracy are to each other. In other words, for a range of images in different focus in-out the result of ensemble is highly consistent; it is the same case for error.

Figure 4.7 shows a comparison between the average of Naïve user and Kappa user accuracy for each algorithm for foreground and background. Figure 4.8 shows a comparison between average of Naïve producer and Kappa producer accuracy for each algorithm for foreground and background. It is very important to analyze that whether the segmentation error is evenly distributed between classes (background and foreground) or if one of them is really bad and other is really good. Therefore, we include class wise accuracies (User’s accuracy and Producer’s accuracy).

In this experiment we worked with a clean (without noise) image dataset to apply the ensemble method and the rest algorithms. Results revealed that the performance of the proposed algorithm is robust, because it is consistently better than the rest, regardless of any channel. In addition to higher performance the proposed method is able to separate congested cells more accurately which also motivated us to propose our model based on the weighted ensemble. The ensemble method takes all the benefits of Bayes graph-cut, binary graph-cut, fuzzy LSM and SFCM segmentation models and shows stronger results in high performance and low error for each image and average of images. The first step in Bayes and binary graph-cut models is specifying some pre-defined points by the user. We tested our Bayes and binary graph-cut models with different number of foreground and background seed points which were interactively selected by a human user. We selected 50 seed points for foreground and 30 seed points for background. The total results for the ensemble method shows that overall error decreased and overall accuracy increased. Since we are dealing with cancer cell segmentation, results are expected to be high in accuracy, precise and robustness. Even if there is a small improvement in the accuracy for such an application it is important. Due to the work
with real data set of cell cancer images, improving even 0.1 percent in accuracy is beneficial.

Figure 4.5. Box plots of (a) Overall Naïve accuracy (w1), (b) Overall Naïve accuracy (w2), (c) Overall Kappa accuracy (w1), (d) Overall Kappa accuracy (w2) for dataset1
Figure 4.6. Box plots of (a) Overall Naïve error (w1), (b) Overall Naïve error (w2), (c) Overall Kappa error (w1), (d) Overall Kappa error (w2) for dataset1
Figure 4.7. Average Naïve and Kappa user accuracies for foreground and background for four segmentation methods and ensemble (dataset1)

Figure 4.8. Average Naïve and Kappa producer accuracies for foreground and background for four segmentation methods and ensemble (dataset1)
4.2.1.4 Dataset 2 results

Figure 4.9. Segmentation results of different methods on a randomly picked image from dataset 2. From left to right: original image, ground truth, Bayes graph-cut, binary graph-cut, fuzzy LSM, SFCM and Ensemble method.

Table 4.7.
Average of naive and kappa accuracy and error per each algorithm and ensemble – dataset 2 - C (1) (%)

<table>
<thead>
<tr>
<th>Segmentation Algorithms</th>
<th>Naive Overall Accuracy</th>
<th>Naive Overall Error</th>
<th>Kappa Accuracy</th>
<th>Kappa Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bayesgraphcut</td>
<td>94.05</td>
<td>5.94</td>
<td>5.72</td>
<td>0.22</td>
</tr>
<tr>
<td>Binarygraphcut</td>
<td>98.58</td>
<td>1.41</td>
<td>86.18</td>
<td>0.23</td>
</tr>
<tr>
<td>fuzzyLSM</td>
<td>94.86</td>
<td>3.13</td>
<td>64.30</td>
<td>0.37</td>
</tr>
<tr>
<td>SFCM</td>
<td>95.01</td>
<td>1.98</td>
<td>77.58</td>
<td>0.29</td>
</tr>
<tr>
<td>Weighted Ensemble</td>
<td>98.58</td>
<td>1.41</td>
<td>86.18</td>
<td>0.29</td>
</tr>
</tbody>
</table>

Table 4.8.
Average of naive and kappa accuracy and error per each algorithm and ensemble – dataset 2 - C (3) (%)

<table>
<thead>
<tr>
<th>Segmentation Algorithms</th>
<th>Naive Overall Accuracy</th>
<th>Naive Overall Error</th>
<th>Kappa Accuracy</th>
<th>Kappa Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bayesgraphcut</td>
<td>96.33</td>
<td>3.66</td>
<td>90.13</td>
<td>0.08</td>
</tr>
<tr>
<td>Binarygraphcut</td>
<td>97.34</td>
<td>2.65</td>
<td>93.44</td>
<td>0.07</td>
</tr>
<tr>
<td>fuzzyLSM</td>
<td>75.74</td>
<td>24.25</td>
<td>26.28</td>
<td>0.18</td>
</tr>
<tr>
<td>SFCM</td>
<td>80.55</td>
<td>19.44</td>
<td>41.91</td>
<td>0.17</td>
</tr>
<tr>
<td>Weighted Ensemble</td>
<td>98.36</td>
<td>1.63</td>
<td>95.85</td>
<td>0.17</td>
</tr>
</tbody>
</table>
In this section, the performance of our weighted ensemble method on dataset 2 is compared with the Bayes graph-cut, binary graph-cut, fuzzy LSM and SFCM (object/background) segmentation methods. Similar to dataset 1, we take into account different evaluation measures to calculate the accuracy and precision of the results on dataset 2 for two channels (C1) and (C2) of images. Figure 4.9 shows the segmentation results of the four mentioned algorithms and ensemble for a randomly selected image from left to right. The results represent that ensemble method performance is higher than the other methods. In this dataset also the performance of binary graph-cut is close to the ensemble and the ensemble result is still better. In this dataset, Fuzzy LSM and SFCM are clearly not performing well for channel 3 which can be seen in the Figure 4.9 when we compare the result with the ground truth. However, it is difficult to infer based on the visual interpretation which algorithm is better. Again to have a better comparison in terms of accuracy and error we need to look for numeric comparison.

An average of Naïve and Kappa accuracy for each algorithm in addition to their error has been shown in Tables 4.7 and 4.8 for channels C1 and C3 respectively. The results represent the same performance in higher accuracy and lower error for the ensemble
method and binary graph-cut in channel 1. But for the third channel weighted ensemble accuracy and performance show the best results. Binary graph-cut and Bayes graph-cut performance has the second rank in this channel and fuzzy LSM and SFCM ranked among the lowest performance and results.

In channel 1, when we look at the Naïve overall accuracy, except Bayes graph-cut, every algorithm seems to perform similarly, but while looking into the Kappa overall accuracy we see that binary graph-cut and weighted ensemble are higher than the other methods. If we look at the overall Naïve errors for both channels in C1 and C3, weighted ensemble has the less Naïve error. However Bayes graph-cut and binary graph-cut have the less Kappa error for C1 and C3 respectively. In this dataset also, one common observation in both C1 and C3 channels is that the weighted ensemble is ranked among the best algorithm both in terms of overall Naïve and Kappa accuracies and errors.

An average of Naïve user and producer accuracy for each algorithm has been shown in Tables 4.9 and 4.10. For Naïve user accuracy in two channels, weighted ensemble and binary graph-cut performance for foreground is higher than the others. But all the algorithms perform well for background. Although binary graph-cut, SFCM and ensemble perform similar for foreground for Naïve producer, the results of these Tables show better and acceptable accuracy for Bayes graph-cut model. From those Tables, we can observe that almost both user's and producer's accuracies for background are almost higher than that of the foreground in both the C1, C3 channels and for all the segmentation algorithms. It means the same as dataset 1, extracting the exact foreground from the image is relatively difficult for all algorithms.

In this dataset according to the Tables 4.7, 4.8, 4.9, 4.10, for both C1 and C2 channels, it is observed that fuzzy-LSM and SFCM have the worst performance amongst all the segmentation algorithms.

But again as those Tables demonstrate, robustness of ensemble in terms of higher accuracy and lower error is higher in both groups of C1 and C2.

Figure 4.10(a)-(d) and Figure 4.11(a)-(d) respectively depict a box plot of Naïve and Kappa accuracies and errors in both channels of C1 and C2 for all the implemented algorithms per image. The results of ensemble for each image show how similar the results of accuracy are to each other. Figure 4.12 shows a comparison between the average of Naïve user and Kappa user accuracy for each algorithm for foreground and
background. As the Figure shows the average of background accuracy for all the algorithms is higher than the foreground. The most interesting point in this Figure is the weighted ensemble behavior for the foreground and background rather than the other methods. A fluctuation and instability is observed for each algorithm except ensemble. For example Bayes algorithm shows low user accuracy in foreground but high in background but the weighted ensemble has a highest normal accuracy for foreground and background as well. Figure 4.13 shows a comparison between average of Naïve producer and Kappa producer accuracy for each algorithm for foreground and background. For this Figure there is also a normal distribution of accuracy for ensemble method.

In this experiment we also worked with a clean (without noise) image dataset to apply the ensemble method and the rest algorithms. Results for dataset 2 revealed that the proposed algorithm is robust to any changes in image format and error. The ensemble method takes all the benefits of Bayes graph-cut, binary graph-cut, fuzzy LSM and SFCM segmentation models and shows stronger results in high performance and low error for each image and average of images. In this dataset also, we tested our Bayes and binary graph-cut models with different number of foreground and background seed points which were interactively selected by a human user. We selected 10 seed points for foreground and 5 seed points for background. The total results for the ensemble method shows that overall error decreased and overall accuracy increased.
Figure 4.10. Box plots of (a) Overall Naïve accuracy (c1), (b) Overall Naïve accuracy (c3), (c) Overall Kappa accuracy (c1), (d) Overall Kappa accuracy (c3) for dataset2.
Figure 4.11. Box plots of (a) Overall Naïve error (c1), (b) Overall Naïve error (c3), (c) Overall Kappa error (c1), (d) Overall Kappa error (c3) for dataset2
Figure 4.12. Average Naïve and Kappa user accuracies for foreground and background for four segmentation methods and ensemble (dataset2)

Figure 4.13. Average Naïve and Kappa producer accuracies for foreground and background for four segmentation methods and ensemble (dataset2)
4.3 Discussion and conclusion

We proposed a weighted ensemble approach to fluorescence cell image segmentation (foreground/background) and cancer detection based on Bayes graph-cut, binary graph-cut, fuzzy LSM and SFCM. We applied our proposed model on two real and simulated microscopy datasets with different channels and focuses. In order to evaluate the performance we calculated the accuracy and error of our method. Different statistical measures such as Naïve and Kappa statistical measures were used for both datasets. Also, we compared our proposed method with the state-of-the-art algorithms and compared their performance on datasets of human disorders. Our results show that for bio-cell images with complicated or unclear cells, the proposed ensemble method is able to exhibit superior performance. Results revealed that the proposed algorithm is robust to changes in image focuses and has higher performance than the others regardless of any channel and dataset. The comparison results of datasets 1 and 2 shows even better results for dataset 2 which contains real images with disorders. It means effectiveness, consistency and stability of the ensemble method in a real environment is absolutely high. We performed a hypothesis testing for weighted ensemble method and all other methods. The ensemble was the winner of the hypothesis test with 95 percent confidence interval.

Figure 4.14 depicts the runtime in seconds for four state-of-the-art segmentation methods and the proposed weighted ensemble for datasets 1 and 2, provided the segmentation results are available. As the time diagram shows SFCM algorithm takes the least time among the mentioned four algorithms and fuzzy-LSM consumes the most time. The overall runtime of the ensemble method is a few seconds more than the others due to the dependency of the method to the other algorithms. For Bayes graph-cut and binary graph-cut algorithms which are interactive methods, the time will be increased by size of the cell in order to choose the seed points.

For our future work, we plan to propose a combination work of probabilistic approach with deterministic graph-cut models embedding ensemble methods for cell imaging. Also, we will apply our proposed method to other different noisy bio-cell images and expand our experimental results on various data sets.
Figure 4.14. Runtime in seconds for Bayes graph-cut, binary graph-cut, fuzzy LSM and SFCM segmentation methods and weighted ensemble. The left bar indicates estimation time for dataset 1 and the right bar is the estimation time for dataset 2.
5

GRAPH-BASED IMAGE RETRIEVAL USING FUZZY LINKING HISTOGRAM AND ARTIFICIAL IMMUNE SYSTEM

5.1 Introduction

The large amount and increased size of image data sets, and the huge variety of domains, such as, remote sensing, fashion, crime prevention, publishing, medicine, architecture, etc., has resulted in efficient image searching, browsing and retrieval tools. Many multi-purpose image retrieval systems have been developed to tackle this problem [100-114]. All the proposed systems can be categorized into two frameworks [2]: (1) Text-based image retrieval systems, and (2) Content-based image retrieval (CBIR) systems. Text-based image retrieval requires manual human text annotation of images, and has issues with the accuracy of annotations [2]. Therefore, we will focus only on automated retrieval of images through CBIR. In this thesis, we introduce and propose a (CBIR) framework to automatically analyze scenery images, extract relevant features based on appropriate segmentation and retrieve similar images through a nature-inspired algorithm. The major contribution of this thesis is twofold: First, we propose an accurate and exact feature extraction based on Figure-ground image segmentation. Two state-of-the-art methods of image segmentation are compared for specifying the foreground. One method is deterministic, the graph cut method, and the other is probabilistic, the conditional random field (CRF) [115] method [3]. A combination of texture, colour and fuzzy linking histogram [16] are extracted from the foreground. The main motivation of combining these feature extraction methods is to
take their accuracy advantages and reduce their deficiencies.

Second, we exploit a nature-inspired algorithm, a kind of artificial immune system, named TECNO-Stream [115] for similarity measurement. The method has been inspired from the biological immune systems in order to apply clustering approach on images. The aim of image segmentation is dividing an image into relevant and meaningful parts or regions that have a common certain visual characteristics based on image attributes like grey level, colour, texture, depth or motion. Image segmentation has a vital role in many applications such as object identification and recognition, content based image retrieval, and medical image analysis. Despite the fact that there are many existing algorithms and techniques for image segmentation, in this thesis we focus only on graph partitioning methods for our local feature extraction. Graph based segmentation is almost a new segmentation method for many researchers. Graph based segmentation algorithms can be broadly divided into two types 1- deterministic 2- probabilistic. Algorithms such as max/min cut and normalized cut belong to the deterministic approach, and algorithms such as Markov Random Field and Bayesian Network belong to the probabilistic graphical approach. Energy minimization and inference are the main tasks, in graph cut algorithms and probabilistic models, respectively. Because of dealing with many variables, the energy function of graph-cut algorithms tends to be very complex and often NP-hard, especially for non-convex problem.

We evaluate the proposed CBIR method on 531 images of a couple of horse Weizmann dataset and Microsoft image dataset. Experimental results of two CRF and multilabel graph-cut methods illustrate that our method is able to capture complex multilevel semantic structure of images. It shows how increasing features with fuzzy linking histogram could significantly outperform other features in ranking quality.

5.2 Proposed components for CBIR

5.2.1 Conditional random field for region-based image segmentation

The proposed CRF Model for segmentation is a modified version of CRF algorithm proposed by Zhang et al. [3]. This model is based on a relationship between two regions (super-pixels) to infer the state of the edges between them. The superpixel model provides a solution of inferring regions with reducing computational problem that is the most important issue in graphical models.
Figure 5.1 shows how normalized-cut oversegmentation algorithm resulted in some superpixels. In this Figure, there are some superpixels which are mapped to CRF variables, and the extracted features like colour and texture are the value of any superpixel. Each superpixel (specified with white colour) in the original image has been mapped with one region node (black colour) in the CRF part. The red node x represents the whole image and Yi, Yj represent the labels of regions. The links in the CRF model obtained from the pairwise relationship. It means if there is a link between two adjacent superpixels, a link also will be appeared between two CRF nodes in the model.

![Figure 5.1. An over-segmented image with related superpixels which show the nodes in CRF and the links between them are the connections between the nodes](image)

The CRF model is created based on the relationship between the superpixels. According to Eq. 5.1, the joint probability (posteriori probability distribution) of a particular assignment to the entire region node variables Yi will be shown as a normalized product of a collection of non-negative factors (functions).

\[
P(Y|X) = \frac{1}{\rho} \prod_{i \in V} \psi(Y_i, X_i) \prod_{i \in V} \prod_{j \in N_i} \exp(Y_i Y_j \omega_{ij}(X))
\]  

(5.1)

The first part in Eq. 5.1, \(\psi(Y_i, X_i)\), is the node potential and the second part, \(\exp(Y_i Y_j \omega_{ij}(X))\), is the edge potential. V and Y indicate the set of all region node variables and joint labelling of them respectively. \(N_i\) represents the automatically achieved neighbourhood of the \(i^{th}\) region from the topological relationships. The node potential or unary potential is defined as a logistic function in Eq. 5.2. \(\eta\) is a constant for adjusting the curve of the logistic function.
\[ \psi(y_i, x_i) = \frac{1}{1+\exp(-y_i / \eta)} \]  

(5.2)

\[ \omega_{ij}(X) = |X_i - X_j| \]  

(5.3)

The second part or pairwise potential shows how two neighbours interact with each other. The pairwise potential receives different values when the value of two neighbours are the same or different. \( \omega_{ij}(X) \) in Eq. 5.3 depicts feature vector for two nodes i and j. \( x_i \) and \( x_j \) are feature vectors of node i and j respectively. \( \rho \) in Eq. 5.1 is a normalization constant which gives the distribution a summation of one, over all possible joint configurations of the variables \( Y \). \( \rho \) is defined as Eq. 5.4.

\[ \rho = \sum_Y \exp \left\{ \sum_{i \in V} \left[ \log \psi(y_i, x_i) + \sum_{j \in N_i} y_j \omega_{ij}(x) \right] \right\} \]  

(5.4)

Sum-product loopy belief propagation (LBP) used for calculating Marginal probability, \( P(Y|X) \).

5.2.2 Immune-based clustering

Artificial immune systems (AIS) use the idea from natural immune systems to solve engineering problems and present principles for learning and pattern recognition ([116]). Since nowadays we need to handle large image data sets, some scalable clustering techniques are required. Some clustering techniques assume that clusters are hyperspherical, noise-free, and similar in size and try to solve problems with these restrictions. Although, other robust clustering algorithms have recently been proposed to handle even noisy data sets, another limitation of most clustering algorithms is the assumption which is related to the number of clusters. Artificial immune systems (AIS) use the idea from natural immune systems to solve this kind of problem and indicate effectiveness for learning and pattern recognition. In [6] a similarity measure for CBIR, based on an unsupervised AIS learning approach has been proposed, which is called TECNO-Streams. This approach improves learning abilities, scalability, robustness and automatic scale estimation. Robustness of the proposed immune cell model, with respect to outliers and the ability to evolve with changing data was formally proven in [5].
In comparison with some existing scalable clustering algorithms in areas like, insensitivity to initialization, robustness to noise, time complexity, required number of clusters, handling evolving clusters and robust automatic scale estimation, they proved that Tecno-Streams is more powerful than the other models [5]. In the proposed CBIR model we exploit a Tecno-Streams algorithm in order to image clustering. Due to using Tecno-Streams algorithm for CBIR purpose we call that method Tecno-CBIR in the next sections.

5.2.2.1 Antigen and antibody creation
In Tecno-CBIR, the input image feature vector is considered an antigen. According to Eq. 5.5, a set of antigens is defined as an antigen \( (AG) \) vector.

\[
AG = [Ag_1, Ag_2, ..., Ag_N]^T
\]  

(5.5)

Where \( Ag_j, j = 1, ..., N_A \) \( (N_A = \text{maximum population of Antigens}) \) is the \( j \)th antigen that has \( n \) genes. The activation function between the antibody vector \( Ab_i (i = 1, ..., N_B, N_B = \text{maximum population of antibodies}) \) and \( j \)th antigen image vector \( Ag_j \) is defined as Eq. 5.6.

\[
w_{Ab_iAg_j} = \exp \left(-\frac{d^2_{Ab_iAg_j} + j}{2\sigma^2_{Ab_iAg_j}}\right)
\]  

(5.6)

\( \tau \) is a criterion to show the control time decay rate of the contribution of old antigens (old image vector) in the clusters. Therefore its emphasis is on new images which enter to the immune network. It means only images which have been encountered recently will be preserved in the network. \( d^2_{Ab_iAg_j} \) is the distance (such as Euclidean distance) from the antigen which is the \( j \)th antigen encountered by the immune network to antibody \( Ab_i \). \( \sigma^2_{Ab_iAg_j} \) in Eq. 5.7 is a scale parameter which attempts to control the decay rate of the weights along the spatial dimensions in order to specify the size of an influence zone around a cluster prototype.

\[
\sigma^2_{Ab_iAg_j} = \frac{\exp\left(-\frac{1}{2}\sigma^2_{Ab_iAg_{j-1}} W_{Ab_iAg_{j-1}} + w_{Ab_iAg_j} d^2_{Ab_iAg_j}\right)}{2\exp\left(-\frac{1}{2}W_{Ab_iAg_{j-1}} + w_{Ab_iAg_j}\right)}
\]  

(5.7)
\[ W_{Ab|Ag_{j-1}} = \sum_{j=1}^{j-1} w_{Ab|Ag_j} \]  \hspace{1cm} (5.8)

Eq. 5.8 shows the contribution of previous antigens, \(Ag_1, Ag_2, ..., Ag_{j-1}\) to the antibody \(i\). Eq. 5.9 depicts how the \(i^{th}\) antibody represents a soft influence zone with the size proportional to \(\sigma_{Ab_i,Ag_j}^2\), which can be interpreted as a robust zone of influence by antigen.

\[ IZ_i = \left\{ Ag_j \in AG \mid w_{Ab_i|Ag_j} \geq w_{min} \right\} \]  \hspace{1cm} (5.9)

### 5.2.2.2 Cloning and memory construction in immune system

One important immune mechanism of defense system is to reproduce those cells capable of recognizing and binding with antigens [115, 117, 118]. The cellular reproduction in the immune system is based on cloning (mitosis), i.e. the creation of offspring cells which are copies of their parent cells and are subject to mutations. This proliferation will result in the production of a clone of cells of the same type. To avoid preliminary proliferation of antibodies (D-W-B-Cells), and to encourage a diverse repertoire, new antibodies will not be cloned before they are matured (their age, \(Age_i\), exceeds a lower limit \(Age_{min}\)). Due to the mutations, the cells within a clone are all similar but present slight differences that are capable of recognizing the antigen which triggered the immune response. A selective mechanism guarantees that those offspring cells (in the clone) which better recognize the antigen and elicited the response to antigen, are selected to have long life spans; these cells are named memory cells. Figure 5.2 shows the cloning and memory creation operations achieved using stimulation for an input antigen from at least one of the existing antibodies. Antibody cloning and memory construction are made with respect to their stimulation levels and their age condition. In this regard, the \(Ag_{clone_i}\) is calculated based on Eq. 5.10 and the condition of \(Age_{min} \leq Age_i \leq Age_{max}\) is imposed.

\[ Ag_{clone_i} = K_{clone} \frac{\text{stimulation}_i}{\sum_{r=1}^{N_B} \text{stimulation}_r} \]  \hspace{1cm} (5.10)

In Eq. 5.10, \(\text{stimulation}_r\) and \(\text{stimulation}_i\) show the stimulation level of all the antibodies against an antigen, respectively. \(K_{clone}\) is a constant in this formula. As illustrated in Figure 5.2, when an antigen enters the immune network for the first time,
the network creates a memory of external agents based on cloning the cells with the most stimulated ones which facilitates faster recognition of the same cells when they re-enter the network. Even after the disappearance of the external antigen, antibodies co-stimulate each other, and sustain each other. Thus a memory network of images will be constructed. With memory construction, image retrieval speed will increase.

Figure 5.2. A binding between an external antigen (new image) and an image of the cluster with high affinity. It shows how an image (antibody) in a cluster survive after stimulating by an external antigen. Also, it depicts a memory construction and producing clones of an stimulated image.

It is obvious that the speed factor has an important role in image retrieval. When the D-W-B-cell population size $N_B$ exceeds a prespecified maximum $N_{B_{\text{max}}}$, the D-W-B-cells
are sorted in ascending order of their stimulation levels, and the top $N_B - N_{B\max}$ B-cells (with lowest stimulation) are killed. New B-cells ($Age_i < Age_{\min}$) are compensated to be able to compete with more mature cells in the immune network by temporarily (for the purpose of sorting) scaling their stimulation level to the network’s average stimulation level.

### 5.3 Feature extraction

For the colour feature extraction, the average CIELAB colour and their standard deviations used as the local features $X_i$ for each image foreground. In this case, a 6-length of the feature vector is extracted from each image. For extracting texture features from the grey-scale images, we use the average intensity and 12 Gabor textures as the features for every image. Using a set of Gabor filter banks, the Gabor textures are calculated by filtering the grey-scale images. Also, at the time of segmentation, in order to extract foreground from background, the average magnitude of the filtered image in each superpixel region is used as the Gabor feature. We deploy the Gabor filter banks in three scales and four orientations. Therefore the length of the texture feature vector $X_i$ is 13.

#### 5.3.1 Fuzzy linking histogram

Konstantinidis et al. [16] proposed the fuzzy linking histogram based on L*a*b* colour space and projecting a 3D histogram to one dimension histogram, for the purpose of CBIR. In their method, a combination of the three components could result in all colours and grey levels. In L*a*b* colour space, L* indicates luminance, a* indicates relative greenness-redness and b* represents relative blueness-yellowness. The experiments on the regions of the L*a*b* colour space show that in order to have an efficient CBIR system to work with more functionality, the a* and b* components should be subdivided into five regions. Green, greenish, the middle of the component, reddish and red defined for a*. Blue, bluish, the middle of the component, yellowish and yellow defined for b*, and L* should be subdivided into only three regions: dark dim and bright areas.

In a colour image $L*$ contributes for shades of colours in three types of white, black and grey and it cannot provide any unique colour. Therefore, the $L*$ component receives a
lower weight of contribution with respect to the other two components of the L*a*b*.

Figure 5.3. (a) Membership function of the input variable L* (b) Membership function of the input variable a* (c) Membership function of the input variable b* (d) Membership function of the output variable of fuzzy system
The input fuzzification is accomplished by using triangular-shaped built-in membership functions (MF) for the three input components L*, a* and b* which have been shown as the regions in Figure 5.3. The middle membership function exists both in a* and b* provides the ability to represent black, grey and white in a* and b* [13]. The bins of the histogram in Figure 5.3 (d) are in respect to: (1) black, (2) dark grey, (3) red, (4) brown, (5) yellow, (6) green, (7) blue, (8) cyan, (9) magenta and (10) white. The fuzzy linking of the three components is performed based on the 30 fuzzy rules (Appendix A), which direct to the output of the system.

5.4 Total framework of the proposed CBIR

The first step in this model is extracting foreground from background through multilabel and CRF models separately. We use the multilabel graph-cut method discussed in section 2.2.4 to extract foreground from background in an image.

In order to extract foreground/background with CRF model, we first apply a normalized-cut superpixel algorithm on each image of the dataset, then deploy CRF to make inference of segmentation. Each image inference is based on the relationship between each two superpixels. CRF, as a probabilistic graphical model, provides the structure of this type of model with a feature that provides easy visualization through a graph [3]. Each node in a probabilistic Graphical Model represents a random variable, or group of random variables, and the links show probabilistic relationships between these variables. The whole graph describes the way in which the joint distribution over all of the random variables can be decomposed into a product of factors. In this way we take the efficiency of the CRF inference as a probabilistic model to segment foreground.

The next step is feature extraction from the foreground. Different features have been used in our CBIR model for extracting appropriate colour and texture features from images foreground. Although there have been many much rapid progress in CBIR filed in recent years, the critical challenge in terms of the computational techniques is the necessity of analyzing each image for extracting accurate global and local features, since the differentiation of most categories of images is highly depends on the low-level information, such as its shape, colour and texture. First we used a combination of Gabor texture and Lab colour for feature extraction, but the experimental results showed
inaccurately retrieved images. Combining those features with a 10-length feature vector derived from a fuzzy linking histogram, provided more accurate results. Figure 5.4, shows how we extracted a 10-length feature vector through a 10-bar fuzzy linking histogram for one image. It does not require the highly set parameters and only combines three Lab spaces into one which results in a notably lower computational expense. One can easily notice the dominant colours in the image shown in Figure 5.4. As it is shown in the Figure 5.4, bin 7, 9 and 10 are activated because of the green bushes, blue and white sky and magenta mountains objects.

![Fuzzy linking histogram and an image output](image)

Given a new image to be recognized, our system automatically segments foreground from background and instead of comparing them with a large-scale training database, efficiently discovers the most relevant images by comparing each representor of each cluster. The retrieved images are decided by classifying each image using artificial immune system, Tecno-CBIR.

Figure 5.5 shows the total procedure of our work. In Tecno-CBIR, the input image’s feature vector is considered as an antigen. According to the Figure 5.5, when a query comes to CBIR system, a feature vector of its foreground is considered as an antigen. Populations of antigens which have been trained before by the system and clustered are recognized as an antibody. When a new antigen enters the system, the extracted feature vector will be compared with each antibody cluster representor. The affinity or similarity of antigen to each antibody representor is decided based on the activation measurement (criterion). The highest activation reflects the highest similarity between antigen and antibody and results in the most similar images to be retrieved. Effective memory is another advantage for this model in that it tries to keep only the images with high references. In other words, the system automatically withdraws the old images.
which have not been referred most of the time.

Figure 5.5. A total framework of CBIR system

5.5 Experimental results and analysis

5.5.1 Image segmentation based on graph-cut algorithms and probabilistic graphical models

In this part, in order to show some experimental results of graph-based segmentation methods, some segmentation experiments on the Weizmann (horse) dataset [119, 120] have been performed. The Weizmann horse dataset includes colour images of 328 different side-views of horses in addition to their ground truth masks. Because of the variety of images with different appearances and side views, this is a challenging dataset for segmentation. We compare the unified graphical model algorithm and multi-label graph-cut algorithm [119] on a group of five, 320*213 pixels images of this dataset. Figure 5.6 shows examples of horse images in colour, their binary segmentation results from the unified graphical model algorithm [3] and the graph-cut algorithm. Our results show that for object extraction or foreground/background segmentation, the unified
graphical model has better performance. The segmentation accuracy is computed using
the Dice index as given in Eq. 5.11. In the case of the foreground/background we have

\[ \text{Dice}(R, S) = \frac{1}{n} \sum_{i=1}^{n} \frac{2|R \cap S|}{|R| + |S|} \]  

(5.11)

Here, \( R \) denotes the ground truth, \( S \) denotes the segmented image and \( n \) is the number
of segmented areas. The results of the graph-cut algorithm show that this algorithm does
not have a good performance on images with one object and a complicated background.
In particular, when there is a common colour between the foreground and background,
or in the case of the same intensity, the model will not show good results. The Unified
graphical model algorithm has problems when the background is the same in
appearance as the foreground.

We also performed another experimental test on a group of 5 grey-scale images of the
Weizmann (horse) dataset to show the robustness of these two methods for precise
segmentation in the absence of colour. Figure 5.7 shows examples of horse images in
grey-scale, their binary segmentation results from the unified graphical model algorithm
[3] and the graph-cut algorithm. In this case also, the unified graphical model algorithm
has better performance than the graph-cut algorithm. Nevertheless as it has been shown
in Figure 5.6 and Figure 5.7 there is a huge difference between graph-cut algorithm’s
accuracy results for colour images and graph-cut algorithm’s accuracy results for grey-
scale images. In Figure 5.7 for grey-scale images, the results of the graph-cut approach
still suffer from lower accuracy. As the main basis of the graph-cut approach is based on
the intensity differences, the performance result for images with high difference
between the foreground and background, is much better than the others.

In terms of speed, the speed time is \( O(mn^2 |C|) \) (Y. Boykov & Kolmogorov, 2004) for
graph-cut, where \( m \) and \( n \) are the number of edges and nodes, respectively, and \( |C| \) is
the cost of minimum cut for the graph-cut model. In unified graphical model the time
complexity is dependent on the constructed graph.
Figure 5.6. Comparison between unified graphical model and graph-cut model for the horse dataset - The top row shows original images in colour, the middle row is the results of unified graphical model [3] with a 96.6 percent accuracy and the bottom row is the results of graph-cut model [119] with 47.8 percent accuracy.

Figure 5.7. Comparison between unified graphical model and graph-cut model for the grey-scale horse dataset - The top row shows original images in grey-scale, the middle row is the results of unified graphical model [3] with a 98 percent accuracy and the bottom row is the results of graph-cut model [119] with 91 percent accuracy.

The graph usually consists of 700 to 1500 nodes [3]. For images of the same size, segmentation time complexity for the graph-cut model is usually less than for the
unified graphical model. Table 5.1, depicts segmentation time for each image which has been segmented by graph-cut or graphical models.

<table>
<thead>
<tr>
<th>Segmentation model</th>
<th>Image size</th>
<th>Time of segmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multilabel Graph-cut</td>
<td>320*213</td>
<td>Less than 1.08 second</td>
</tr>
<tr>
<td>Unified Graphical Model</td>
<td>320*213</td>
<td>Less than 30 second</td>
</tr>
</tbody>
</table>

5.5.2 Content-based image retrieval

In our method, the superpixel generation is performed based on the Normalized-cut oversegmentation algorithm. All of the training images and the test images are first oversegmented into 80 segments using the Normalized-cut method. Figure 5.8 shows three examples of oversegmented images of Weizmann dataset.

Figure 5.8. Output of N-cut algorithm on Weizmann dataset

In order to construct our CRF inference model, a ground-truth of each image is needed. Figure 5.9 depicts an image from Weizmann dataset and ground truth.
To assess a Mamdani style inference system and hence creating the fuzzy colour histogram the Mathworks Matlab fuzzy logic toolbox was used. The main reason of selecting fuzzy logic [121-125] was its effectiveness in many applications such as automatic control and image understanding due to its flexibility and tolerance to imprecise data.

In this section, we show several experimental results and comparisons to evaluate the effectiveness and efficiency of our proposed CBIR method on a real world image database. All of the images are from 18 different categories in 160*160 size, with average of 30 images per category. Images in the same category belong to the same semantic concept, such as book, cat, horse and etc. In other words, images from the same category are judged relevant and otherwise irrelevant. In Figure 5.10, we randomly select and show three image samples from three different categories. In this figure, foreground/Background image segmentation through CRF model and retrieval is the aim. Selecting a proper dimension for feature vector is very important in image retrieval. It is main part of a CBIR system, which make it possible to obtain good retrieval performances while not requiring a large amount of storage space and computational devices. Figure 5.10 shows a retrieval example of using fuzzy linking histogram on Weizmann dataset and Microsoft image dataset. The query is a book, airplane and horse respectively from dataset, and the number of correctly retrieved images are ranked within the top 10 images. The most left image in each row is the query image, and the rest images are similar images to the query which are retrieved from dataset. The first row for each query in Figure 5.10 shows a 30–length feature vector (included fuzzy linking histogram features) result of image retrieval for CRF
model on book, airplane and horse categories and the second row depicts the result of a 19–length feature vector on the same categories.

Figure 5.10: Results of our CBIR system through CRF image segmentation model with feature vector containing fuzzy linking histogram (top row) and features without fuzzy linking histogram (bottom row)

As it is shown in the Figure, the results of increasing features by including fuzzy linking histogram are mainly different from the results without including fuzzy linking histogram. In our CBIR model, artificial immune system parameters as a similarity measure are defined as below:
The activation threshold is \( w_{min} = 1.2183 e^{-125} \) and the control of time decay rate is \( = 100 \). The maximum size of population is 531.

There are many measures to evaluate the retrieval results such as precision, recall, \( F \) measure, MAP and NDCG [126]. We used 5.12 and 5.13 formulas as precision and recall measures in this thesis.

\[
\text{Precision} = \frac{N_{RIR}}{N_{RIR} + N_{IRIR}} = \frac{\text{Number of relevant images retrieved}}{\text{Number of relevant images retrieved} + \text{Number of irrelevant images retrieved}} \tag{5.12}
\]

\[
\text{Recall} = \frac{N_{RIR}}{T_{RID}} = \frac{\text{Number of relevant images retrieved}}{\text{Total no. of relevant images in the database}} \tag{5.13}
\]

Where, \( NRIR \) is the number of relevant images retrieved, \( NRIR \) is the number of irrelevant images retrieved. \( TRID \) is the total number of relevant images in the database. In our image retrieval system, we set \( N_{RIR} + N_{IRIR} = 10 \); \( TRID = 30 \) on horse Weizmann dataset and Microsoft image dataset. For each measure, all the query results
are averaged on the datasets to obtain overall performance data.

To see the performance distribution in the whole data set, we plot the retrieval precision at top 10 (@10) returns for all 18 categories for ten iterations, in Figure 5.11. Figure 5.11 depicts a comparison between CRF model and multilabel algorithm which are foreground/background segmentation feature extraction and general feature extraction (from whole image). As can be seen, the mean average precision of each algorithm varies with different categories. We found that CRF algorithm performed better than Multilabel in 9 categories and Multilabel also performed better than CRF model in another 9 categories. But for general feature extraction, the retrieval is different with less precision. Retrieval performance (in terms of precision) with general feature extraction performed worst among three methods. CRF performed better than general in precision for 10 out of 18 categories: airplane, building, bicycle, cat, chair, dog, flower, horse, road and tree. It shows a little more precision for CRF feature extraction compared with general feature extraction.

Multilabel performed better than general in precision for 9 out of 18 categories: airplane, building, chair, cow, dog, duck, humanhead, road, tree. It shows similar precision for Multilabel feature extraction compared with general feature extraction.

Figure 5.12 represents the recall retrieval performance of combining different texture, colour and fuzzy linking histogram features on the similar datasets with precision evaluation. As it is shown, the mean average recall of each algorithm varies with different categories. It is observed that in recall the CRF algorithm performed better than Multilabel in 10 categories. Also for general feature extraction, the retrieval is different. Retrieval performance (in terms of recall) with general feature extraction performed worst among three methods.

Tables 5.2 and 5.3 respectively depict average retrieval precision for top 10 retrievals in the 10th iteration and compartmental results of evaluation of image segmentation for all categories in the horse Weizmann dataset and Microsoft image dataset to evaluate CRF, Multilabel and general feature extraction for retrieval. Table 5.3 shows a big difference between CRF segmentation model and Multilabel. In all categories CRF performed better than Multilabel in Dice, Percent and FN evaluation measures except bicycle and tree classes.

As can be seen in the Figures 5.11, 5.12 and Table 5.2, the performance of CRF model, Multilabel algorithm and general feature extraction are very close in retrieval but CRF
model performed better in precision and recall. However, in general feature extraction we lose a little precision.
Chair

**Precision @10**

**Iteration**

**Cowinggrass**

**Precision @10**

**Iteration**
Precision [@10]

Iteration

Duck

CRF | Multi-label | General

Flower

CRF | Multi-label | General

Iteration
Horse

- CRF
- Multi-label
- General

Humanhead

- CRF
- Multi-label
- General
Figure 5.11: The mean average precision of the three different methods for ten iterations on Microsoft and Weizmann datasets.
Figure 5.12: The recall retrieval performance of combining different texture, colour and fuzzy linking histogram features on Microsoft and Weizmann datasets

Table 5.2
Average retrieval precision for top 10 retrievals in 10th iteration

<table>
<thead>
<tr>
<th>Categories</th>
<th>CRF Mean</th>
<th>CRF STD</th>
<th>Multilabel Mean</th>
<th>Multilabel STD</th>
<th>General Mean</th>
<th>General STD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airplane</td>
<td>0.5467</td>
<td>0.2968</td>
<td>0.5733</td>
<td>0.3886</td>
<td>0.4600</td>
<td>0.3047</td>
</tr>
<tr>
<td>Book</td>
<td>0.7033</td>
<td>0.3102</td>
<td>0.5933</td>
<td>0.2778</td>
<td>0.7233</td>
<td>0.2329</td>
</tr>
<tr>
<td>Building</td>
<td>0.3900</td>
<td>0.2280</td>
<td>0.4233</td>
<td>0.1960</td>
<td>0.3167</td>
<td>0.1262</td>
</tr>
<tr>
<td>Bicycle</td>
<td>0.5100</td>
<td>0.2746</td>
<td>0.5467</td>
<td>0.3014</td>
<td>0.4833</td>
<td>0.2086</td>
</tr>
<tr>
<td>Car</td>
<td>0.6100</td>
<td>0.2964</td>
<td>0.3167</td>
<td>0.1704</td>
<td>0.6233</td>
<td>0.2897</td>
</tr>
<tr>
<td>Cat</td>
<td>0.4667</td>
<td>0.2599</td>
<td>0.3458</td>
<td>0.1865</td>
<td>0.4208</td>
<td>0.3036</td>
</tr>
<tr>
<td>Chair</td>
<td>0.3833</td>
<td>0.2102</td>
<td>0.3633</td>
<td>0.2619</td>
<td>0.3600</td>
<td>0.1380</td>
</tr>
<tr>
<td>Cowinggrass</td>
<td>0.3130</td>
<td>0.1766</td>
<td>0.3435</td>
<td>0.1779</td>
<td>0.5043</td>
<td>0.2915</td>
</tr>
<tr>
<td>Cow</td>
<td>0.3100</td>
<td>0.2006</td>
<td>0.4267</td>
<td>0.2449</td>
<td>0.3567</td>
<td>0.1960</td>
</tr>
<tr>
<td>Dog</td>
<td>0.3467</td>
<td>0.1613</td>
<td>0.3667</td>
<td>0.2139</td>
<td>0.2800</td>
<td>0.1606</td>
</tr>
<tr>
<td>Duck</td>
<td>0.3941</td>
<td>0.1825</td>
<td>0.4324</td>
<td>0.2760</td>
<td>0.3588</td>
<td>0.2105</td>
</tr>
<tr>
<td>Flower</td>
<td>0.5967</td>
<td>0.3327</td>
<td>0.5200</td>
<td>0.3134</td>
<td>0.5700</td>
<td>0.3344</td>
</tr>
<tr>
<td>Horse</td>
<td>0.4333</td>
<td>0.3198</td>
<td>0.3600</td>
<td>0.1773</td>
<td>0.3500</td>
<td>0.2162</td>
</tr>
<tr>
<td>Humanhead</td>
<td>0.3767</td>
<td>0.2359</td>
<td>0.6000</td>
<td>0.4177</td>
<td>0.3967</td>
<td>0.1921</td>
</tr>
<tr>
<td>Notifboard</td>
<td>0.3567</td>
<td>0.2285</td>
<td>0.3933</td>
<td>0.2753</td>
<td>0.4633</td>
<td>0.2606</td>
</tr>
<tr>
<td>Road</td>
<td>0.7900</td>
<td>0.2869</td>
<td>0.6367</td>
<td>0.3449</td>
<td>0.5200</td>
<td>0.2917</td>
</tr>
<tr>
<td>Sheep</td>
<td>0.5433</td>
<td>0.2712</td>
<td>0.5567</td>
<td>0.3181</td>
<td>0.7433</td>
<td>0.3390</td>
</tr>
<tr>
<td>Tree</td>
<td>0.6767</td>
<td>0.3256</td>
<td>0.8400</td>
<td>0.2486</td>
<td>0.5933</td>
<td>0.2766</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>0.485956</strong></td>
<td><strong>0.255428</strong></td>
<td><strong>0.479911</strong></td>
<td><strong>0.266144</strong></td>
<td><strong>0.473544</strong></td>
<td><strong>0.242939</strong></td>
</tr>
</tbody>
</table>
Table 5.3
Compartmental results of evaluation of image segmentation

<table>
<thead>
<tr>
<th>Classes</th>
<th>CRF Dice</th>
<th>CRF Percent</th>
<th>CRF FN</th>
<th>Multilabel Dice</th>
<th>Multilabel Percent</th>
<th>Multilabel FN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airplane</td>
<td>0.9477</td>
<td>10.4414</td>
<td>0.2252</td>
<td>0.6833</td>
<td>60.7802</td>
<td>1.9453</td>
</tr>
<tr>
<td>Book</td>
<td>0.8739</td>
<td>25.2059</td>
<td>0.2066</td>
<td>0.8032</td>
<td>37.1</td>
<td>0.2566</td>
</tr>
<tr>
<td>Building</td>
<td>0.8719</td>
<td>25.6028</td>
<td>0.3867</td>
<td>0.7985</td>
<td>37.8072</td>
<td>0.6643</td>
</tr>
<tr>
<td>Bicycle</td>
<td>0.7959</td>
<td>40.8109</td>
<td>0.5063</td>
<td>0.7971</td>
<td>30.9592</td>
<td>0.6695</td>
</tr>
<tr>
<td>Car</td>
<td>0.8842</td>
<td>23.1427</td>
<td>0.2421</td>
<td>0.8425</td>
<td>29.0802</td>
<td>0.5335</td>
</tr>
<tr>
<td>Cat</td>
<td>0.9093</td>
<td>18.1393</td>
<td>0.3357</td>
<td>0.7845</td>
<td>40.8113</td>
<td>1.3952</td>
</tr>
<tr>
<td>Chair</td>
<td>0.9038</td>
<td>19.2252</td>
<td>0.3396</td>
<td>0.7360</td>
<td>49.3507</td>
<td>1.2853</td>
</tr>
<tr>
<td>Cowingrass</td>
<td>0.8848</td>
<td>23.0251</td>
<td>0.4462</td>
<td>0.7840</td>
<td>40.4955</td>
<td>1.6813</td>
</tr>
<tr>
<td>Cow</td>
<td>0.8744</td>
<td>25.1062</td>
<td>0.5158</td>
<td>0.7579</td>
<td>45.5634</td>
<td>1.3299</td>
</tr>
<tr>
<td>Dog</td>
<td>0.8904</td>
<td>21.9169</td>
<td>2185.016</td>
<td>0.7063</td>
<td>56.2764</td>
<td>3.5801</td>
</tr>
<tr>
<td>Duck</td>
<td>0.9329</td>
<td>13.4018</td>
<td>0.4871</td>
<td>0.7520</td>
<td>47.4416</td>
<td>3.5393</td>
</tr>
<tr>
<td>Flower</td>
<td>0.8759</td>
<td>24.8110</td>
<td>1.6631</td>
<td>0.8415</td>
<td>28.4859</td>
<td>0.9095</td>
</tr>
<tr>
<td>Horse</td>
<td>0.6724</td>
<td>38.8058</td>
<td>18.4058</td>
<td>0.5383</td>
<td>65.6246</td>
<td>26.8387</td>
</tr>
<tr>
<td>Humanhead</td>
<td>0.8632</td>
<td>27.3518</td>
<td>1.1865</td>
<td>0.6985</td>
<td>58.4290</td>
<td>1.4401</td>
</tr>
<tr>
<td>Notifboard</td>
<td>0.9131</td>
<td>17.3781</td>
<td>2184.715</td>
<td>0.7705</td>
<td>43.6544</td>
<td>0.8810</td>
</tr>
<tr>
<td>Road</td>
<td>0.7981</td>
<td>40.36146</td>
<td>10961.27</td>
<td>0.6389</td>
<td>71.1317</td>
<td>1.8673</td>
</tr>
<tr>
<td>Sheep</td>
<td>0.8995</td>
<td>20.0924</td>
<td>0.3198</td>
<td>0.8435</td>
<td>28.4428</td>
<td>0.6602</td>
</tr>
<tr>
<td>Tree</td>
<td>0.8808</td>
<td>23.8210</td>
<td>0.2259</td>
<td>0.8975</td>
<td>17.1735</td>
<td>0.2459</td>
</tr>
</tbody>
</table>
5.6 Conclusion

In this thesis, we proposed a novel content-based image retrieval framework based on foreground feature extraction and bio-inspired similarity measure. We tested our model on two datasets. We used CRF model as a state-of-the-art probabilistic image segmentation method and compared this model with a deterministic approach named, multilabel graph-cut algorithm. The results show that the CRF model is more robust and precise in retrieval than multilabel method.

Also, we tested our CBIR framework with a general feature extraction from whole image. The experimental results depict us a lack of precision and recall in retrieval when we replace foreground features with general ones.
6 Conclusion

6.1 Contributions

The main aim of this study is twofold:

(a) **Bio-cell image segmentation:**

1- Proposing a Bayes model of interactive graph-cut that takes all advantages of a Bayes-Gaussian model and graph-cut approach. In experimental results, we compared our proposed method with state-of-the-art algorithms and compared their performance on several images from one popular dataset of human disorders. Our results show that for bio-cell images with complicated or unclear cells, the proposed Bayes based method is able to take into account multiple prior constraints and exhibits superior performance.

2- Proposing a weighted ensemble framework for accurate and robust segmentation of cancer bio-cell images based on four state-of-the-art segmentation methods namely Bayes graph-cut algorithm, binary graph-cut algorithm, spatial fuzzy C-means, and fuzzy Level-Set algorithm. We apply the aforementioned segmentation algorithms on bio-cell images in order to provide an appropriate infrastructure for a weighted ensemble model. Then, the results of different models will be sent to the weighted ensemble algorithm to make a final decision based on the weighted majority. The ensemble based segmentation method takes advantages of all the member methods to improve the segmentation accuracy. Our results show that for bio-cell images with complicated or unclear cells, the proposed ensemble method is able to exhibit superior performance. Results revealed that the proposed algorithm is robust to changes in image focuses and has higher performance than the others regardless of any channel and dataset. The comparison results of datasets 1 and 2 shows even better results for dataset 2 which contains real images with disorders. It means effectiveness, consistency and stability of the ensemble method in a real environment is absolutely high. We performed
a hypothesis testing for weighted ensemble method and all other methods. The ensemble was the winner of the hypothesis test with 95 percent confidence interval.

(b) **Content-based image retrieval:**

First, we proposed an accurate and exact feature extraction based on Figure-ground image segmentation. Two state-of-the-art methods of image segmentation are compared for specifying the foreground. One method is deterministic, the graph cut method, and the other is probabilistic, the conditional random field (CRF) method. A combination of texture, colour and fuzzy linking histogram are extracted from the foreground. The main motivation of combining these feature extraction methods is to take their accuracy advantages and reduce their deficiencies.

Second, we exploit a nature-inspired algorithm, a kind of artificial immune system, named TECNO-Stream for similarity measurement. The method has been inspired from the biological immune systems in order to apply clustering approach on images.

The aim of image segmentation is dividing an image into relevant and meaningful parts or regions that have a common certain visual characteristics based on image attributes like grey level, colour, texture, depth or motion. Image segmentation has a vital role in many applications such as object identification and recognition, content based image retrieval, and medical image analysis. Despite the fact that there are many existing algorithms and techniques for image segmentation, in this thesis we focused only on graph partitioning methods for our local feature extraction.

We tested our model on two datasets. We used CRF model as a state-of-the-art probabilistic image segmentation method and compared this model with a deterministic approach named, multilabel graph-cut algorithm. The results show that the CRF model is more robust and precise in retrieval than multilabel method.

Also, we tested our CBIR framework with a general feature extraction from whole image. The experimental results depict us a lack of precision and recall in retrieval when we replace foreground features with general ones.

### 6.2 Further research

Lack of knowledge is one of the main disadvantages of CBIR systems. Regarding integrating the proposed CBIR system with knowledge-based techniques, a robust CBIR
with low semantic-gap is the aim. We propose an accurate and scalable content-based image retrieval method combined with ontology-based concepts for future work [127-129]. Figure 6.1 represents a schematic of ontology-based system which will be combined with our CBIR Model [129].

In CBIR systems, some knowledge-based techniques are used in addition to relying on extracting primitive features and in order to prevent any semantical gap. The first criterion for having more strong CBIR system in a specified domain is designing ontology for that domain.

Three steps of designing ontology are as below:
- Defining the hierarchy of the domain concepts
- Applying segmentation algorithms and feature extraction methods on image
- Assigning ontology concepts to image objects (image features like colour, texture and spatial relations), Low-level features \(\rightarrow\) High level concepts

Note: Assigning the exact match of an object in an image to a concept is a main problem of mapping. Every different concept even with a small difference should be considered as a new concept. Also only exact feature extraction will be considered as an appropriate feature to be matched with a concept.

Solution for above problem: a degree of membership to a concept (Fuzzy logic approach)

In this way, the concept names will be constructed with information about the degree of membership which results in a concept that we call \(\mu\) – concept. For example \(\mu\) – concept (Tree) \(\geq 0.5\) means an instance of concept of Tree with a degree \(\geq 0.5\). The logical relations between the different \(\mu\) – concepts is defined inside the ontology.

For classifying objects in a new image it has to be segmented into different regions. For every region the spatial relations to its neighbors are determined. When we are trying to retrieve a concept we should attend that one of them might be preferable than the others.

If we want to consider to the spatial relationship between objects, having this sentence is not correct, “water is never above sky”. Instead it is better to have a non-concept description of spatial relationship, “water above sky is non-water”.

Finally the domain dependent knowledge allows us to search for different kinds of images in different categories.
Figure 6.1. An ontology-based system
Bibliography


Appendix A

Thirty rules used to derive fuzzy linking histogram.

1. If (L* is Black) and (a* is Middle) and (b* is Middle) then (Fuzzy_Histogram is black) (1)
2. If (L* is White) and (a* is Middle) and (b* is Middle) then (Fuzzy_Histogram is white) (1)
3. If (a* is Reddish) and (b* is Yellow) then (Fuzzy_Histogram is brown) (1)
4. If (L* is White) and (a* is Green) and (b* is Yellow) then (Fuzzy_Histogram is green) (1)
5. If (L* is White) and (a* is Green) and (b* is Yellowish) then (Fuzzy_Histogram is green) (1)
6. If (L* is Black) and (b* is Blue) then (Fuzzy_Histogram is blue) (1)
7. If (L* is White) and (a* is Green) and (b* is Bluish) then (Fuzzy_Histogram is green) (1)
8. If (a* is Reddish) and (b* is Bluish) then (Fuzzy_Histogram is blue) (1)
9. If (L* is White) and (a* is Reddish) and (b* is Bluish) then (Fuzzy_Histogram is red) (1)
10. If (L* is Black) and (a* is Middle) and (b* is Yellowish) then (Fuzzy_Histogram is yellow) (1)
11. If (L* is Black) and (a* is Reddish) and (b* is Bluish) then (Fuzzy_Histogram is red) (1)
12. If (a* is Reddish) and (b* is Yellow) then (Fuzzy_Histogram is brown) (1)
13. If (L* is Black) and (a* is Middle) and (b* is Black) then (Fuzzy_Histogram is darkgrey) (1)
14. If (L* is Grey) and (a* is Middle) and (b* is Yellowish) then (Fuzzy_Histogram is red) (1)
15. If (L* is Grey) and (a* is Middle) and (b* is Bluish) then (Fuzzy_Histogram is magenta) (1)
16. If (L* is Grey) and (a* is Middle) and (b* is Bluish) then (Fuzzy_Histogram is magenta) (1)
17. If (L* is Grey) and (a* is Middle) and (b* is Bluish) then (Fuzzy_Histogram is cyan) (1)
18. If (L* is Grey) and (a* is Middle) and (b* is Bluish) then (Fuzzy_Histogram is cyan) (1)
19. If (L* is Grey) and (a* is Middle) and (b* is Bluish) then (Fuzzy_Histogram is cyan) (1)
20. If (L* is Grey) and (a* is Middle) and (b* is Bluish) then (Fuzzy_Histogram is cyan) (1)
21. If (L* is Grey) and (a* is Middle) and (b* is Bluish) then (Fuzzy_Histogram is cyan) (1)
22. If (L* is Grey) and (a* is Middle) and (b* is Bluish) then (Fuzzy_Histogram is cyan) (1)
23. If (L* is Grey) and (a* is Middle) and (b* is Bluish) then (Fuzzy_Histogram is cyan) (1)
24. If (L* is Grey) and (a* is Middle) and (b* is Bluish) then (Fuzzy_Histogram is cyan) (1)
25. If (L* is Grey) and (a* is Middle) and (b* is Bluish) then (Fuzzy_Histogram is cyan) (1)
26. If (L* is Grey) and (a* is Middle) and (b* is Bluish) then (Fuzzy_Histogram is cyan) (1)
27. If (L* is Grey) and (a* is Middle) and (b* is Bluish) then (Fuzzy_Histogram is cyan) (1)
28. If (L* is Grey) and (a* is Middle) and (b* is Bluish) then (Fuzzy_Histogram is cyan) (1)
29. If (L* is Grey) and (a* is Middle) and (b* is Bluish) then (Fuzzy_Histogram is cyan) (1)
30. If (L* is Grey) and (a* is Middle) and (b* is Bluish) then (Fuzzy_Histogram is cyan) (1)