Evaluation Methods of Accuracy and Reproducibility for Image Segmentation Algorithms

A Thesis Submitted to the
College of Graduate and Postdoctoral Studies
in Partial Fulfillment of the Requirements
for the degree of Master of Science
in the Department of Computer Science
University of Saskatchewan
Saskatoon

By
Yu Sun

Permission to Use

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
University of Saskatchewan
176 Thorvaldson Building, 110 Science Place
Saskatoon, Saskatchewan S7H 5C9
Canada

Dean
College of Graduate and Postdoctoral Studies
University of Saskatchewan
116 Thorvaldson Building, 110 Science Place
Saskatoon, Saskatchewan S7N 5C9
Canada

Abstract

Segmentation algorithms perform different on differernt datasets. Sometimes we want to learn which segmentation algoirithm is the best for a specific task, therefore we need to rank the performance of segmentation algorithms and determine which one is most suitable to that task.

The performance of segmentation algorithms can be characterized from many aspects, such as accuracy and reproducibility. In many situations, the mean of the accuracies of individual segmentations is regarded as the accuracy of the segmentation algorithm which generated these segmentations. Sometimes a new algorithm is proposed and argued to be best based on mean accuracy of segmentations only, but the distribution of accuracies of segmentations generated by the new segmentation algorithm may not be really better than that of other exist segmentation algorithms. There are some cases where two groups of segmentations have the same mean of accuracies but have different distributions. This indicates that even if the mean accuracies of two group of segmentations are the same, the corresponding segmentations may have different accuracy performances. In addition, the reproducibility of segmentation algorithms are measured by many different metrics. But few works compared the properties of reproducibility measures basing on real segmentation data.

In this thesis, we illustrate how to evaluate and compare the accuracy performances of segmentation algorithms using a distribution-based method, as well as how to use the proposed extensive method to rank multiple segmentation algorithms according to their accuracy performances. Different from the standard method, our extensive method combines the distribution information with the mean accuracy to evaluate, compare, and rank the accuracy performance of segmentation algorithms, instead of using mean accuracy alone. In addition, we used two sets of real segmentation data to demonstrate that generalized Tanimoto coefficient is a superior reproducibility measure which is insensitive to segmentation group size (number of raters), while other popular measures of reproducibility exhibit sensitivity to group size.

Acknowledgements

Firstly, I would like to express my special appreciation and heartfelt thank to my supervisor Professor Mark Eramian. Mark is a tremendous mentor, who always spare no effort to help me whenever I am in trouble. Thanks for encouraging my resreach and for allowing me to grow as a research scientist.

Secondly, I would like to thank the Beijing Institute of Technology and the University of Saskatchewan, who gave me the oppotunity to study in Canada and work with top-class academicians.

Third, I am grateful to my parrents for their understanding and for their support of my study and my life. They gave me the courage to study and live abroad alone.

I thank my fellow lab mates Hao Song, Imran Ahmed, and Rahat Yasir for programing problems and all the fun we have had in the past two years.

I want to say thank you to my roomates Qi Guo, Kang Jiang, and Ning Chang for sharing mathematical and cooking knowledge with me.

Last but not least, I would like to thank all my friends in Canada, with whom I have had a great time here.

Contents

List of Tables

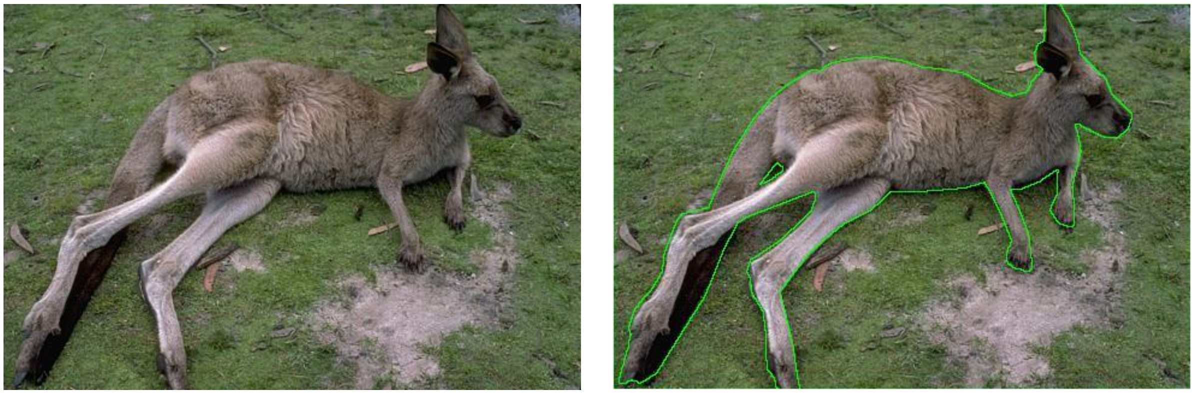
List of Figures

List of Abbreviations

ANN	artificial neural networks
AUC	area under the curve
BSDS500 dataset	Berkeley segmentation dataset and benchmark 500
CA	classification accuracy
CDF	cumulative distribution function
CNN	convolutional neural network
CT	computed tomography
CV	coefficient of variation
DDSM	digital database for screening mammography
DRLSE	distance regularized level set evolution using closed contour interaction
DRLSEIC	distance regularized level set evolution using closed iso-contour interaction
DSC	Dice similarity coefficient
GCBS	graphcut with star shape prior using brush stroke interaction
GCnoSP	graphcut without star shape prior using seed point interaction
GSC	geodesic star convexity using brush stroke interaction
GSCSeq	sequential geodesic star convexity using brush stroke interaction
GTC	generalized Tanimoto coefficient
HD	Hausdorff distance
HOG	histogram of oriented gradient
HOF	human ovarian follicles
ICC	intra-class correlation coefficient
JDC	joint Dice coefficient
JTC	joint Tanimoto coefficient
kNN	k-nearest neighbors
LBP	local binary pattern
MAD	mean absolute distance
MaxD	maximum difference
MS	Morphological Snakes
MRI	magnetic resonance imaging
MSD	mean squared distance
PDF	probability distribution function
RMAD	root mean absolute distance
RMSD	root mean squared distance
RW	Random Walkers
ROI	region of interest
SDMR dataset	shadow detection and texture segmentation dataset for mobile robots
SIFT	scale-invariant feature transform
SOM	self organizing maps
SVM	support vector machines
TC	Tanimoto coefficient
TRC	trust region convexity using brush stroke interaction
UN	U-Net

1. Introduction

Image segmentation is an essential issue in the field of computer vision, which provides a delineation of one or more regions of interest (ROI) of the input image. The goal of image segmentation is to partition the ROI from the given image. An example is shown in Figure 1.1. Figure 1.1(a) is an image in the Berkeley segmentation dataset and benchmark 500 (BSDS500 dataset) [51], where the kangaroo is the ROI. In Figure 1.1(b), the kangaroo is segmented and outlined in green.



(a) An image with a kangaroo as ROI.

(b) The ROI is segmented and outlined.

Figure 1.1: Example images in the BSDS500 dataset.

The quality of segmentations can significantly affect the result of other operations such as image representation and image recognition. Therefore, the segmentation algorithms which have superior performance should be selected to perform the segmentation work. They can be selected by applying some measuring and ranking methodology.

The performance of segmentation algorithms can be evaluated in different ways. The most often used performance metric is the accuracy of segmentations. In most papers, e.g. [21, 26, 32, 33, 41, 52, 53, 55, 60, 61, 66, 68, 71, 85], the performance of segmentation algorithms are characterized by the means of accuracies of the segmentations of individual test images. In [85], the percentage of correctly classified pixels, which is also known as classification accuracy (CA), was calculated to represent the performance of the proposed Bayesian network model. Top et al. [78] used mean Dice similarity coefficient (DSC) to measure the performance of their method. By averaging all the individual accuracy values, such as DSCs, the effect of accidental error in accuracy measure will be reduced. Generally speaking, the higher the mean value of accuracy, the better the performance of an image segmentation algorithm.

Standard deviation of accuracies can provide auxiliary information for ranking different image segmentation algorithms. A lower standard deviation indicates that the values of data points are closer to the mean value of all accuracy measure values. While a higher standard deviation indicates that the values of data points are farther away from the mean value or, in other words, are spread out over a wider range of values. Therefore, the algorithm which has a lower standard deviation is generally preferred when the means are close. In [33, 41, 71, 75], both the mean and the standard deviation of accuracy are used to show the performance of segmentation algorithms of segmentation algorithms.

Sometimes the mean and standard deviation of accuracies of segmentations generated by different algorithms are similar. In such a scenario, some statistical methods, such as hypothesis test, may be adopted to determine which algorithm is the best. For instance, given two sets of accuracy data, the non-parametric Wilconxon rank-sum test (Mann-Whitney U test) can be used to test the hypothesis that the accuracy samples of segmentations generated by two image segmentation algorithms come from the same distribution [32, 35]. If the hypothesis is rejected, it means the accuracy samples are from different distributions. In other word, the two algorithms have different performance of segmentation algorithms at a statistically significant level and the one which has a higher mean value of accuracy is considered superior to the other. If the hypothesis can't be rejected, it means more evidence is needed to make a decision. In this thesis, the method of using mean, standard deviation, and hypothesis test to rank segmentation algorithms will be referred to as the *standard method*.

There are problems when using mean and standard deviation of accuracy to measure the performance of segmentation algorithms. The most significant one is that using the mean and standard deviation makes the implicit assumption that the accuracies of different segmentations are distributed normally. It may be unsuitable to use mean and standard deviation of accuracy to characterize the performance of segmentation algorithms without determining the distribution type of the accuracies of segmentations, especially when the accuracies don't follow normal distribution.

It is easy to imagine two very different distributions that have the same mean and the same standard deviation. Figure 1.2 shows the distributions of two groups of synthetic data. The means and standard deviations of both of these two groups of data are 0.0263 and 0.0152 respectively, but their shapes differ a lot. When using the means and standard deviations of these two groups of data to determine the performance of the corresponding segmentation algorithms, the conclusion would be that these two algorithms have comparable performance of segmentation algorithms, which is obviously incorrect. Therefore, sometimes it may be inappropriate to compare segmentation algorithms just using mean and standard deviation without considering the distribution of the accuracies of their segmentations. In this thesis, a proposed new methodology that

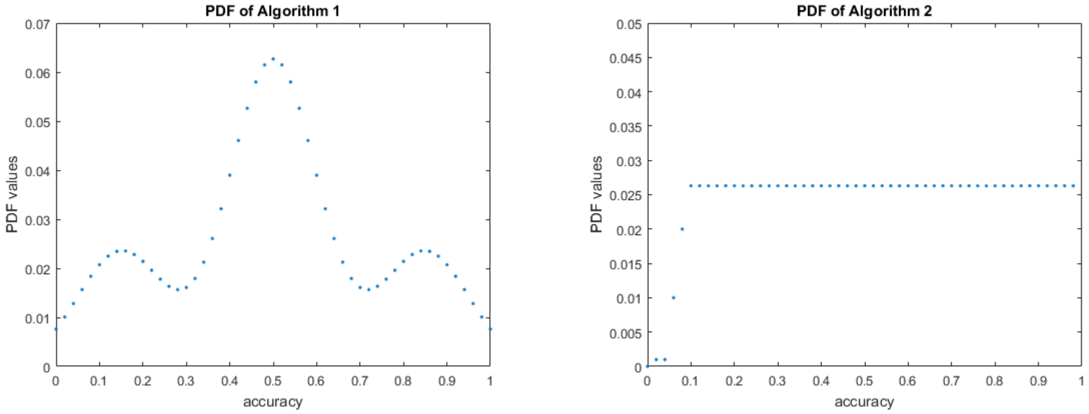


Figure 1.2: Two different distributions have the same mean and the same standard deviation.

can evaluate the accuracy of segmentation algorithm is based on the cumulative distribution function of the segmentation generated by the segmentation algorithm.

Apart from the accuracy, other metrics such as *reproducibility* [18, 21, 26, 53, 55, 59, 61, 66, 68], and *efficiency* [26, 53, 55, 59, 60, 85] can also characterize the performance of segmentation algorithms of segmentation algorithms. Reproducibility is a measure of the mutual similarity of a group of segmentations. It is also referred to as reliability in [19, 63, 86]. Using synthesized data, Eramian demonstrated that generalized Tanimoto coefficient (GTC) is the only measure among joint Dice similarity coefficient, joint Tanimoto coefficient, coefficient of variation and intra-class correlation coefficient that is not significantly affected by group size of segmentations [22]. But it is not known whether these findings remain true for real data. The term of ”*efficiency*” has been used to refer to different characteristics. For example, in [55], McInerney noted that the efficiency of segmentation algorithms can be measured by the total time to perform the segmentations, or the quantity and quality of user interaction. Since the computation time of the same segmentation algorithm may be different on different machines and for different types of interactions, such as making annotations on images and setting parameters of segmentation algorithms, the measure of efficiency of segmentation algorithms is not discussed in this thesis.

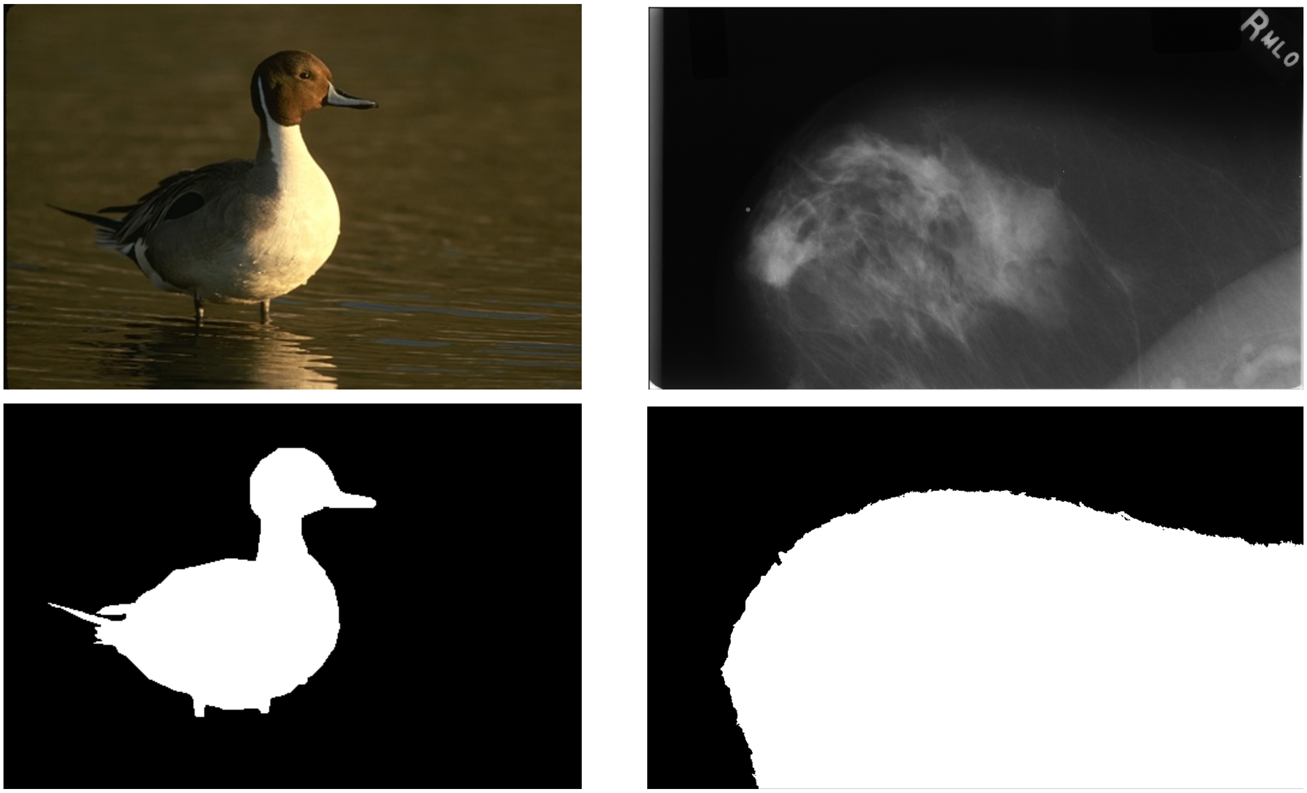
This thesis focuses on how to determine the segmentation algorithm with the best accuracy and reproducibility when more than one segmentation algorithm is available. An extensive method is proposed to characterize the accuracies of segmentation algorithms. As for the reproducibility, the values of several metrics mentioned in [22] are applied to several groups of segmentations in order to verify that the generalized Tanimoto coefficient (GTC) is the best metric to evaluate reproducibility of semi-automatic segmentation algorithms in practice.

The rest of the paper is organized as follow. Chapter 2 introduces the background of image segmentation, which consists of datasets, operating modes, types of segmentation algorithms, and measures for segmentation algorithms. Research objectives are introduced in chapter 3. In chapter 4, the new evaluation method, $\overline{CDF}(\theta)$, is proposed basing on the distribution of accuracies of segmentations. Two examples are used to show its validity by evaluating and comparing the accuracy performance of segmentation algorithms of the Morphological Snakes algorithm [2], the Random Walkers algorithm [25], and the U-Net algorithm [1]. Chapter 5 presents how to develop an extensive method to evaluate and rank the accuracy performance of segmentation algorithms. Chapter 6 verifies the GTC is the best metric to measure the reproducibility of semi-automatic segmentation algorithms using some real segmentation data. The experiment result is compared with Eramian's result which is generated using synthetic data [22]. The contributions of this thesis and future works are discussed in Chapter 7.

2. Background and Literature Review

In order to evaluate segmentation algorithms, three components are necessary: the dataset, which contains the images that need to be segmented, the implementations of image segmentation algorithms, and the methodology of evaluation.

2.1 Dataset



(a) Color image (top) and its ground truth (bottom). (b) Gray scale image (top) and its ground truth (bottom).

Figure 2.1: Examples of original images and their corresponding ground truths.

In general, datasets which can be used to test the performances of segmentation algorithms consist of two kinds of images. First, there are original images, which will be segmented by segmentation algorithms. The original image can be a color image such as images in the BSDS500 dataset [3], or a gray scale image such as images in the digital database for screening mammography (DDSM) [30, 31]. Examples are shown in Figure 2.1. Second, there are ground truths. The ground truths usually binary images. Typically, the black parts of them represent the background region, and the white parts of them represent the foreground region. The ground truth images are usually used to compare with the segmentations to calculate the accuracies of the segmentations.

2.2 Segmentation Algorithms

2.2.1 Operating Modes of Segmentation Algorithms

Segmentation algorithms can be classified into three categories depending on whether there is participation of human users. Algorithms which operate without the guidance of humans are called automatic segmentations. Segmentations are generated without the help of segmentation algorithms and only by human operators are called manual segmentations. In some situations, human users are required to provide expert knowledge or some guidance to initialize or refine segmentations. This is called semi-automatic segmentation or interactive segmentation.

Automatic segmentation is preferable to semi-automatic or manual segmentation, since it doesn't require the assistance of human. So the automatic segmentation is time-saving and efficient in many situations. However, some segmentation problems are still difficult to solve by automatic segmentation algorithms, especially when the number, the size and the shape of foreground regions are variable or the boundary of foreground regions are indistinct.

Manual segmentations are usually used as ground truth images of many datasets to test the performances of segmentation algorithms. The human users who do the manual segmentation work usually have expert knowledge of the segmentation problem. However, manual segmentation is time-consuming and laborious. That's why the manual segmentations are not preferred in practice, except as a means of validating (semi-) automatic segmentation algorithms.

Semi-automatic segmentation is a trade off between the accuracies of segmentations and the time and effort required of human users. Human users provide high-level contextual knowledge to initialize the segmentations and sometimes give some feedbacks to improve the segmentation results. Thus the semi-automatic segmentation can save a lot of time for human operators compared to manual segmentation, while improving the accuracies of segmentations significantly compared with automatic segmentations.

2.2.2 Types of Segmentation Algorithms

Segmentation algorithms can be classified into different groups basing on different classification criteria. In this section, several segmentation algorithms will be introduced according to the type of their theories. In the recent twenty years, the most common image segmentation algorithms are graph-based segmentation algorithms, learning-based segmentation algorithms and functional-based segmentation algorithms.

Graph-Based Segmentation Algorithms

Graph-based segmentation algorithms always adopt graph theory and related knowledge to segment images. In general, the input image is represented as an undirected graph with a pixel for each node and edges representing the relationship between neighboring pixels. Then, some mathematical methods are used to partition the graph and thus segment the image.

Graph theory was first adopted to analyze images in [83] and [84]. However, it didn't draw people's attention until the emergence of the Normalized Cuts [69] algorithm. Normalized Cuts is a typical graph-based algorithm. It maps the input image into a weighted graph, where the nodes of the graph represent the pixels in the input image. The weight on each edge represent the similarity of two pixels, which is related to the locations and intensities of these two pixels. When segmenting the input images, some of the edges in the corresponding graph will be removed, which is called a *cut* in graph theoretic language. In order to find the optimal segmentations, the minimum cut of the graph should be calculated. This can be achieved by calculating the eigenvalues and eigenvectors of a cost function [69]. It is an NP-hard problem to find the minimum of the Normalized Cuts, but approximate solutions are available, such as [12]. However, the approximate solutions can be arbitrarily far from the optimal solution. Because of this, the segmentation results generated by the Normalized Cut algorithm are barely satisfactory.

Graph Cuts [5, 9, 36, 46, 48, 49, 56, 77], a classical graph-based algorithm, is an improvement of Normalized Cuts. Similar to the Normalized Cuts, Graph Cuts map pixels of an image into a weighted graph and employ min-cut methods to divide the images into foreground and background. First, the Graph Cuts algorithm will convert the input image into a undirected graph. Each pixel of the input image is a node of the graph and the connection of two adjacent pixels is an edge of the graph, which are called n-links. There are another two special nodes (node S and node T), which are called terminals in the graph. S is the object terminal and T is the background terminal. Both of these two terminals connect with all other nodes. Second, users need to provide annotations on the input image as hard constraints. In this step, users identify foreground and background of the input image by marking pixels in different ways, such as drawing strokes or seed points on the input images. Third, the cut operations are conducted to partition the nodes in the graph. The min-cut [13] will generate a segmentation that satisfies the hard constraints and at the same time optimize the cost of cuts. Although the Graph Cuts algorithms are widely used in many fields, there are some minor problems. For instance, when the amount of the provided seeds are small, it is very likely that only a small part of the foreground is segmented, which is called the ”small cuts” mis-segmentation problem in [26]. This is because the Graph Cuts algorithm always makes a minimum cut to separate the seeds from the rest of the input image. Therefore, users may have to set seeds on the input image continually to improve the segmentation result and overcome the ”small cuts” problem.

Later, the GrabCuts method was proposed by Rother et al. [64]. The GrabCuts method is developed from the Graph Cuts algorithm. The ”iterative estimation” and ”incomplete labeling” mechanism can reduce the interaction work of users considerably. When using GrabCuts, users just need to draw a box around a foreground region, instead of placing seed points liberally and evenly. Using iterated Graph Cuts, the GrabCuts algorithm can also reduce the blur and mixed background pixels on foreground boundaries. Although the GrabCuts works well when segmenting color images, its iterative nature consumes a lot of computation resource. In addition, the box-based interaction is not efficient sometimes, as the further editing of interaction is required in many situations when users are not satisfied with the segmentation results.

The Random Walkers algorithm [25] is also a well known graph-based segmentation algorithm. It can segment images in the following 3 steps. Similar to Normalized Cuts, the Random Walker algorithm maps the input image to a graph at first. Second, the probability that a random walker starting at a pixel and first reaches a particular seed point label are calculated by solving the linear equations of a Dirichlet problem. Since the random walker has the same solution as the Dirichlet problem [10], the result of the Dirichlet problem can be used to indicate the foreground boundaries. Third, each unlabeled pixel will be labeled according to the higher probability calculated in the second step. Thus the input image can be segmented.

For many graph-based segmentation algorithms, seeds are necessary to initialize the segmentation process or refine the segmentation result. The seeds may be automatically determined or come from users' annotation. Thus, most of semi-automatic segmentation algorithms are graph-based segmentation algorithms.

Learning-Based Segmentation Algorithms

Learning-based algorithms refers to those which use the machine learning theory to perform the segmentation. They employ classifiers, which can classify the pixels or voxels in an image into different groups and therefore segment the input images. Learning-based segmentation algorithms can be classified into two categories depending on whether labeled images are used to train classifiers. The algorithms which can segment images without users providing samples are unsupervised learning-based segmentation algorithms. Others which use labeled images to train their models and segment images are supervised-learning-based segmentation algorithms. When taking an input image, the corresponding label can be used to train a model, which can generate a prediction, or more specific, a segmentation. But in unsupervised learning, no labels are available, which makes the prediction more difficult. Therefore, supervised-learning-based segmentation is preferred when the labeled images are sufficient for training the segmentation model.

The unsupervised learning-based segmentation algorithms refer to those algorithms which applied unsupervised machine learning methods to help segmenting images. The unsupervised neural network and k-means clustering are often-used unsupervised learning-based segmentation algorithms.

Unsupervised neural networks can be used for preprocessing, feature extraction, segmentation etc. [20]. The images that need to be segmented are the same as those used to train the neural network [65]. The input images are firstly used to determine the weights of neurons of the unsupervised neural network according to some specific learning rules. Then the well-trained model is used to classify pixels in the input images into different categories. Hopfield neural networks [34] and Kohonen self-organizing map (SOM) [42], which are also called Kohonen feature maps, are the most often used unsupervised neural networks for image segmentation.

The k-means clustering methods are usually combined with the features of input images, such as scale-invariant feature transform (SIFT), histogram of oriented gradient (HOG), local binary pattern (LBP) and so on, to segment images [4, 16, 74]. Those pixels whose features are close to each other in some measures are gathered and classified into the same category. Different categories will be illustrated as different kinds of regions. If there are only two categories, the input images can be segmented into the foreground and the background. One disadvantage of the segmentation algorithms that applied k-means clustering methods is that the number of clusters should be pre-defined [15]. For example, for semantic segmentation, there may be more than two categories in one image. In such a scenario, the number of the clusters centers should be defined to be the same as the total number of foreground regions and background regions before the segmentation. In addition, it may be difficult to select suitable features for different kinds of images to implement high accurate segmentation [15].

Some algorithms used both the unsupervised neural network and clustering to segment images. For example, in [6], Mohamad Awad used SOM [42] to map the input images from three-dimensional space to two-dimensional space. The new data in the two-dimensional space is used as the input of the proposed T-Cluster technique to determine the cluster centers. In this way, the features in the two-dimensional space will be classified into different groups, which indicate the pixels of the input images belong to foreground regions or background regions.

Traditional machine learning methods are mostly based on statistical methods and can not feedback the learning results to the inputs, which may lead to the fact that in a real-world environment, the unsupervised segmentation algorithms are not very effective and adaptable.

For supervised-learning-based segmentation, the training set which contains images and their labels, namely the corresponding segmentations, is necessary for extracting the common information of the images to be segmented. There are many supervised-learning-based segmentation algorithms, such as k-nearest neighbors (kNN), supervised artificial neural networks (ANN), support vector machines (SVM), and so on [50]. In the following part of this section, the theories of the aforementioned supervised machine learning methods will be introduced.

kNN is easy to tune, since the number of neighbors, i.e. k, is the only parameter. In the training phase of a k-nearest neighbor classifier, the feature vectors and labels of training samples are stored. In the testing phase, the distance between the feature vectors of unlabeled input image and all other samples in feature space will be calculated. The unlabeled points will be classified according to the votes of the k nearest samples' labels. The example application of segmentation algorithm using kNN classifier can be seen in [81].

Different from the kNNs, the ANNs consist of many parameters which are used to characterize the weights of neurons and the complex structures of the networks. Artificial neural network is inspired by biological neural network in human brain, and it is the most often used algorithm for driving deep leaning. A typical example of ANN is the convolutional neural network (CNN). CNN is one of the most often used model for deep learing and it is developed from ANN. For ANN, each neuron can generate an output when applying weights and biases on the input values. As there are usually hundreds of neurons in each layer and several layers in ANN, the number of parameters may be extremely great, which make the training of the ANN to be difficult. For CNN, the vector of weights and biases is called a filter. The shapes of filters are usually squares, such as 3 by 3. The dot product of an input layer and a filter is the output of that input layer, which contains some specific features. The goal of the training of CNN is to learn the values of these filters, so that the each layer can extract some useful information from its corresponding input. The most special feature of CNN is that neurons in the same layer share the same filter, which can reduce the number of parameters and the complexity of the network. Therefore, CNN could be deeper (contains more layers). In the training phase, all samples are used to update these parameters to minimize the loss functions, which indicate the difference between the outputs and the ground truths. Then the well-trained model can be used to classify the pixels in an input image into different categories and thus segment the input image. The advantages of ANN is that all features that used for classification are learned from the training phase, instead of those pre-defined features, such as SIFT, HOG, LBP etc.. However, it takes a long time and requires thousands of labeled data to train the model of ANN. The U-Net [1], an implementation that is used in chapter 4 and chapter 5, is a modification of traditional ANN for automatic segmentation.

The goal of SVM is to find a hyper-plane to separate data points in a high-dimensional feature space [82]. SVM can map data points which are not linearly separable in the original low-dimensional space onto a higher-dimensional feature space (Hilbert space) where the transformed data points are linearly separable. In the training phase, two parallel hyper-planes are contrasted to separate data points into two parts and at the same time the distance of these two hyper-planes are maximized. Between these two hyper-planes, a new hyper-plane, which has the same distance to the existing two parallel hyper-planes, will be constructed to classify the testing data. It is believed that the greater the distance of the parallel hyper-planes, the smaller the overall classification error. Combined with extracted features, the SVM can be used to segment images.

Functional-Based Segmentation Algorithms

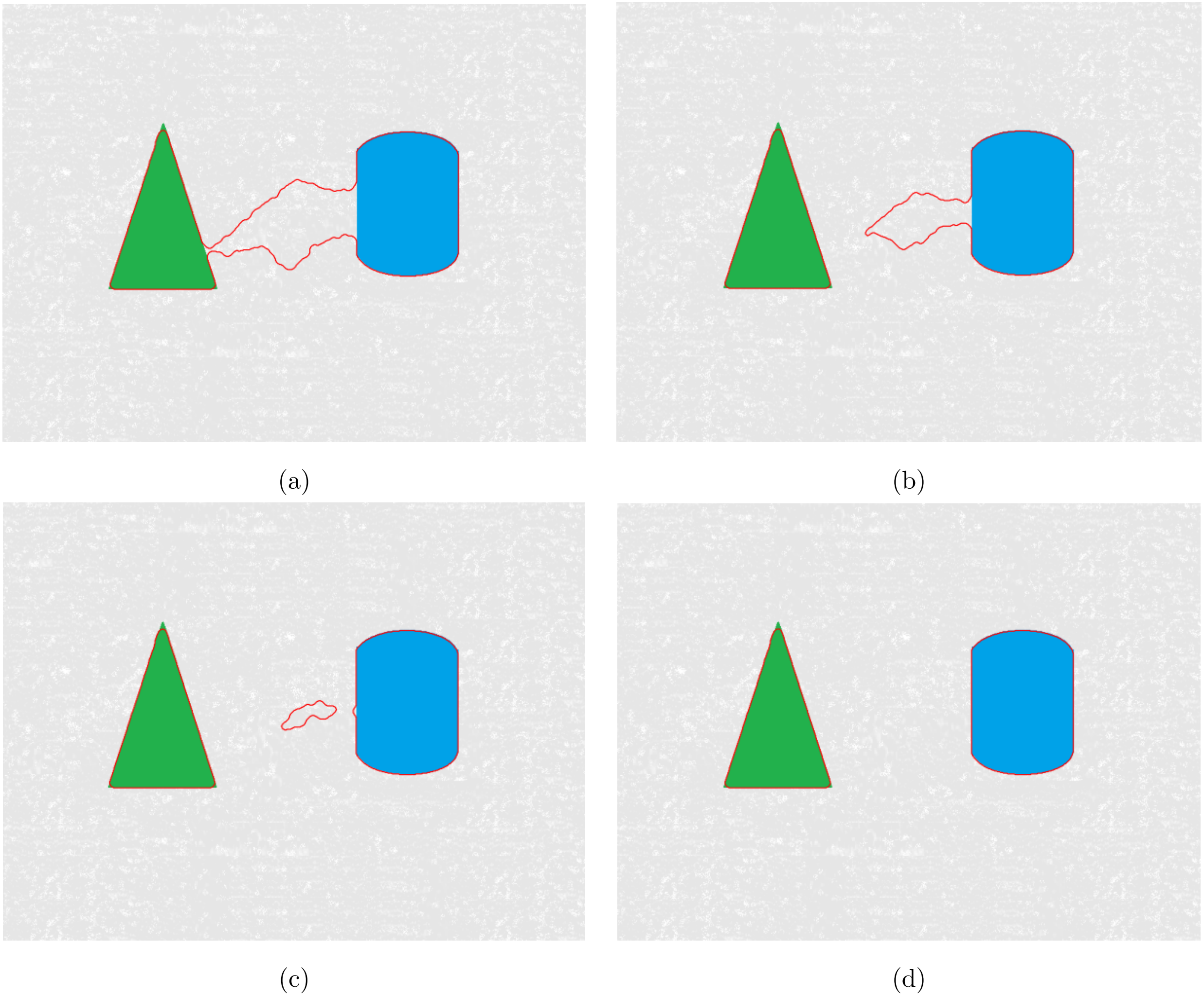
The basic idea of functional-based algorithms is defining an energy function whose independent variables contain the information of the closed boundary curves of the foreground. As the defined energy function has minimum energy along the foreground region's boundary, the segmentation problem will be transformed into the problem of finding the minimum of the defined energy function. This minimum can be obtained by solving the corresponding Euler-Lagrange function. In general, this class of algorithms mainly contains the active contour model and derivative algorithms. It can be classified into two sub-categories, namely the parametric active contour model and the geometric active contour model. The difference of these two kinds of functional-based segmentation algorithms is that the parametric active contour model uses parameters to characterize the shape of the foreground boundary explicitly, while the geometric active contour model represents the foreground boundary implicitly with the level sets of a 3D function.

In [39], Kass et al. proposed a parametric active contour model, Snakes, which used several control points on the foreground contour to form the basic curves and deformed the curves by changing the values of their parameters. The energy function of the Snakes model consists of two kinds of energy, namely the internal energy and the external energy. The internal energy can control the elastic deformation and keep the smoothness and continuity of the curve. The external energy is also called image energy, which indicates the degree of similarity between the form of the curve and the local feature, such as gradient, intensity, or texture feature. Snakes is a semi-automatic segmentation algorithm and users can impose constraint forces to guide the deformation of the contours near the features of interest. However, the segmentation results strongly depend on the original contours determined by the control points. The Snake model can not represent some boundaries well when the topologies of boundaries changed [54], such as the breaking and merging. Figure 2.2 and figure 2.3 show the situations where Snake model can't handle.

The level set methods [23, 44, 67, 70] is a geometric active contour model. It determines the contours of the foreground region by converting the 2D images to a 3D surface basing on gradient value of intensity of pixels in images. The 3D surface is evolved until its energy function is minimized. At first, a 2D image is usually converted to a 3D surface basing on gradient values of pixels' intensity. The projection of the 3D surface on the zero-level set represents the contour of the foreground region of the original 2D image. Suppose $C(x, y)$ is a 2D contour curve and $C(t)$ is its position at time t, which is implicitly defined as the zero-level set of the 3D surface $\psi(C(t), t)$, namely $\psi(C(t), t) = 0$. The evolution of the 3D surface can be describe by equation 2.1,

$$\frac{\partial\psi\left(C\left(t\right),t\right)}{\partial t}+F\left(C\left(t\right)\right)\cdot\left|\nabla\psi\left(C\left(t\right),t\right)\right|=0, \tag{2.1}$$

where $F\left(C\left(t\right)\right)$ is a speed function. Given the value of $\psi\left(C\left(t\right), t=0\right)$, the value of $\psi\left(C\left(t\right), t\right)$ can be iteratively calculated according to equation 2.1 and thus the contour of the foreground region of the 2D image can be evolved. As the level set methods don't depend on the parameters of curves to describe their shape, they can deal with the problems that the topologies of curves may change in the process of curve evolution, such as breaking and merging [66, 80]. The disadvantage is that it is computationally expensive to find the minimum of the energy function iteratively, which limits the speed of the level set algorithm.



(a)

(b)

(c)

(d)

Figure 2.2: An example of the change in topology: breaking. (a) The original closed curve (the red curve). (b) The original curve breaks into two curves. (c) The two curves break into three curves. (d) The red curves stop on the boundaries of the foreground regions.

2.3 Evaluation Methods of Segmentation Algorithms

The performance of segmentation algorithms can be evaluated in different ways. Accuracy is the most often used metric compared to others, such as reproducibility, efficiency, computation time, and etc. Reproducibility is only used for semi-automatic segmentation algorithms or other types of algorithms which need users' interaction or intervention. Computation time [8, 26, 29, 38, 53, 58, 82] is reported to show the speed of segmentation algorithms. In some papers, ”*efficiency*” refers to some measure of the amount of annotation [60, 73].

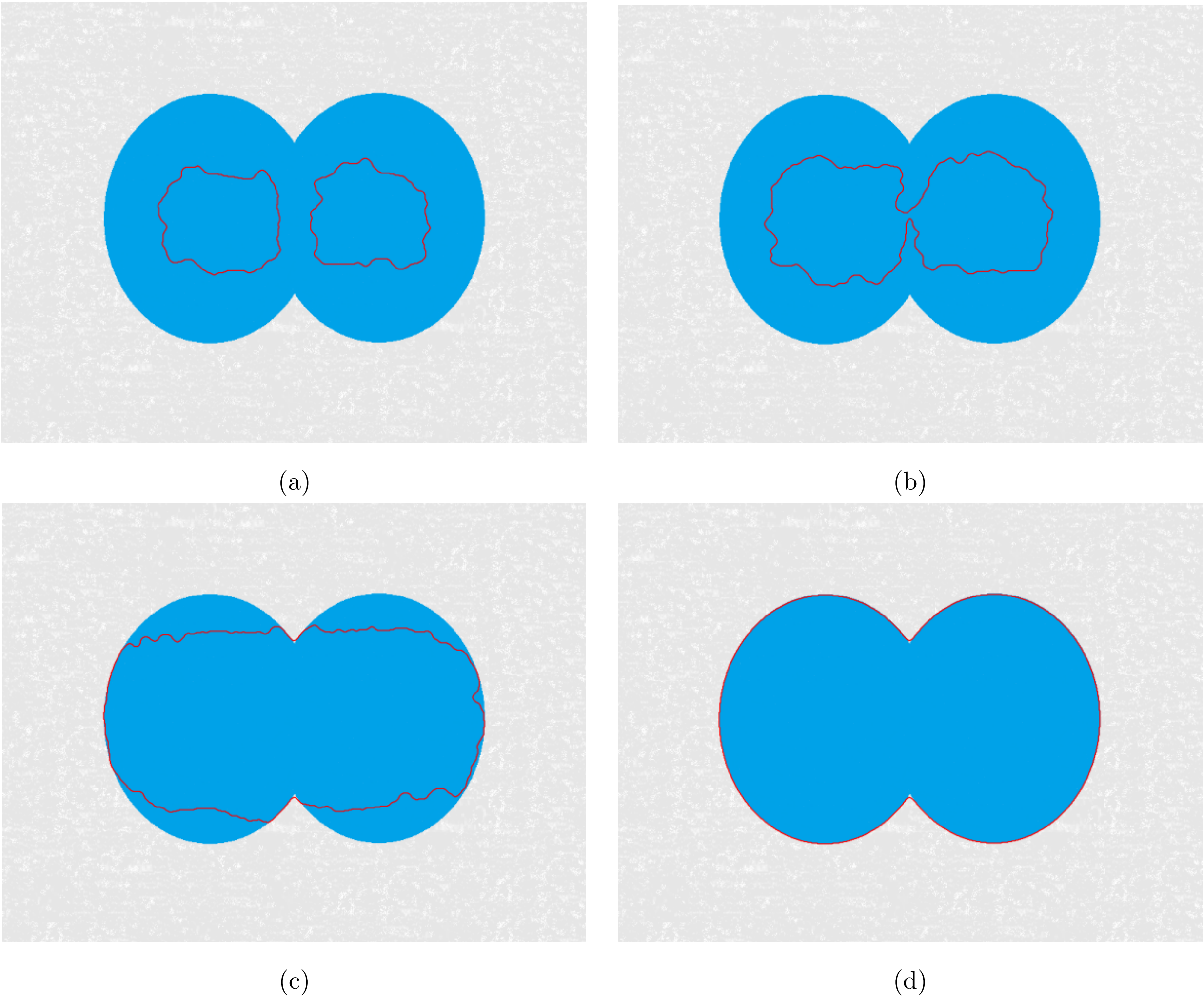


Figure 2.3: An example of the change in topology: merging. (a) The original closed curves (the red curves). (b) The original two curves merge into one curves. (c) The merged curve keeps evolving. (d) The red curve stops on the boundary of the foreground region.

2.3.1 Accuracy

Accuracy is the measure of the similarity of a segmentation to its ground truth. There are many ways to measure that similarity. Accuracy measures can be roughly categorized as the boundary-based accuracy measures and the region-based accuracy measures.

Boundary-Based Accuracy Measures

Boundary-based accuracy measures always use the distance between the boundary of foreground region of the segmentations and the boundary of foreground region of the ground truth to quantify segmentation accuracy. Let $B = \{b_1, b_2, ..., b_M\}$ be a finite pixel set, in which the b_i ($i \in \{1, 2, ..., M\}$) is a pixel on the segmented boundary and $T = \{t_1, t_2, ..., t_N\}$ be a finite pixel set, in which the t_j ($j \in \{1, 2, ..., N\}$) is a pixel on the ground truth boundary. Define $d(b_i, t_j)$ as the distance between b_i and t_j, where the distance may be Euclidean distance, Chebyshev distance, Manhattan distance or other metric of spatial distance. The theories of boundary-based accuracy measures are illustrated in this section, including mean squared distance (MSD), mean absolute distance (MAD), root mean squared distance (RMSD) and root mean absolute distance (RMAD), maximum difference (MaxD) and Hausdorff distance (HD).

- **Mean squared distance and mean absolute distance**

  The mean squared distance and the mean absolute distance are defined in equation 2.2 and equation 2.3,

$$MSD(B,T) = \frac{\sum_{b \in B}(\min_{t \in T}\ d^2(b,t))}{M}, \tag{2.2}$$

$$MAD(B,T) = \frac{\sum_{b \in B}(\min_{t \in T}\ |d(b,t)|)}{M}, \tag{2.3}$$

  where B and T are the set of pixels in the segmented boundary and the set of pixels in the ground truth boundary. MSD is also referred to as mean squared difference or mean squared error. It is used to measure the average minimum deviation of the segmented boundary from the ground truth boundary. MAD also refers to mean absolute error in [14]. It can measure the average minimum absolute distance between the segmented boundary and the ground truth boundary too.

- **Root mean squared distance**

  RMSD is the square root average minimum distance between the segmented boundary and the ground truth boundary. The definition of RMSD is given in equation 2.4.

$$RMSD(B,T) = \sqrt{MSD} = \sqrt{\frac{\sum_{b \in B}(\min_{t \in T}\ d^2(b,t))}{M}} \tag{2.4}$$

  It is easy to know that MAD and RMSD have the same unit with the spatial distance. But that doesn't mean only one of them should be used to represent the performances of segmentation algorithms. Chai and Draxler [14] demonstrated that when the error distribution is expected to be Gaussian, RMSD is more appropriate to characterize the model performance than the MAD. Instead, the combination of different metrics should be used to assess the performances of segmentation algorithms.

- **Maximum difference**

  MaxD provides a measure of the maximum error in segmentations [37]. In order to calculate MaxD, the center of gravity of the ground truth boundary is determined at first, which is the average of the weighted position of the intensity of pixels in the ground truth boundary. Then, l, the distance between the segmented boundary and the ground truth boundary can be calculated as a function of angle of θ_i, where $i = 1, 2, ..., N$ and N is the number of radial angles. At last, the maximum of the distance of the segmented boundary and the ground truth boundary will be determined. The definition of MaxD is in equation 2.5:

$$MaxD(B,T) = \max\{|l(\theta_i)|\} \tag{2.5}$$

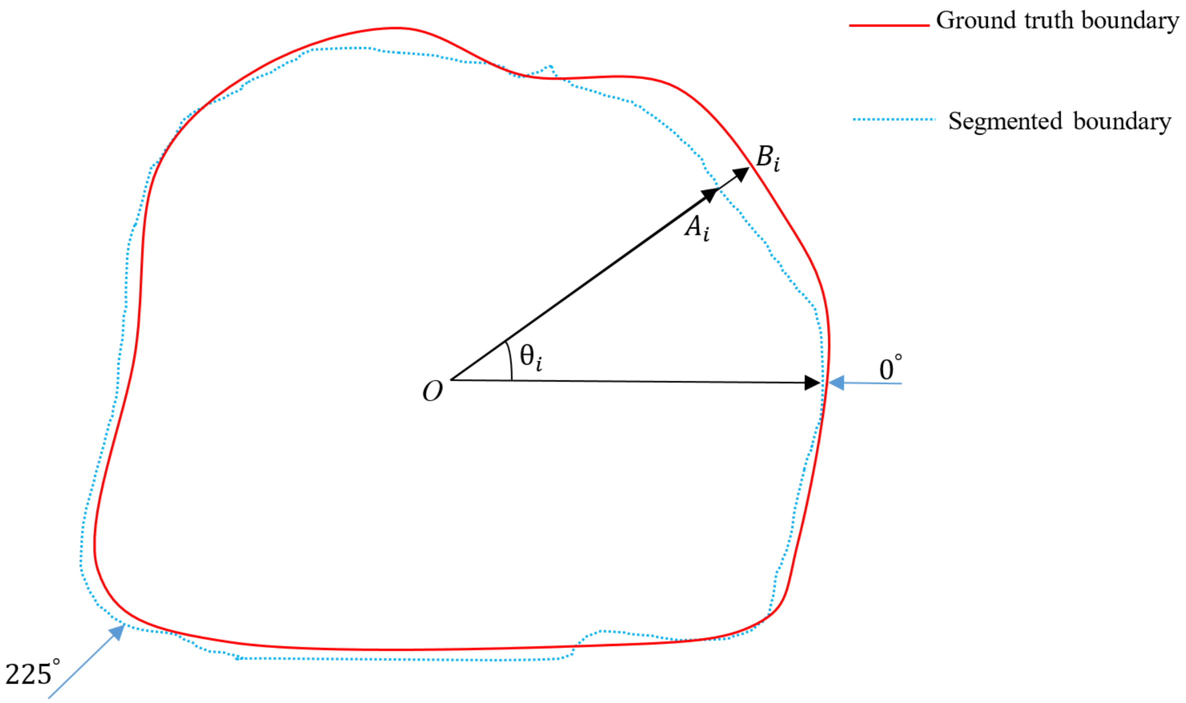


Figure 2.4: Local difference between the ground truth boundary and the segmented boundary.

Figure 2.4 shows the local difference between the two different boundaries and $l(\theta_i) = \overline{OA_i} - \overline{OB_i}$, where O is the center of gravity of the ground truth boundary, A_i is a pixel on the segmented boundary, B_i is a pixel on the ground truth boundary.

- **Hausdorff distance**

The Hausdorff distance (HD) can be applied on two sets of boundary pixels extracted from the segmented boundary and the ground truth boundary. It measures the maximum distance between the pixels in the segmented boundary set and the pixels in the ground truth boundary set. The HD is defined in equation 2.6:

$$HD(B,T) = \max \left\{ \max_{b \in B} \min_{t \in T} d(b,t), \max_{t \in T} \min_{b \in B} d(t,b) \right\} \tag{2.6}$$

HD has the same unit with MaxD. Both of them measure the max difference between the segmentation boundary and the ground truth boundary. However, no related work was done to analyze which measure is more suitable to measure the accuracies of segmentations. Therefore, it is recommended to use both of them to characterize the accuracies of segmentations.

This metric can be used to evaluate the performance of both 2D images and 3D images. In many situations, HD is used together with MSD as a complementary measure, since the MSD measures the mean error and the HD measures the maximum error.

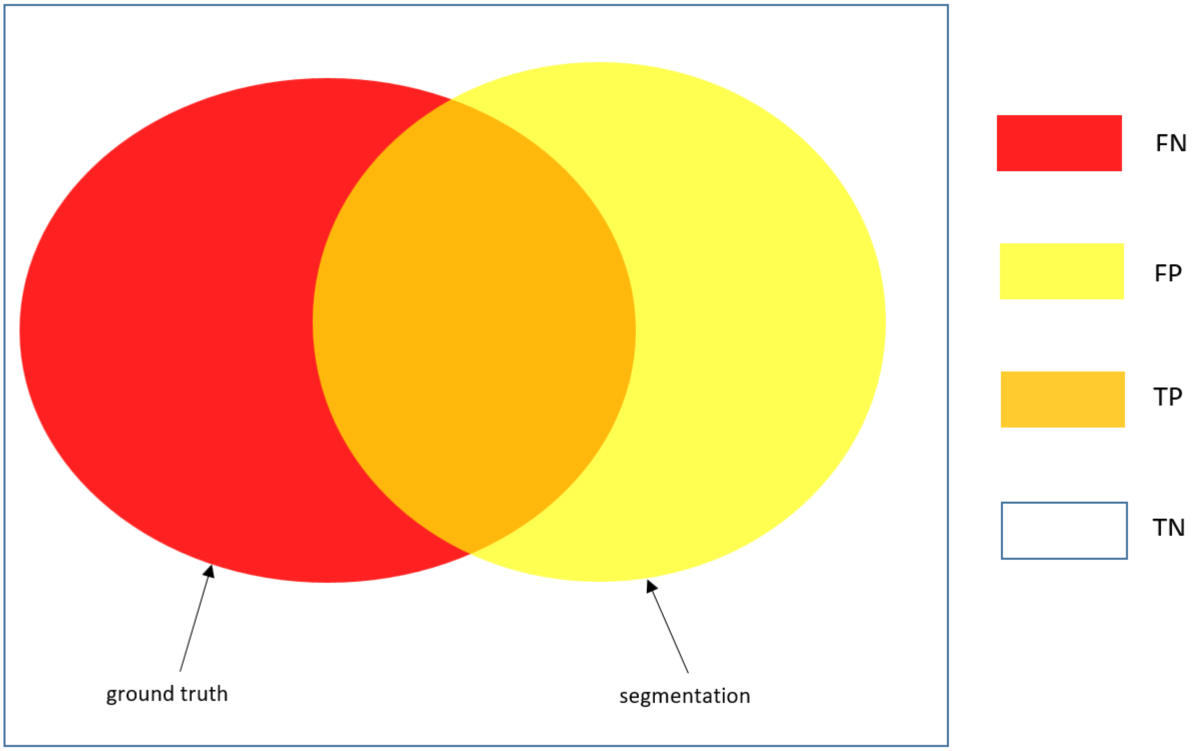


Figure 2.5: The illustration of TP, TN, FP, and FN. The two color ellipses are the foreground regions in the segmentation and the corresponding ground truth, respectively.

Region-Based Accuracy Measures

Region-based accuracy measures characterize the accuracy of segmentation algorithms basing on the overlapped region of the ground truths and the segmentations rather than only considering the boundaries.

The pixels in the segmentations and their corresponding ground truth can be classified into four categories: true positive pixels (TP), false positive pixels (FP), true negative pixels (TN) and false negative pixels (FN). An example is used to illustrated the relationship of these four kinds of pixels in figure 2.5. Suppose $cnt(\cdot)$ means the count of pixels which belong to a category or a region of an image.

- **Sensitivity and specificity**

  Sensitivity, which is also called true positive fraction (TPF), is defined in equation 2.7. It is the proportion of the pixels in the foreground region are correctly classified as foreground.

  $$sensitivity = TPF = \frac{cnt(TP)}{cnt(TP) + cnt(FN)} \tag{2.7}$$

  The specificity has some relation with false positive fraction (FPF) and is defined in equation 2.8:

  $$specificity = 1 - FPF = 1 - \frac{cnt(FP)}{cnt(FP) + cnt(TN)} = \frac{cnt(TN)}{cnt(FP) + cnt(TN)} \tag{2.8}$$

  It is the proportion of the pixels in the background are classified into the background region.

- **Classification accuracy**

  The classification accuracy (CA) is the proportion of the number of correctly classified pixels to the total number of pixels in the segmented image. CA can be calculated using the following equation:

  $$CA = \frac{cnt(TP) + cnt(TN)}{cnt(TP) + cnt(FP) + cnt(FN) + cnt(TN)} \tag{2.9}$$

  All of sensitivity, specificity and classification accuracy are usually used in medical diagnosis. However, it may be not appropriate to use these three metrics to evaluate the performance of segmentation algorithms, because they will inflate the performance rating when $cnt\,(TN)$ is far greater or far smaller than $cnt\,(TP)$.

- **Dice similarity coefficient**

  Dice similarity coefficient is a statistic used for comparing the similarity of samples in two sets. The DSC, defined in equation 2.10, is the ratio of the area of the overlapped region of the segmentation and the ground truth to the average area of the segmentation and the ground truth.

  $$DSC = \frac{2cnt(TP)}{2cnt(TP) + cnt(FP) + cnt(FN)} \tag{2.10}$$

  DSC takes values from 0 to 1. It means the segmentation and the ground truth are perfectly matched when $DSC = 1$ and the segmentation and the ground truth are completely separated from each other when $DSC = 0$.

- **Tanimoto coefficient**

  The Tanimoto coefficient (TC), also referred to as Intersection over Union, Jaccard Similarity, and Jaccard Index, is the size of the intersection divided by the size of the union of the sample sets:

  $$TC = \frac{S \cap G}{S \cup G} = \frac{cnt(TP)}{cnt(TP) + cnt(FP) + cnt(FN)} \tag{2.11}$$

  where S is the set of pixels in the foreground of segmentation and G is the set of pixels in the ground truth. TC also takes values from 0 to 1, where 1 means the segmentation and the ground truth matched perfectly and 0 means they are mismatched completely.

2.3.2 Reproducibility

The reproducibility of an image segmentation algorithm refers to the consistency of the segmentations that are generated by the implementation of the image segmentation algorithm using different annotations of users. It is only used for the semi-automatic or manual segmentations where the users' interactions are necessary. In addition, the reproducibility of a segmentation algorithm is evaluated without the use of ground truths. Therefore, the reproducibility does not include the comparisons against the ground truths, only between different users' segmentations.

To measure the algorithms thoroughly, both the inter-observer reproducibility and the intra-observer reproducibility should be evaluated. The inter-observer reproducibility is used to represent the similarity of segmentations generated by using the interactions of different users. The intra-observer reproducibility is used to represent the similarity of segmentations generated by using the interactions of the same user.

As the reproducibility measures the similarity of several groups of segmentations, the metrics that used to evaluate accuracy can also be used to access reproducibility, as long as these metrics can be operated on several segmentations of an image and don't depend on the ground truths or accuracies of them. In this section, the theories of several reproducibility measures are illustrated, such as joint Dice coefficient (JDC), joint Tanimoto coefficient (JTC), coefficient of variation (CV), intra-class correlation coefficient (ICC), generalized Tanimoto coefficient (GTC).

- **Joint Dice coefficient and joint Tanimoto coefficient**

  JDC and JTC are the generalization of DSC and TC to assess reproducibility. It is not necessary to use ground truths to calculate JDC and JTC, which is different from DSC and TC. However, to determine the JDC and JTC, a group of segmentations are necessary. Suppose n is the number of the segmentations and $S = \{S_1, S_2, ..., S_n\}$ is a group of segmentations of the same image. JDC and JTC can be expressed by equation 2.12 and equation 2.13.

$$JDC = \frac{n \cdot cnt\left(\cap_{i=1}^{n} S_i\right)}{\sum_{i=1}^{n} cnt\left(S_i\right)}, \tag{2.12}$$

$$JTC = \frac{cnt(\cap_{i=1}^{n} S_i)}{cnt(\cup_{i=1}^{n} S_i)}, \tag{2.13}$$

  JDC is the fraction of the size of the overlapped region of several segmentations to the average size of all these segmentations. JTC is the proportion of the overlap area of all segmentations in the set S to the area of their union.

- **Coefficient of variation and intra-class correlation coefficient**

  Both coefficient of variation (CV) and intra-class correlation coefficient (ICC) are statistical metrics. They can not only be used to measure the reproducibility of segmentation algorithms, but also be used in other fields for reliability studies.

  CV is also referred to as coefficient of dispersion or relative standard deviation, which measures the variability of a group of samples. As for image segmentation, CV is often used to measure the reproducibility of area or volume measurements of medical images, especially for computed tomography (CT) and magnetic resonance imaging (MRI). CV can be defined as:

$$CV = \frac{\sigma}{\mu}, \tag{2.14}$$

  where σ and μ are the standard deviation and mean of the area (the number of pixels for 2D images) or volume (the number of voxels for 3D images) of foreground parts of segmentations.

ICC is widely used in reliability analysis. For image segmentation, ICC can measure the reproducibility of both the inter-observers segmentations and the intra-observer segmentations [18, 40].It describes how similar the value of samples in the same group are to each other and how different they are from samples in other groups. The sample in a group is a segmentation of an input image. Many forms of ICC are summarized by Koo and Li in [43] to help clinical researchers to choose the correct form of ICC. According to them, the reproducibility of a segmentation algorithm can be interpreted as poor, moderate, good, and excellent when the values of ICCs are in the intervals of $[0, 0.5]$, $(0.5, 0.75)$, $(0.75, 0.9]$, and $[0.9, 1]$, respectively, based on the 95% confidential interval of ICC estimate. But it remains unclear whether this interpretation of ICC is reasonable for the reproducibility of semi-automatic segmentation algorithms.

- **Generalized Tanimoto coefficient**

  Several different forms of GTC are defined in [17] to characterize the consistency of multiple segmentations. Suppose $S = \{S_1, S_2, ..., S_n\}$ is the set of all segmentations of the same image. One form of GTC, which is suitable for measuring the reproducibility of binary segmentation algorithms, is defined in the following equation:

  $$GTC = \frac{\sum_{i=1}^{n}\sum_{j=i+1}^{n}\sum_{k}\min(S_{ik}, S_{jk})}{\sum_{i=1}^{n}\sum_{j=i+1}^{n}\sum_{k}\max(S_{ik}, S_{jk})}, \tag{2.15}$$

  where S_{ik} is the label value (foreground: 1, background: 0) of the kth pixel in the ith segmentation. GTC takes on values from 0 to 1.0, where $GTC = 0$ represents entirely disjoint segmentations and $GTC = 1.0$ represents all $S_i \in S$ are identical.

To get a statistically significant measurement of the reproducibility of a segmentation algorithm, a large number of segmentations are required. However, it may be difficult to collect enough segmentations in practice because of the limitation of funding, the number of well-trained operators who can perform annotations or labeling, and so on. In [22], Eramian used synthetic data to demonstrate that CV, GTC and ICC are not sensitive at all to the group size of segmentations when the group size is greater than 10. However, JDC, as well as JTC, varies a lot. But when the group size of segmentations is smaller than 5, only GTC is stable. It means in the situation where the group size of segmentations is between 2 and 5, only GTC is suitable for measuring the reproducibility of segmentation algorithms. In practice, the group size is always smaller than 5 due to the limitations such as funding issue and logistic support. So it is very likely that only GTC is not sensitive to the group size of segmentations.

3. Research Objectives

3.1 Research Statement

As is mentioned in chapter 1 and 2, there are some problems when evaluating and ranking segmentation algorithms. The first one is what aspects of performance of a segmentation algorithm should be evaluated. In most papers, only the accuracy performance of their proposed segmentation algorithm is reported. However, accuracy itself can't tell the total story of a segmentation algorithm. Efficiency is also important, especially for those algorithms which are designed for real-time segmentation problems. In particular, the reproducibility of semi-automatic segmentation algorithms should be evaluated, too. The second problem is what metrics should be applied. For example, many metrics can measure reproducibility, such as CV, GTC, ICC, JDC and JTC. Which one, or any combination of them, should be used to evaluate the reproducibility of a semi-automatic segmentation? The third one is how to interpret the values of these measures. For example, when the accuracies of several segmentations are calculated, sometimes it's not a good idea to use the mean and standard deviation of accuracies to characterize the accuracy performance of a segmentation algorithms. Different segmentation algorithms may have the same mean and standard deviation of accuracies, while their distributions differ a lot. In this situation, mean and standard deviation are unfavorable for ranking segmentation algorithms.

3.2 General Objectives

This study focuses on how to evaluate and rank segmentation algorithms on the basis of their accuracies, as well as which measure should be used to evaluate the reproducibility of semi-automatic segmentation algorithms.

3.3 Specific Objectives

- To illustrate how to use a distribution-based methodology to evaluate and compare the accuracy of segmentation algorithms.
- To develop an extensive method for ranking the accuracies of segmentation algorithms that is richer than simply comparing mean accuracy measures.
- To demonstrate generalized Tanimoto coefficient (GTC) is more stable than JDC, JTC, CV and ICC for measuring the reproducibility of segmentation algorithms when the group size varies.

4. $\overline{CDF}(\theta)$: A Distribution-based Function

4.1 Research Problem

Many image segmentation algorithms had been reported to have excellent performances. However, most of these results are based on the mean value of accuracy measures, which ignore the distributions of the accuracies of segmentation algorithms. When using mean and standard deviation to characterize a group of data points, an assumption is made that these data points are normally-distributed. If the distribution is highly non-normal, it may be inappropriate to use mean value of accuracy measures to evaluate and rank segmentation algorithms. Because of this, the accuracy of segmentation algorithms should be evaluated in other ways that account for these properties. In this chapter, a distribution-based function, $\overline{CDF}(\theta)$, is introduced to analyze and compare the accuracy performances of segmentation algorithms.

4.2 Data Analysis Methodology

4.2.1 The Standard Method

The *standard method* is using the mean, and standard deviation of accuracy of segmentations to characterize the accuracy performance of the implementations of segmentation algorithms. When comparing different segmentation algorithms, hypothesis tests may also be necessary to find whether the means of accuracies of the segmentations generated by these segmentation algorithms are significantly different.

As is introduced in the introduction section, the mean of accuracies is widely used to characterize the performances of segmentation algorithms. When the mean accuracies are calculated, the performances of segmentation algorithms can be ranked. The segmentation algorithm with a higher mean accuracy is considered to have a better accuracy performance. This may be a good characterization when the accuracy of segmentations generated by different segmentation algorithms have very different mean accuracies. Otherwise, it is more convincing to use hypothesis tests to determine whether those accuracies are from the same distribution. If those accuracies are not from the same distribution, the segmentation algorithm which has higher mean accuracy is better. If not, it means there is insufficient evidence to rank one over the other.

The Mann-Whitney U test, a non-parametric hypothesis test, might be used to see whether two groups of data are from the same distribution. Given two sets of data, the null hypothesis of Mann-Whitney U test is that the randomly selected sample from a set of data is equally likely greater or smaller than the randomly selected sample from the other set of data. It can also be illustrated as whether the randomly selected two samples are from the same distribution.

4.2.2 The Explanation of $\overline{CDF}(\theta)$

It is easy to imagine that a probability density function (PDF) can be used to characterize the accuracy performance of a segmentation algorithm. When measuring the accuracy performance of a segmentation algorithm, the independent variable of a PDF is the accuracy of segmentations generated by the segmentation algorithm and it takes values in the continuous interval of $[0, 1]$.

However, we can't get the PDF of an algorithm directly. What we can do is to sample it. We can generate a finite number of segmentations and a finite number of accuracy values. Then we can estimate the underlying PDF of a segmentation algorithm using histograms. A histogram is a representation of the distribution of numerical data. In this thesis, the histogram estimation of a probability density function is denoted by HPDF.

The method of calculating HPDF is as follow. First, we use an algorithm to segment N images and calculate the accuracies of these N segmentations. Second, we divide the interval of $[0, 1]$ into M intervals equally, so that each interval has the same width of $\frac{1}{M}$. The ith interval is $\left[\frac{i-1}{M}, \frac{i}{M}\right)$, where $i \in \{1, 2, ..., M\}$. Third, for each interval (bin), the number of accuracy values which fall into it is counted. Fourth, the counts of accuracy values of each bin are divided by N. In these N segmentations, if there is a segmentation whose accuracy is x , x will fall into the $bin_{\lfloor x/\frac{1}{M}\rfloor}$ ($\lfloor\cdot\rfloor$ is the floor fuction), which is the $\lfloor x/\frac{1}{M}\rfloor$th bin of the constructed histogram. So the HPDF of an algorithm can be expressed as

$$HPDF(x) = \frac{cnt\left(bin_{\lfloor x/\frac{1}{M}\rfloor}\right)}{N} = \frac{cnt\left(bin_{\lfloor x\cdot M\rfloor}\right)}{N}, \tag{4.1}$$

where $x \in [0, 1]$ and $cnt(\cdot)$ is the count of accuracies that fall into a bin. Figure 4.1 shows the relationship of the HPDF, the underlying PDF of a group of Gaussian-distributed data, as well as one bin of HPDF (colored in gray).

It should be noted that the number of bins of the constructed histogram can affect the shape of the HPDF curve. The examples are shown in Figure 4.2. If there are too many bins, there may be some bins in which few or no accuracy samples fall, resulting in erroneous probability estimates. But if there are too few bins, the width of each bin will be very large, which makes it difficult to learn the shape of the underlying PDF.

There are different ways to decide the number of bins, such as the square-root choice, or using Sturges' formula. For the square-root choice, the number of bins is equal to the square-root of the number of samples,

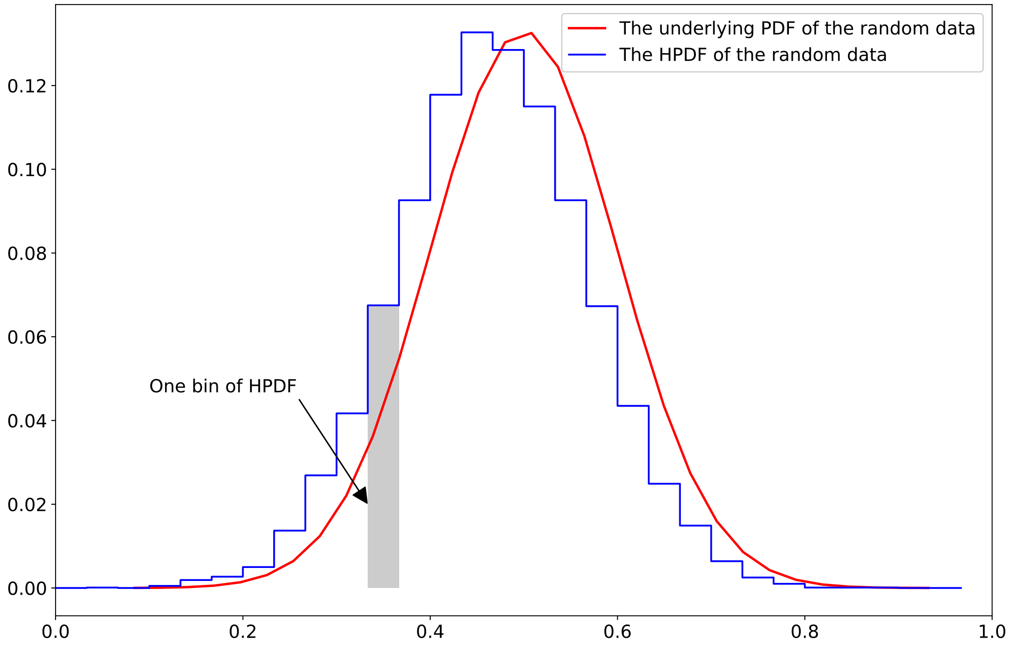


Figure 4.1: The HPDF and the underlying PDF of a group of synthetic Gaussian-distributed data (number of samples is 10000, $\mu = 0.5$, $\sigma = 0.1$, number of bins is 30). One bin of the HPDF is colored in gray.

namely

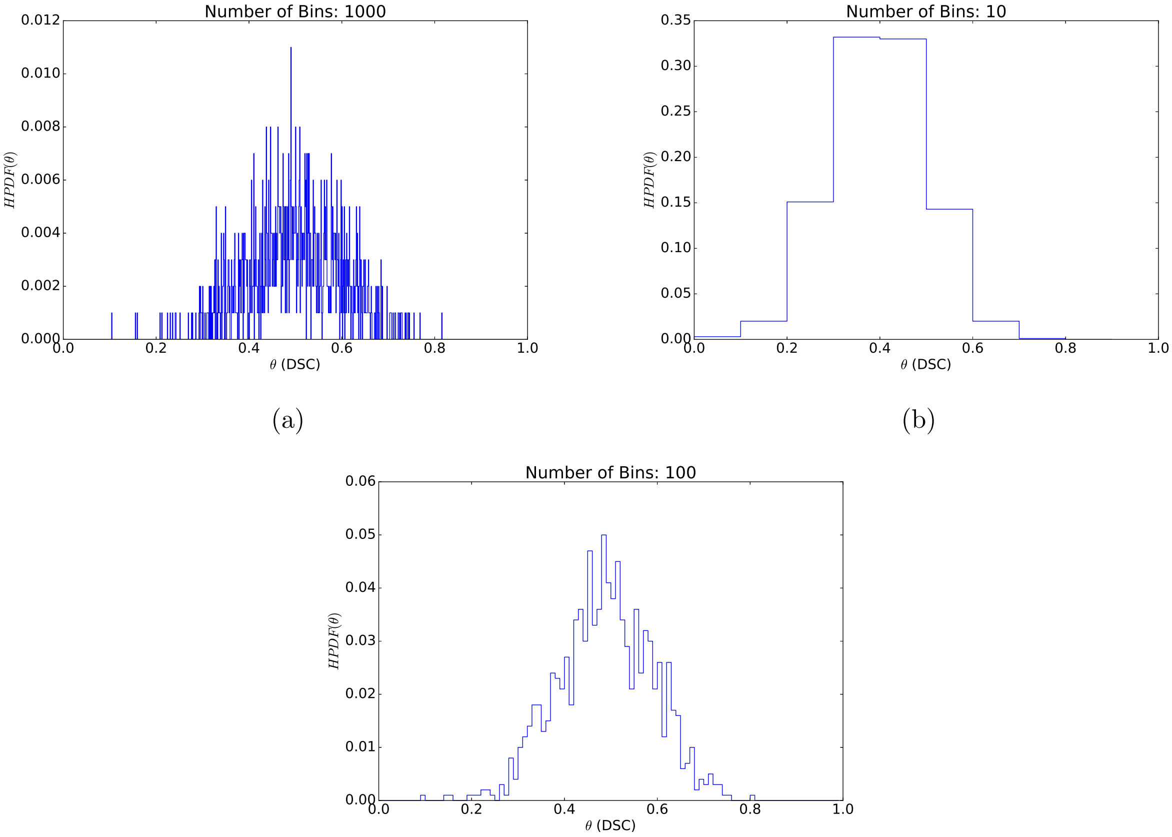
$$M = \sqrt{N}. \tag{4.2}$$

In [72], Sturges suggested the number of bins can be calculated with

$$M = \lceil \log_2 N \rceil + 1, \tag{4.3}$$

where $\lceil \cdot \rceil$ indicate the ceiling function. However, when we try these methods, the number of bins is too few and does not capture the shape of the underlying PDF well, as is shown in Figure 4.2. Therefore, it is recommended to try different number of bins and find a trade-off about the effects of bins' number.

HPDF can be used to estimate the underlying PDF of a group of data generated by a segmentation algorithm, but it is not a wise choice to use HPDF to compare two or more groups of data directly. If we use HPDFs to compare two groups of data generated by different segmentation algorithms, the values of all bins of these two HPDFs should be compared. However, the values of some bins of these two HPDFs could be 0. This may happen when there are too many bins in the constructed histogram, or there are too few segmentations and therefore too few accuracies. In this situation, we can't decide which algorithm is better. Since we are not sure whether it is because both these two algorithms can't generate segmentations with these specific accuracies, or we don't observe these accuracies over the finite number of segmentations used



(a)

(b)

(c)

Figure 4.2: The HPDFs of a group of Gaussian-distributed data (number of samples is 1000, $\mu = 0.5$, $\sigma = 0.1$) with different numbers of bins. (a) Too many bins. (b) Too few bins. (c) Proper number of bins.

to generate the HPDF.

In order to avoid the problem that the counts of some bins of HPDFs are 0, we consider to use cumulative density function (CDF) of accuracies of segmentations to characterize the accuracy performances of segmentation algorithms. CDF can be expressed as

$$CDF(x) = \sum_{i=0}^{\lfloor x/\frac{1}{M} \rfloor} HPDF(\frac{i}{M}) = \sum_{i=0}^{\lfloor x \cdot M \rfloor} HPDF(\frac{i}{M}), \tag{4.4}$$

which is the summation of values of the first $\lfloor x \cdot M \rfloor$ bins of the HPDF. It represents the probability that the accuracy of a segmentation generated by an algorithm will take a value less than or equal to x, where $x \in [0, 1]$. $CDF(x)$ is a monotone increasing function.

CDF can be used to characterize and compare the accuracy performances of two segmentation algorithms. Suppose the CDFs of segmentations generated by algorithm A and algorithm B are $CDF_A(x)$ and $CDF_B(x)$, where x is accuracy level. If $CDF_A(x) > CDF_B(x)$, it means algorithm A is more likely to generate segmentations whose accuracies are no greater than x.

In practice, we are more interested in the likelihood of an accuracy of x or higher, which is not consistent with the implication of CDF. Because of this, $\overline{CDF}(\theta)$, which is defined in equation 4.5, is used to characterize the accuracy performances of segmentation algorithms in this paper, instead of $CDF(x)$.

$$\overline{CDF}(\theta) = 1 - CDF(\theta) = 1 - \sum_{i=0}^{\lfloor \theta \cdot M \rfloor} HPDF(\frac{i}{M}), \tag{4.5}$$

where $\theta \in [0, 1]$ is the accuracy level. Note that the x in equation 4.4 has the same meaning with the θ in equation 4.5. The reason why we used different parameters to indicate accuracy is that we want to distinguish the CDF and $\overline{CDF}$. $\overline{CDF}(\theta)$ is the estimated probability of that the algorithm will generate segmentations with accuracies of θ or higher. We thought it would be easier to interpret the metric in the context of segmentation algorithm performance, so we use $\overline{CDF}(\theta)$ instead of $CDF(\theta)$. As the $CDF(\theta)$ is a monotone increasing function and ranging from 0 to 1.0, $\overline{CDF}(\theta)$ is monotonically decreasing with increasing θ. A superior segmentation algorithm's $\overline{CDF}(\theta)$ equals to 1 when $\theta < 1.0$ and sharply decreases when $\theta \approx 1.0$.

$\overline{CDF}(\theta)$ is not only a probability, but also a metric for characterizing the performance of segmentation algorithms. We can use $\overline{CDF}(\theta)$ to characterize the accuracy performance of a segmentation algorithm in three steps. First, we use the segmentation algorithm to segment images and calculate the accuracies of the segmentations. Second, the $\overline{CDF}(\theta)$ of these accuracies is calculated using equation 4.5. Third, check the values of $\overline{CDF}(\theta)$s when θ takes different accuracy values that we are concerned about and see how likely the segmentation algorithm is to generate segmentations with accuracies of θs or higher. For example, suppose we are concerned about the probability that an algorithm can generate segmentations with accuracies of 0.9 or higher. First, we need to use this algorithm to segment several images and calculate the accuracies of these segmentations. Then, we can draw the $\overline{CDF}(\theta)$ of these accuracies values. Next, the value of $\overline{CDF}(0.9)$ should be calculated. The greater the value of $\overline{CDF}(0.9)$, the better the accuracy performance of this segmentation algorithm.

We can also use the area under the $\overline{CDF}(\theta)$ curve (AUC) to characterize the overall accuracy performance of a segmentation algorithm. The AUC refers to the area of the region that above the x-axis, on the right of the y-axis and under the $\overline{CDF}(\theta)$ curve. As the accuracy $\theta \in [0, 1.0]$ and $\overline{CDF}(\theta) \in [0, 1.0]$, $AUC \in [0, 1.0]$. According to equation 4.5, $\overline{CDF}(\theta)$ is the probability that an algorithm can generate segmentations with accuracies of θ or higher. Therefore, when $\theta < 1.0$, the $\overline{CDF}(\theta)$ of a superior segmentation algorithm equals to 1.0 and when $\theta \approx 1.0$, $\overline{CDF}(\theta)$ of a superior segmentation algorithm decreases sharply from 1.0 to 0. So the AUC of a superior segmentation algorithm is close to 1.0. The greater the AUC, the better the performance of a segmentation algorithm. Therefore, the AUC can also be used to compare the overall accuracy performances of two segmentation algorithms.

$\overline{CDF}(\theta)$ can not only be used to characterize the accuracy performance of one segmentation algorithm, it can also be used to compare the accuracy performances of two segmentation algorithms. The difference of two $\overline{CDF}(\theta)$s can be expressed as

$$\overline{CDF}_{diff}(\theta) = \overline{CDF}_1(\theta) - \overline{CDF}_2(\theta), \tag{4.6}$$

where $\overline{CDF}_1(\theta)$, $\overline{CDF}_2(\theta)$ are the $\overline{CDF}(\theta)$s of segmentation algorithm 1 and segmentation algorithm 2. According to equation 4.6, given an accuracy level θ, the possibility that the algorithm 1 can segment images with accuracy of θ or higher is greater than that of algorithm 2 if $\overline{CDF}_{diff}(\theta) > 0$, and less otherwise. Therefore, given a θ, the accuracy performances of two algorithms can be compared using the value of $\overline{CDF}_{diff}(\theta)$ directly.

In fact, $CDF_{diff}(\theta)$, which is defined in equation 4.7, can also be used to compare accuracy performances of segmentation algorithms.

$$CDF_{diff}(\theta) = CDF_1(\theta) - CDF_2(\theta), \tag{4.7}$$

However,if the value of $CDF_{diff}(\theta)$ is greater than 0, it means algorithm 2 has a better accuracy performance than the algorithm 1. If the value of $CDF_{diff}(\theta)$ is smaller than 0, it means algorithm 1 has a better accuracy performance than the algorithm 2. Comparing equation 4.6 and equation 4.7, we can see that the $\overline{CDF}_{diff}(\theta)$ is easier to interpret.

4.3 Examples of Using $\overline{CDF}(\theta)$ to Characterize and Compare the Accuracy Performances of Segmentation Algorithms

4.3.1 Example 1: the Leafsnap Dataset

The Leafsnap dataset [45] contains the photos of 85 kinds of trees' leaves as well as their binary ground truths. For the original images of this dataset, the backgrounds are white papers and the foregrounds are leaves. 7717 photos of leaves of different kinds of trees in the Leafsnap dataset were adopted in total. These images were taken by mobile devices (iPhones mostly) in outdoor environments. They contain varying amounts of blur, noise, illumination patterns, shadows, etc. The example images are shown in Figure 4.3. For the ground truths, foreground regions (leaves) are marked in white and background regions are in black. As leaf is the only foreground object in the image in the Leafsnap dataset, the accuracy of segmentations will be high in general, even in the presence of blur, noise and etc.

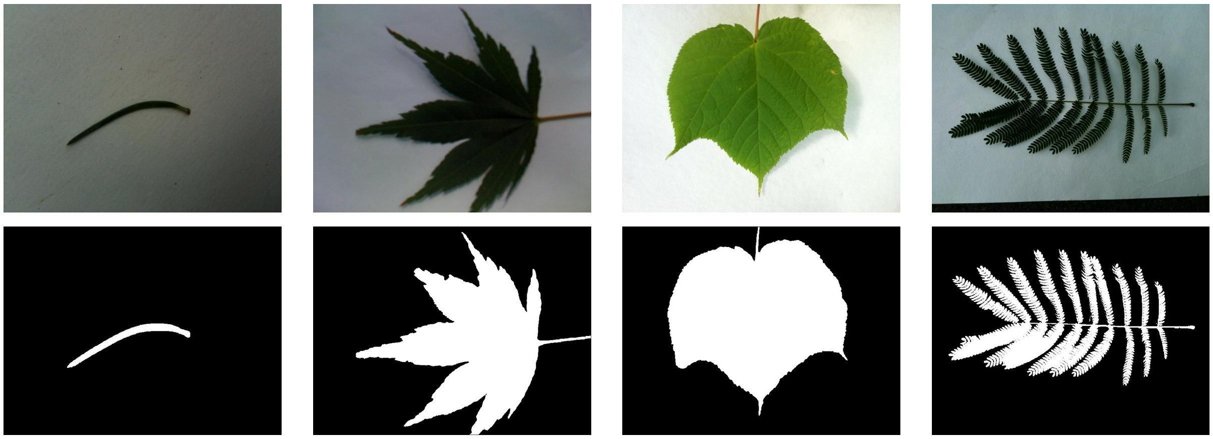


Figure 4.3: Example images in the Leafsnap dataset. Top: original images, bottom: ground truths.

The implementations used to do the segmentation work are based on three algorithms: the Morphological Snakes [2], the Random Walker [25], and the U-Net algorithms [1]. The theories of these three algorithms can be found in the background and literature review section.

For all of the segmentations generated by these three segmentation algorithms, the accuracies are measured with Dice similarity coefficient (DSC), which was introduced in section 2.3.1.

Then we plot the $\overline{CDF}(\theta)$s of these three algorithms and they are shown in Figure 4.4 . The $HPDF(\theta)$s of these algorithms are also shown in Figure 4.4 in order to reveal the underlying probability density functions of these three algorithms. Note that for this example, the number of bins of $HPDF(\theta)$s and $\overline{CDF}(\theta)$s are 200, therefore the width of each bin is 0.005.

D'Agostino-Pearson normality tests are applied on accuracy data of these three groups of segmentations separately. As the statistics of the data of the Morphological Snakes, the Random Walkers and the U-Net algorithms are 4805.37, 5763.48, and 3825.55 and all of their p values are 0.000, the null hypothesis that these data are from normal distributions can be rejected. In other words, the accuracies of segmentations from all three algorithms do not follow normal distribution. In addition, it can be seen that the accuracies of most non-zero data points are in the interval of $(0.8, 1]$.

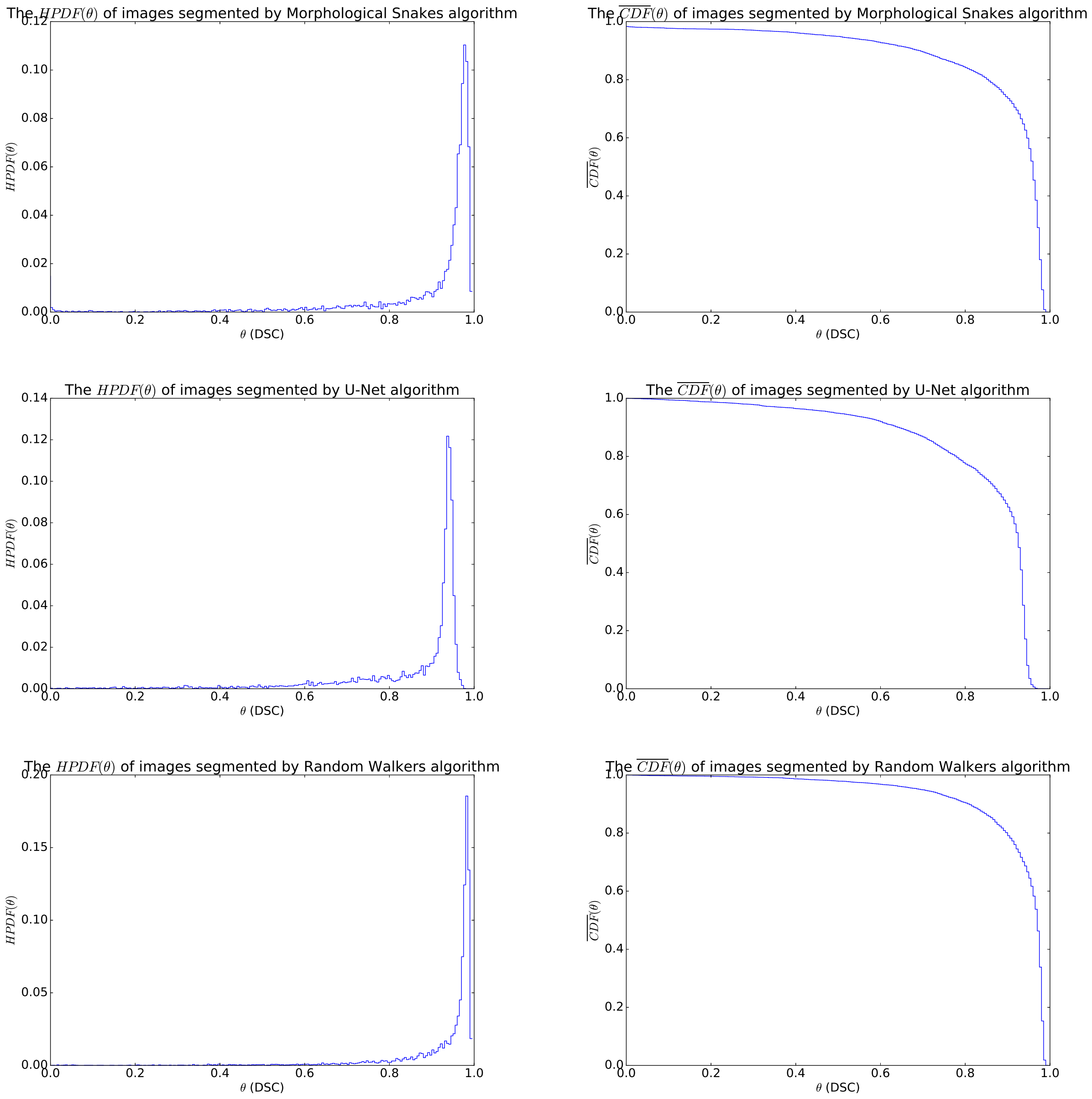


Figure 4.4: The $HPDF(\theta)$s (left) and $\overline{CDF}(\theta)$s (right) of three segmentation algorithms (for Leafsnap dataset).

As is shown in Figure 4.4, the aforementioned three algorithms have similarly shaped $HPDF(\theta)$s and $\overline{CDF}(\theta)$s. Starting from the point of $(0, 1)$, these three $\overline{CDF}(\theta)$ curves decrease slowly in the interval of $[0, 0.8]$ and decrease sharply in the interval of $(0.8, 1]$. All of these three curves end at the point $(1.0, 0)$. According to equation 4.4, in an interval I, the steeper the curve is, the greater the possibility that the segmentation algorithm can segment images with accuracies fall in the interval of I. Because of this, it can be easily known that the accuracies of most of segmentations generated by these three algorithms are in the interval of $(0.8, 1]$.

We also calculated the AUCs of these three algorithms. The AUCs of the Morphological Snakes algorithm, the U-Net algorithm, and the Random-Walkers algorithm are 0.884, 0.850, and 0.923. Therefore, the AUCs show that the Random Walkers has the best accuracy performance on the Leafsnap dataset and followed by the Morphological Snakes. The U-Net has the worst accuracy performance among these three segmentation algorithms.

Figure 4.5 shows the pairwise comparisons of the Morphological Snakes algorithm, the U-Net algorithm, and the Random Walkers algorithm basing on their $\overline{CDF}(\theta)$s. The red dotted lines mean that the pairwise differences of $\overline{CDF}(\theta)$s equals to 0, namely $\overline{CDF}_{diff}(\theta) = 0$.

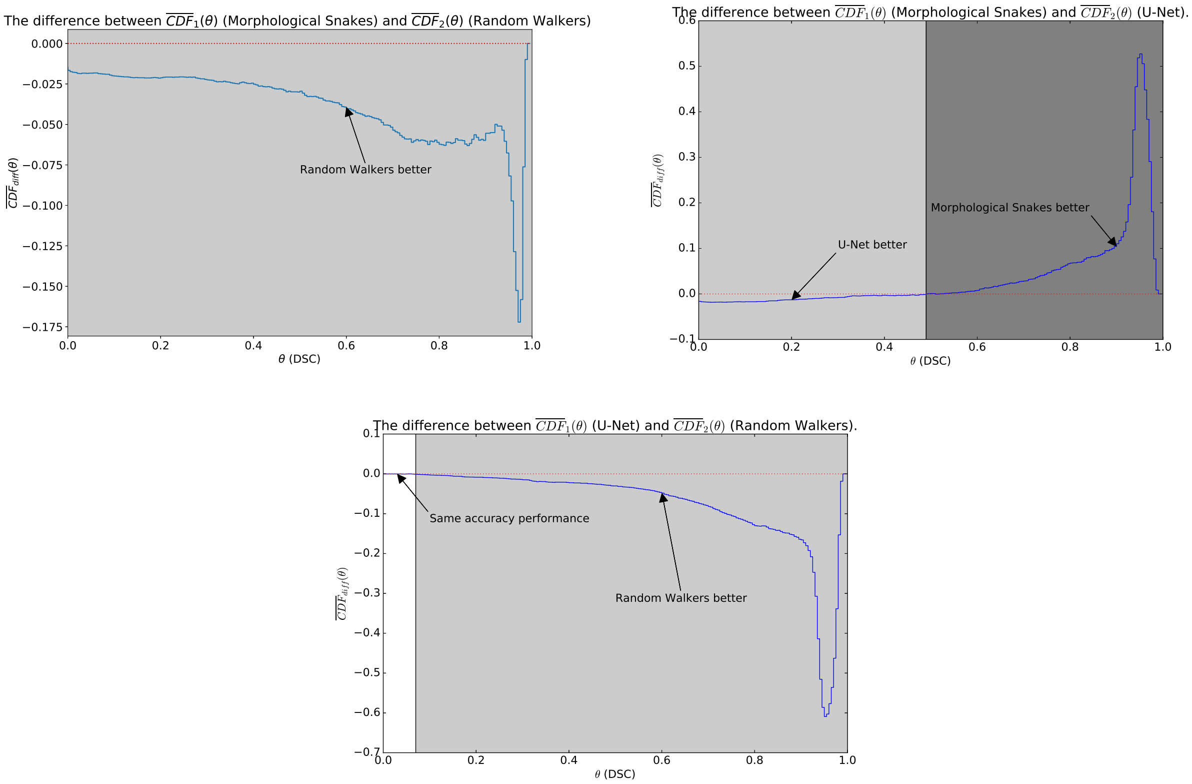


Figure 4.5: The pairwise $\overline{CDF}_{diff}(\theta)$s of three segmentation algorithms (for Leafsnap dataset). White regions: the two algorithms have the same accuracy performance. Light gray regions: the latter algorithm has better accuracy performance than the former algorithm. Dark gray regions: the latter algorithm has worse accuracy performance than the former algorithm.

According to equation 4.6, the accuracy performances of two algorithms (algorithm 1 and algorithm 2) can be compared using the value of $\overline{CDF}_{diff}(\theta)$ directly. If $\overline{CDF}_{diff}(\theta) > 0$, the possibility that the algorithm 1 can segment images with accuracy of θ or higher is greater than that of the algorithm 2. Therefore, the following results of the comparisons of these algorithms are obtained. We also show the results in Figure 4.5.

- **The comparison of the Morphological Snakes algorithm and the Random Walkers algo-**

rithm.

The $\overline{CDF}_{\mathit{diff}}(\theta)$ of Morphological Snakes and Random Walkers is below the red dotted line. In other words, when $\theta \in [0, 1]$, $\overline{CDF}_{\mathit{diff}}(\theta) < 0$. Therefore, for any $\theta \in [0, 1]$, the possibility that the Random Walkers algorithm can segment images with accuracy of θ or higher is greater than that of the Morphological Snakes algorithm. In other words, the Random Walkers algorithm is better. Because the Random Walkers algorithm is more likely to generate segmentations with high accuracies and it has lower chance of total failure.

- **The comparison of the Morphological Snakes algorithm and the U-Net algorithm.**

  The $\overline{CDF}_{\mathit{diff}}(\theta)$ of Morphological Snakes and U-Net is under the red dotted line when $\theta \in [0, 0.55]$ and it is above the red dotted line when $\theta \in (0.55, 1]$. In other words, when $\theta \in [0, 0.55]$, $\overline{CDF}_{\mathit{diff}}(\theta) < 0$ and when $\theta \in (0.55, 1]$, $\overline{CDF}_{\mathit{diff}}(\theta) > 0$. Therefore, for any $\theta \in [0, 0.55]$, the possibility that the U-Net algorithm can segment images with accuracy of θ or higher is greater than that of the Morphological Snakes algorithm. But for any $\theta \in [0.55, 1]$, the possibility that the Morphological Snakes algorithm can segment images with accuracy of θ or higher is greater than that of the U-Net algorithm. That is to say, the Morphological Snakes algorithm is more likely to generate segmentations with high accuracies, but it has slightly higher chance to generate segmentations with low accuracies.

- **The comparison of the U-Net algorithm and the Random Walkers algorithm.**

  The $\overline{CDF}_{\mathit{diff}}(\theta)$ of U-Net and Random Walkers, is overlapped with the red dotted line when $\theta \in [0, 0.065]$ and it is under the red dotted line when $\theta \in (0.065, 1]$. In other words, when $\theta \in [0, 0.065]$, $\overline{CDF}_{\mathit{diff}}(\theta) = 0$ and when $\theta \in (0.065, 1]$, $\overline{CDF}_{\mathit{diff}}(\theta) < 0$. Therefore, for any $\theta \in [0, 0.065]$, the possibility that the U-Net algorithm can segment images with accuracy of θ or higher is the same as that of the Random Walkers algorithm. But for any $\theta \in [0.065, 1]$, the possibility that the Random Walkers algorithm can segment images with accuracy of θ or higher is greater than that of the U-Net algorithm. In other words, the Random Walkers algorithms is more likely to generate segmentations with high accuracies and these two algorithm have about the same chance to generate totally wrong segmentations.

According to the values of $\overline{CDF}_{\mathit{diff}}(\theta)$s of these three algorithms, we prefer the Random Walkers algorithm. In addition, comparing with the U-Net algorithm, we prefer the Morphological Snakes algorithm, even if it has slightly higher chance (< 0.05) to generate segmentations with low accuracies than the U-Net algorithm, but at least we know this, which would be not possible with the standard method.

We can also use the *standard method* to analyze the accuracy performances of these three segmentation algorithms. The mean and standard deviation of the accuracies of segmentations generated by the Morphological Snakes algorithm, the U-Net algorithm, and the Random Walkers algorithm are calculated and the results are shown in Table 4.1.

Table 4.1: The means, standard deviations of DSCs of the Leafsnap segmentations generated by three algorithm.

Algorithms	Morphological Snakes	U-Net	Random Walkers
Mean	0.891	0.857	0.930
Standard deviation	0.188	0.162	0.121

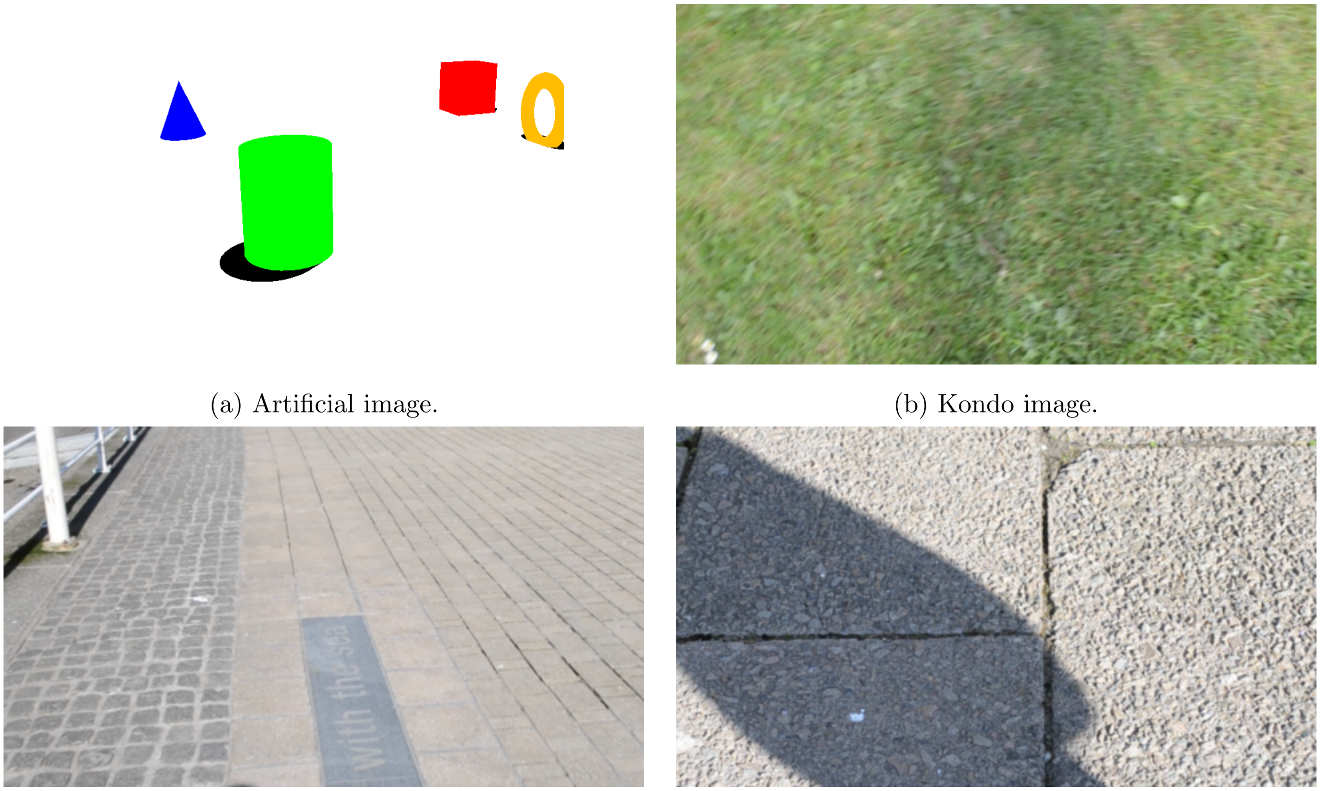
For the Leafsnap dataset, when applying Mann-Whitney U tests on these three groups of accuracy data, the results show that there are statistically significant differences in accuracy between the Morphological Snakes and the Random Walkers ($U = 22624202.5$ and $p = 0.000$), the Morphological Snakes and the U-Net ($U = 15245953.5$ and $p = 0.000$), and the Random Walkers and the U-Net ($U = 12125822.5$ and $p = 0.000$). From Table 4.1, we can see that the Random Walkers has the greatest mean and followed by the Morphological Snakes and the U-Net. So we can draw a conclusion using the *standard method* that the Random Walkers has the best accuracy performance on the Leafsnap dataset and followed by the Morphological Snakes. The U-Net has the worst accuracy performance among these three segmentation algorithms. This conclusion is the same as what we got using AUCs.

We can see the result of the standard method and result of using $\overline{CDF}(\theta)$s to analyze the accuracy performances of three segmentation algorithms are similar, except for some nuances. For example, the second plot of Figure 4.5 shows that the Morphological Snakes algorithm has a slightly higher chance of generating segmentations with low accuracies than the U-Net algorithm. The $\overline{CDF}_{diff}(\theta)$ of the Morphological Snakes algorithm and the U-Net algorithm is smaller than 0, which means for any $\theta \in [0, 0.55]$, the possibility that the U-Net algorithm can segment images with accuracy of θ or higher is greater than that of the Morphological Snakes algorithm. In short, the $\overline{CDF}(\theta)$s tell us that when $\theta \in [0, 0.55]$, the U-Net algorithm is better. But the Table 4.1 tells the different result that the Morphological Snakes algorithm is better than the U-Net algorithm, because the mean accuracy of the Morphological Snakes algorithm is greater than that of the U-Net algorithm. The Mann-Whitney U test also proves that there are statistically significant differences between these two algorithms.

The reason why the conclusions obtained by using *standard method* and by comparing $\overline{CDF}(\theta)$s of three algorithms are similar may be that the three $\overline{CDF}(\theta)$ curves have similar shapes on the Leafsnap dataset. But when the shapes of the curves of $\overline{CDF}(\theta)$s differ a lot, it is unclear whether we can draw similar conclusions using the standard method and comparing $\overline{CDF}(\theta)$s. So we tried on the another dataset. The process is shown in Example 2.

4.3.2 Example 2: the Shadow Dataset

The shadow detection and texture segmentation dataset for mobile robots (SDMR dataset) [57] contains some natural images and artificial images, as well as their binary ground truths. The backgrounds of the images in the SDMR dataset have textures or objects, such as tarmac and bricks. The foregrounds of the images in the SDMR dataset are shadows. Figure 4.6 gives some examples.



(a) Artificial image.

(b) Kondo image.

(c) Active image.

(d) Static image.

Figure 4.6: Example images in the SDMR dataset.

The SDMR dataset consists of 4 kinds of images: the artificial images, kondo images, active images, and static images. The artificial images contain many geometric shapes with different colors and their backgrounds are white. So they are easily segmented. The active images are generated from footage using an active camera, which is carried by a person walking or panning across the ground. The camera takes photos of the passing scenery. Similar to the active images, kondo images are captured from a webcam attached to a robot. The quality of the kondo images are very low, and the noise levels are considerable. Thus, the active images and kondo images are not suitable for this experiment. The static images are generated from footage using a static camera. The sequence of scenes that a person's shadow passing in front of a textured surface are captured as static images.

For this example, we selected 1282 static images from the SDMR dataset. All of these static images have unambiguous foregrounds. The example images are shown in Figure 4.7. In the ground truths, foreground (shadow) regions are marked in black and the background regions are in white. For simplicity, we refer to the selected static images and their ground truths as the shadow dataset.

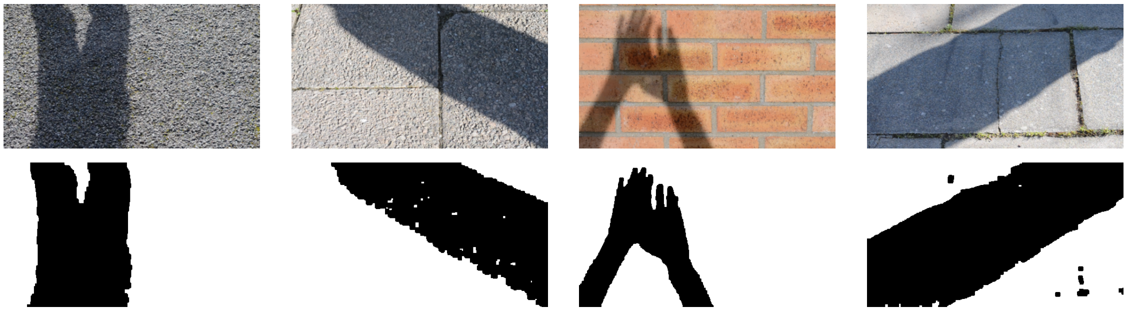


Figure 4.7: Example images and their ground truths in the shadow dataset. Top: original images, bottom: ground truths.

Similar to example 1, the same implementations of the the Morphological Snakes [2], the Random Walker [25], and the U-Net algorithms [1] are used to segment the original images in the shadow dataset. The accuracies of segmentations are also measured with the Dice similarity coefficient (DSC).

The $HPDF(\theta)$s and $\overline{CDF}(\theta)$s of the Morphological Snakes algorithm, the U-Net algorithm, and the Random Walkers algorithm are shown in Figure 4.8. Their difference $\overline{CDF}_{diff}(\theta)$s are shown in Figure 4.9. The mean DSCs of the segmentations generated by the implementations of the Morphological Snakes algorithm, the U-Net algorithm, and the Random Walkers algorithm are shown in table 4.3. Note that the number of bins of $HPDF(\theta)$s and $\overline{CDF}(\theta)$s are 100, therefore the width of each bin is 0.01.

As is shown in Figure 4.8, the accuracies of most segmentations are in the interval of $[0.9, 1]$, but the shapes of these three $HPDF(\theta)$s curves have no other obvious common features. D'Agostino-Pearson normality tests are applied on accuracy data of these three groups of segmentations separately. As the statistics of the data of the Morphological Snakes, the Random Walkers and the U-Net algorithms are 506.11, 1007.05, and 548.50 and their p values are all smaller than 0.001, the null hypothesis that these data are from normal distributions is rejected. In other words, the accuracies of all these three groups of segmentations do not follow normal distribution.

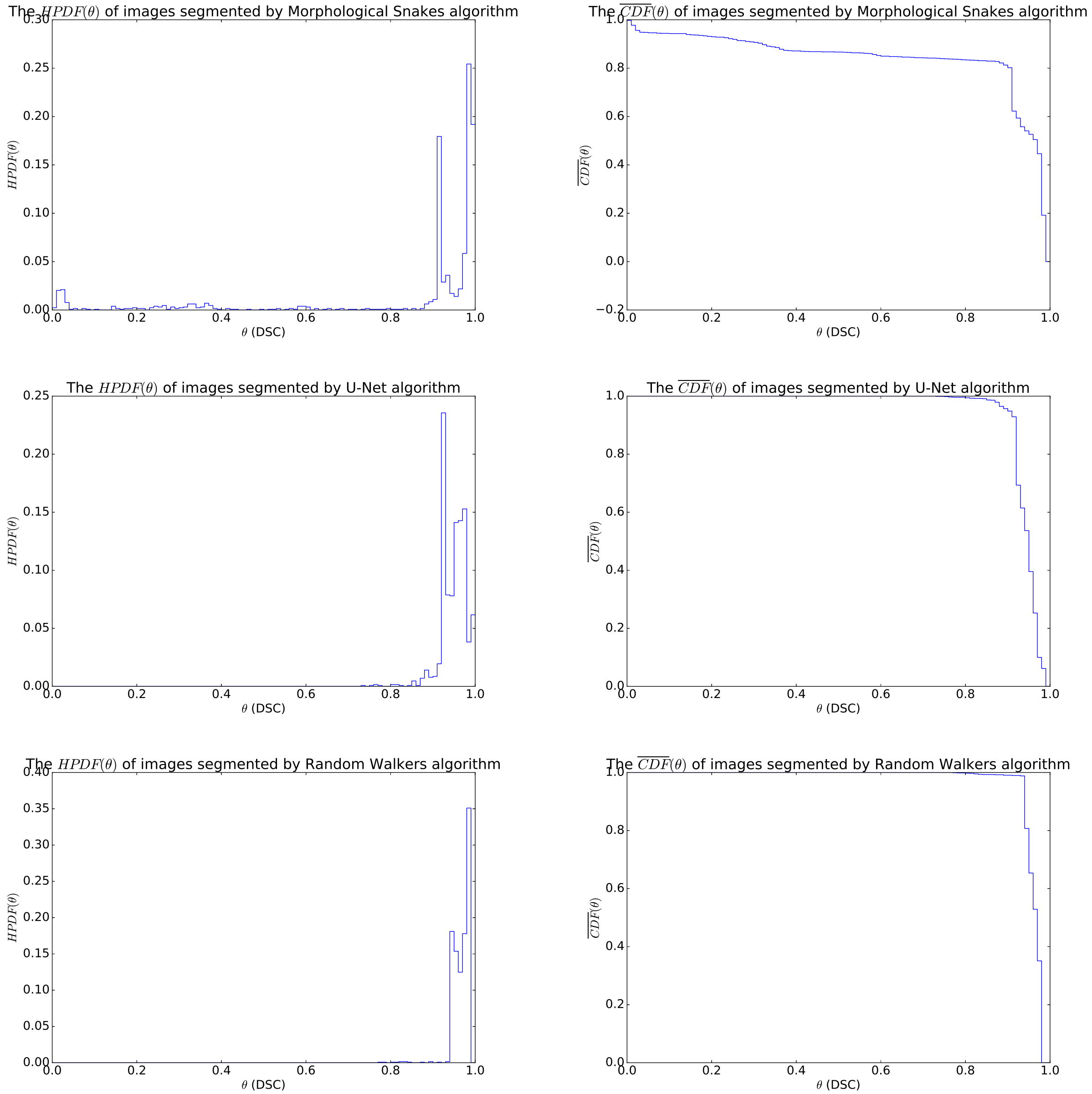


Figure 4.8: The $HPDF(\theta)$s (left) and $\overline{CDF}(\theta)$s (right) of three segmentation algorithms (for the shadow dataset).

The $\overline{CDF}(\theta)$ curves of segmentations generated by U-Net and Random Walkers have the similar shapes. In the interval of $[0, 0.7]$, the $\overline{CDF}(\theta)$s of these two algorithms are equal to 1, which means the accuracies of all the segmentations generated by these two algorithms are greater than 0.7. As these two curves decrease sharply in the interval of $[0.9, 1]$, the accuracies of most of the segmentations generated by these two algorithms are in the interval of $[0.9, 1]$.

When it comes to the Morphological Snakes algorithm, it's a little different. As the value of $\overline{CDF}(\theta)$ doesn't always equal to 1 in the interval of $[0, 0.8]$, the accuracies of some segmentations are smaller than

0.8. In the interval of $[0.9, 1]$, the curve decreases sharply as θ increases, meaning the accuracies of many segmentations generated by the Morphological Snakes algorithm fall in the interval of $[0.8, 1]$.

Figure 4.9 shows the pairwise comparisons of the Morphological Snakes algorithm, the U-Net algorithm, and the Random Walkers algorithm basing on their $\overline{CDF}(\theta)$s. The red dotted lines mean that the pairwise differences of $\overline{CDF}(\theta)$s equals to 0, namely $\overline{CDF}_{diff}(\theta) = 0$.

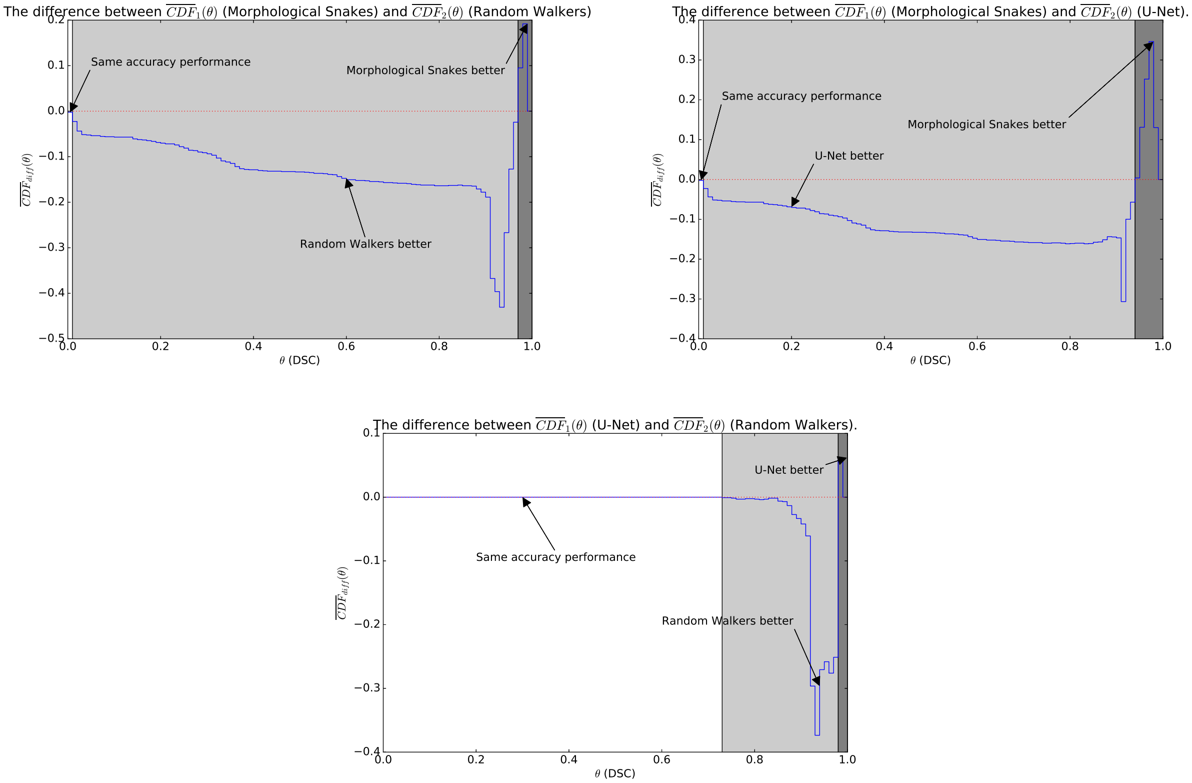


Figure 4.9: The pairwise $\overline{CDF}_{diff}(\theta)$s of three segmentation algorithms (for the shadow dataset). White regions: the two algorithms have the same accuracy performance. Light gray regions: the latter algorithm has better accuracy performance than the former algorithm. Dark gray regions: the latter algorithm has worse accuracy performance than the former algorithm.

As the results of comparisons are based on the values of $\overline{CDF}_{diff}(\theta)$s, for different θ, we may get different results. All the possible results are summarized in Table 4.2. We also marked the results of comparisons on Figure 4.9.

Table 4.2: The pairwise comparisons of the accuracy performances of three segmentation algorithms (MS: Morphological Snakes algorithm, RW: Random Walkers algorithm, UN: U-Net algorithm).

Algorithms	**MS and RW**	**MS and UN**	**UN and RW**
θ	[0, 0.01]	[0, 0.01]	[0, 0.73]
$\overline{CDF}_{diff}(\theta)$	=0	=0	=0
Conclusion	same performance	same performance	same performance
θ	(0.01, 0.97]	(0.01, 0.94]	(0.73, 0.98]
$\overline{CDF}_{diff}(\theta)$	<0	<0	<0
Conclusion	Random Walkers better	U-Net better	Random Walkers better
θ	(0.97, 1]	(0.94, 1]	(0.98, 1]
$\overline{CDF}_{diff}(\theta)$	>0	>0	>0
Conclusion	Morphological Snakes better	Morphological Snakes better	U-Net better

According to Table 4.2, for each pair of algorithms (denote them as algorithm A and algorithm B), there are three situations: 1. The accuracy performances of two algorithms are the same. 2. The accuracy performance of algorithm A is better than the accuracy performance of algorithm B. 3. The accuracy performance of algorithm B is better than the accuracy performance of algorithm A. The only difference is the ranges of accuracy intervals that we used to draw the conclusions.

Take the comparison of the accuracy performances of the Morphological Snakes and the Random Walkers algorithms for instance. The comparison result is shown in the second column of Table 4.2 and the first plot of Figure 4.9).

- For any accuracy $\theta \in [0, 0.01]$, the possibility that the Morphological Snakes algorithm can segment images with accuracy of θ or higher is the same as that of the Random Walkers algorithm. In other words, the two algorithms have the same chance of total failure.

- For any accuracy $\theta \in (0.01, 0.97]$, the possibility that the Random Walkers algorithm can segment images with accuracy of θ or higher is greater that of the Morphological Snakes algorithm.

- For any accuracy $\theta \in (0.97, 1]$, the possibility that the Morphological Snakes algorithm can segment images with accuracy of θ or higher is greater that of the Random Walkers algorithm. In other words, the Morphological Snakes algorithm is more likely to generate perfect segmentations.

When it comes to choosing an algorithm from the Morphological Snakes algorithm, the Random Walkers algorithm, and the U-Net algorithm basing on $\overline{CDF}_{diff}(\theta)$s, the results may vary. According to top-left subfigure of Figure 4.9, the Morphological Snakes algorithm has 0.18 higher chance than the Random Walkers algorithm to generate segmentations with really high accuracy (> 0.98). Meanwhile, we have to suffer from the fact that the Morphological Snakes algorithm is less likely to generate segmentations which are not perfect, but acceptable. Especially, the Morphological Snakes algorithm has 0.43 lower chance than the Random Walkers algorithm to generate segmentations with accuracies of 0.95 or higher. Therefore, we have to make tradeoffs when choosing an algorithm from the Morphological Snakes algorithm and the Random Walkers algorithm. The part of $\overline{CDF}_{diff}(\theta)$ of segmentation algorithms which is smaller than 0.8 also matters, because it indicates which algorihtm is more likely to give catastrophic fialure. Morphological Snakes algorithm might appear to be better, but it is more likely to be really bad at the expense of having fewer really good segmentations. And it is similar when we need to choose one from the Morphological Snakes algorithm and the U-Net algorithm, or the U-Net algorithm and the Random Walkers algorithm.

Similar to example 1, for the shadow dataset, we apply the standard method on the accuracy data generated by the Morphological Snakes algorithm, the U-Net algorithm, and the Random Walkers algorithm. The means and standard deviations of these three algorithms are shown in Table 4.3.

Table 4.3: The means, standard deviations and ranks of DSCs of the segmentations generated by three algorithm (for the shadow dataset).

Algorithms	Morphological Snakes	U-Net	Random Walkers
Mean	0.858	0.959	0.978
Standard deviation	0.275	0.029	0.020

For the shadow dataset, the means and standard deviations of the segmentations generated by the implementations of three segmentation algorithms differ a lot. When applying Mann-Whitney U tests on these three groups of accuracy data, the results show that there are statistically significant differences between the Morphological Snakes and the Random Walkers ($U = 783708.0$ and $p = 0.021$), the Morphological Snakes and the U-Net ($U = 771049.0$ and $p = 0.003$), and the Random Walkers and the U-Net ($U = 458640.0$ and $p = 0.000$). According to Table 4.3, the mean accuracy of Random Walkers is the greatest and then the U-Net. The mean accuracy of the Morphological Snakes algorithm is the smallest. Therefore, the standard method shows that the Random Walkers has the best accuracy performance on the shadow dataset and followed by the U-Net. The Morphological Snakes algorithm has the worst accuracy performance among these three segmentation algorithms.

In example 1, the three $\overline{CDF}(\theta)$ curves have similar shapes. But In this example, the shapes of the three $\overline{CDF}(\theta)$ curves are different. This might be the reason why we get similar results in example 1 and we get different results in this example when using the *standard method* and the differences of $\overline{CDF}(\theta)$s to analyze the accuracy performances of the Morphological Snakes algorithm, the U-Net algorithm and the Random Walkers algorithms.

4.4 Summary

$\overline{CDF}(\theta)$ can be used to characterize the accuracy performances of segmentation algorithms and the differences of the $\overline{CDF}(\theta)$s of segmentation algorithms can be used to compare their accuracy performances. Different from the *standard method* that using mean and standard deviation of accuracies of segmentations to evaluate and compare the overall accuracy performances of segmentation algorithms, the $\overline{CDF}(\theta)$ focuses on the detailed accuracy performance, namely the possibility that the segmentation algorithm can generate segmentations with specific accuracies or higher. In addition, $\overline{CDF}(\theta)$ can characterize and compare segmentation algorithms no matter whether the accuracies of segmentations are normal-distributed or not. The drawback of the proposed distribution-based method is that the θs should be chosen manually when comparing segmentation algorithms, which may lead to different conclusions and thus make the result to be very complex.

From example 1 and example 2, we can see that the distribution of these three segmentation algorithms are different. As the HPDFs are sampled from the underlying CDFs of the segmentation algorithm, we may see that HPDF and CDF of the same segmentation algorithm my be different when we use them to segment images in different dataset, which means that the same segmentation algorithm will behave differently. In other word, when we use other datasets to analyze the acccuracy performance of the aforementioned three segmentation algorithms, we may get different decision on which algorithm to be used to do the segmentation work for that specific dataset.

In the following chapter, an extensive method for ranking the accuracies of segmentation algorithms will be illustrated basing on the $\overline{CDF}(\theta)$ of segmentation algorithms. Taking the distribution factors into account, the extensive method to be proposed can give an overall measurement of the accuracy performances of segmentation algorithms. At the same time, it can avoid the drawbacks of the needs of choosing θs manually and corresponding different results.

5. An Extensive Method for Ranking the Accuracy Performance of Segmentation Algorithms

According to chapter 4, although $\overline{CDF}_{diff}(\theta)$ can be used to compare the accuracy performances of image segmentation algorithms, the testing result may vary because of the situations where we need to make trade-offs. In order to simplify the evaluation process, as well as making the result more convincing, some modifications should be done. In this chapter, an extensive method is proposed for ranking the accuracy performances of different segmentation algorithms basing on $\overline{CDF}(\theta)$.

5.1 Comprehensive Method

People usually rank segmentation algorithms based on their mean accuracies. Similar to the comparison of two segmentation algorithms, when using the mean accuracies, an assumption is made that these accuracies are normally distributed. Therefore it may be unreasonable to rank segmentation algorithms using mean accuracies alone without learning the distribution of these accuracies. We claim that segmentation algorithms should be ranked using mean accuracy, the area under the $\overline{CDF}(\theta)$ curve (AUC), and accuracy thresholds θ_δs.

The $\overline{CDF}(\theta)$ and AUC were introduced in section 4, the details of θ_δs are as follow. If $\overline{CDF}(\theta) = \delta$, we can interpret it as " The probability that an algorithm can generate segmentations with accuracies of θ or higher is δ". Given a specific δ, we would like to find the largest θ for which $\overline{CDF}(\theta) \geq \delta$, and denote this as θ_δ. Therefore, the threshold θ_δ can be expressed as

$$\theta_\delta = \max\left\{\theta | \overline{CDF}(\theta) \geq \delta\right\}. \tag{5.1}$$

Figure 5.1 shows an example of how to find the threshold θ_δ giving a specific likelihood $\delta = 15\%$ and the $\overline{CDF}(\theta)$ of a group of synthetic data. At first, we can draw a horizontal line, which means $\delta = 15\%$. Therefore, at the intersection of the horizontal line and the $\overline{CDF}(\theta)$ curve, $\overline{CDF}(\theta) = \delta = 15\%$. Then we can draw a vertical line crossing the intersection of the line of $\delta = 15\%$ and the $\overline{CDF}(\theta)$ curve. The x-coordinate of the intersection of the vertical line and the x-axis is the θ_δ we are looking for.

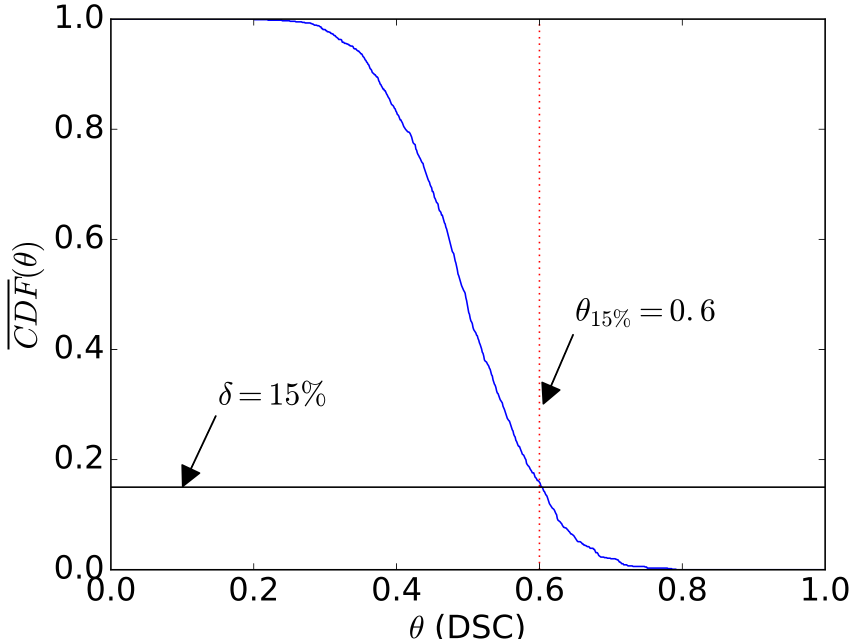


Figure 5.1: The illustration of how to find θ_δ when given a specific likelihood δ and a $\overline{CDF}(\theta)$ curve. The blue solid curve is the $\overline{CDF}(\theta)$ of a group of Gaussian-distributed data (number of samples is 1000, $\mu = 0.5$, $\sigma = 0.1$). The black solid line is the likelihood $\delta = 15\%$. The red dotted line is the threshold $\theta_{15\%}$.

Given a likelihood δ, we can rank the accuracy performances of segmentation algorithms according to their respective θ_δs. Suppose there are two segmentation algorithms and we denote them as algorithm A and algorithm B. If the likelihood δ takes a specific value, such as $\delta = 15\%$, we can calculate the θ_δs of algorithm A and algorithm B according to equation 5.1. So we get $\theta_{\delta A} = \max\left\{\theta | \overline{CDF}_A(\theta) \geq 15\%\right\}$ and $\theta_{\delta B} = \max\left\{\theta | \overline{CDF}_B(\theta) \geq 15\%\right\}$. It can be interpreted as that the algorithm A is 15% likely to generate segmentations with accuracies of $\theta_{\delta A}$ or higher and algorithm B is 15% likely to generate segmentations with accuracies of $\theta_{\delta B}$ or higher. If $\theta_{\delta A} > \theta_{\delta B}$, the algorithm A is better than algorithm B. Because if the likelihood (δ) that an segmentation algorithm can generate some segmentations whose accuracies are no smaller than a specific accuracy threshold (θ_δ), we hope this threshold to be as big as possible.

In order to characterize the detailed information of the distribution of the segmentations generated by a segmentation algorithm, we need to calculate θ_δs using different δs. In this thesis, the δs we are going to use are from the set of $\{5\%, 15\%, 25\%, 35\%, 45\%, 55\%, 65\%, 75\%, 85\%, 95\%\}$.

The extensive method can be used to rank segmentation algorithms in three steps. First, the ranks of all segmentations will be calculated using different evaluation metrics, including the mean of DSCs, the area under the $\overline{CDF}(\theta)$ curve (AUC), the thresholds θ_δs (δ takes different values). A superior algorithm will have a greater mean DSC, a greater AUC, greater θ_δs. The better the accuracy performance of an algorithm, the smaller the rank of this algorithm. Second, for each segmentation algorithm, the ranks calculated with different metrics each will be summed up. Third, the summations of the ranks calculated in the second step will be re-ranked. The smaller the summation, the smaller the new rank, and therefore the accuracy performance of the corresponding segmentation algorithm is better.

5.2 An Example of Ranking Segmentation Algorithms Using the Comprehensive Method

5.2.1 Dataset

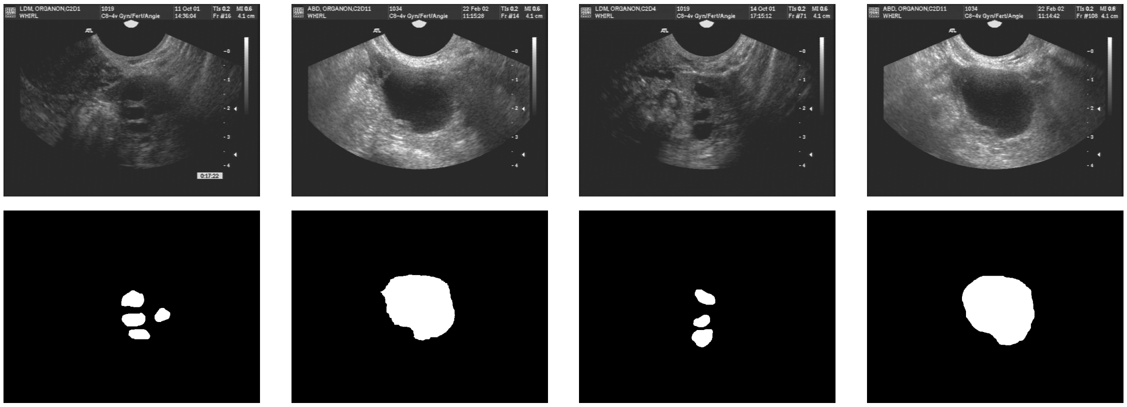


Figure 5.2: Example images in the HOF dataset. Top: original images, bottom: ground truths.

The dataset used in this chapter is from Haque [28]. Haque used the image dataset of R. Pierson et al. [7] to develop simulated interactive models, which contains human ovarian follicles (HOF) images. Example images and their corresponding ground truths are shown in Figure 5.2.

The HOF dataset contains 32 ultrasound images. The size of each image is 640 × 480 pixels and the maximum number of follicles in any image is 14. The diameters of all the follicles shown in the ground truth images are larger than 2.5mm. There may be some follicles whose diameters are smaller than 2.5mm, but they are not shown in the ground truth segmentations, because it is difficult for humans to identify follicles smaller than 2.5mm. Manually delineated ground truth segmentations of these follicles are provided by a highly experienced human operator.

5.2.2 Image Segmentation Algorithms

Eight semi-automatic segmentation algorithms are used in [28]. These semi-automatic segmentation algorithms are representative of different segmentation theories and three types of interaction modes. The detail information is shown in Table 5.1.

Table 5.1: List of the algorithms and interaction modes used in [28].

Name of the algorithm	Interaction mode	Acronym
Graphcut with Star Shape Prior [79]	brush stroke	GCBS
Geodesic Star Convexity [27]	brush stroke	GSC
Sequential Geodesic Star Convexity [27]	brush stroke	GSCSeq
Trust Region Convexity [24]	brush stroke	TRC
Onecut [76]	brush stroke	Onecut
Distance Regularized Level Set Evolution [47]	closed contour	DRLSE
Distance Regularized Level Set Evolution [47]	closed iso-contour	DRLSEIC
Graphcut without Star Shape Prior [12]	seed point	GCnoSP

These semi-automatic segmentation algorithms are based on five kinds of segmentation theories, namely the Graph Cut [12] theory, the Level Set [47] theory, the Geodesic Star Convexity [27] theory, the Onecut [76] theory, and the Trust Region Convexity [24] theory.

5.2.3 Interaction Modes

Hanque synthetically generated several kinds of interactions including brush stroke, closed contour, closed iso-contour and seed point and used these synthetic interactions to segment the images in the HOF dataset. These synthetic interactions can be classified as peripheral, intermediate and central according to the distance of the interactions from the centroid of the follicle. These three groups of interactions and images in the HOF dataset are used as input of the 8 segmentation algorithm mentioned in Table 5.1. The definitions of different kinds of interactions as well as the classification of their locations are as follow.

The seed point interactions can be classified into three groups basing on a and b, which represent the distance between the seed point and the nearest boundary point of the follicle, and the distance between the seed point and the centroid of the follicle. According to equation 5.2, the classes of all interactions can be determined.

$$location = \begin{cases} peripheral, & \frac{a}{a+b} \leq \frac{1}{3} \\ intermediate, & \frac{1}{3} < \frac{a}{a+b} \leq \frac{2}{3} \\ central, & \frac{a}{a+b} > \frac{2}{3} \end{cases} \tag{5.2}$$

The curved brush strokes are segments of iso-contours of the distance transform of the foreground regions and roughly parallel to the foreground boundaries. Their widths vary from 3 to 5 pixels are most often observed in published works. The curved brush stroke interactions can be classified into two or three groups basing on the diameter of follicles. For those follicles whose diameters are greater than 3.75mm, the curved brush strokes are classified into the peripheral group, the intermediate group and the central group according to the distance between the strokes and the central of follicles. For those follicles whose diameters are between 2.5mm and 3.75mm, the curved brush strokes are classified into the central group and the peripheral group.

The straight brush stroke interactions are randomly generated in the foreground regions. They are segments of straight lines, whose widths vary between 3 and 5 pixels and lengths vary in a range, which is related to the size of foreground regions. It can be classified into two or three groups basing on the diameter of follicles. The categories of all pixel of a straight brush stroke will be determined using equation 5.2 at first. Then the majority votes of the categories will be seemed as the category of this straight brush stroke interaction.

The closed contour interactions have two kinds of shapes. One is elliptical, whose shape is independent to the foreground regions. The other one is iso-contours, whose shape is roughly parallel to the foreground boundaries. The location of closed contour interactions can be classified using the same method as brush strokes. For the iso-contour interactions, the mean distance of all pixels of the curve from the boundary of the follicles is calculated. The location class of the iso-contour interactions can be determined using equation 5.2.

5.2.4 Experiment Design

We are going to rank different segmentation algorithms using the proposed extensive method. The accuracy data generated by different segmentation algorithms is from Haque [28]. In [28], Haque segmented the HOF images using synthetic interactions, which have different shapes and different locations. In this experiment, we only rank the accuracy performances of the GCBS, GSC, GSCSeq, Onecut, and the TRC algorithm. This is because the all these algorithms used the brush stroke interactions to segment the HOF images. When the interactions have the same location, the only factor which affect the accuracies of segmentations is the segmentation algorithm.

In order to control the variables that affect the accuracies of segmentations, we classified the accuracy data into 4 groups according to the locations of interactions (the peripheral group, the intermediate group, the central group, and the group which contain the interactions at all kinds of locations). For each group of accuracies, we apply the proposed extensive method and rank the accuracy performance of the GCBS, GSC, GSCSeq, Onecut, and the TRC algorithm.

People usually ranks segmentation algorithms with their mean accuracies and it is believed that the algorithms which have higher mean accuracies are better. Therefore, we also ranks the aforementioned five segmentation algorithms with the mean accuracies (DSCs) alone and compare it with the result that we obtain using the proposed extensive method.

5.2.5 Experiment Results

In order to reveal the underlying probability density functions of the GCBS, GSC, GSCSeq, Onecut, and the TRC algorithm, we first calculate the $HPDF(\theta)$s of DSCs of the segmentations generated by the 5 semi-automatic segmentation algorithms using different interactions. The results are shown in Figure 5.3, Figure 5.4, Figure 5.5, and Figure 5.6. For each algorithm, there are about 1000 segmentations generated using interactions at different locations, the bin numbers of $HPDF(\theta)$ are 100, therefore the width of each bin is 0.01.

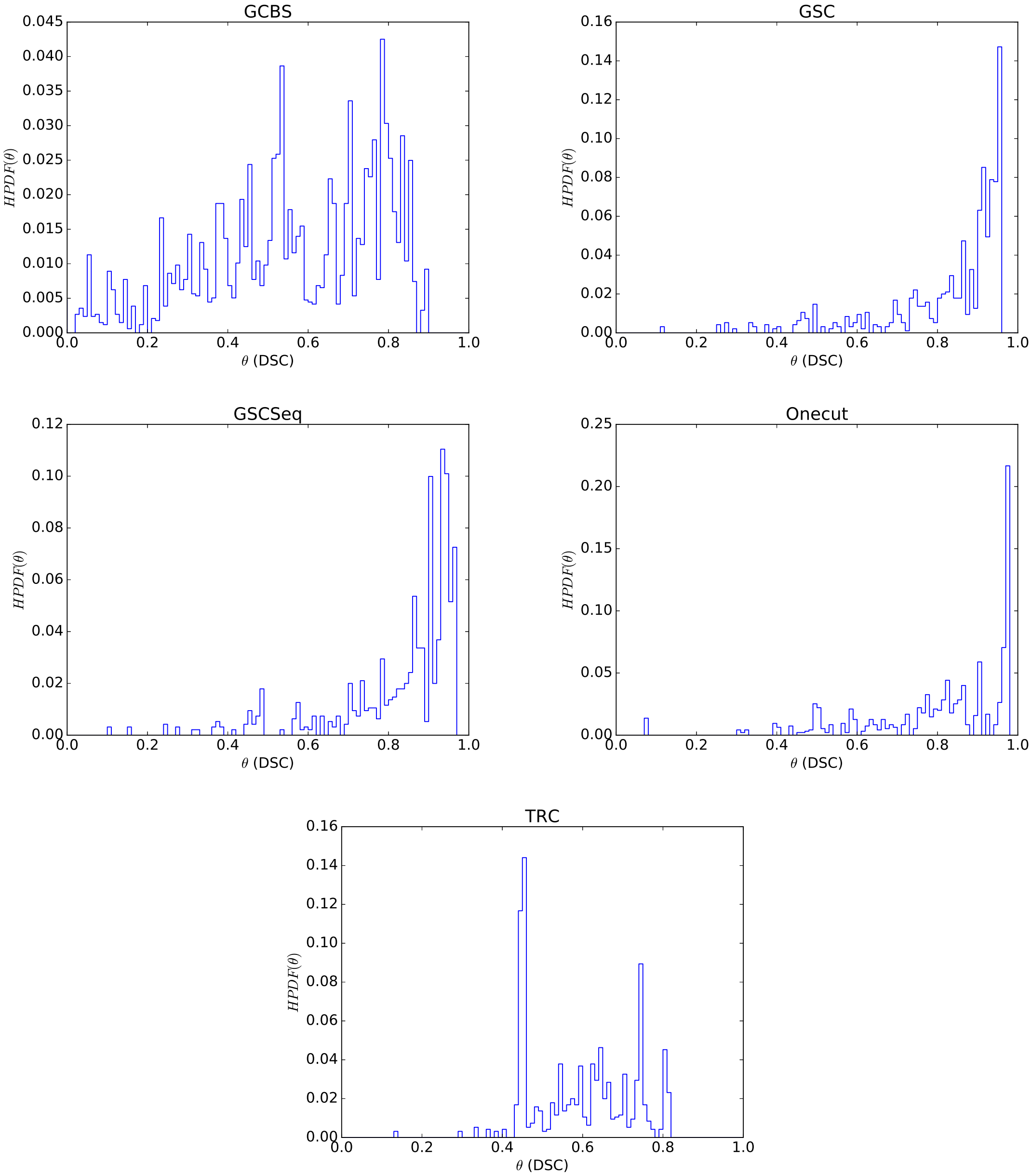


Figure 5.3: The $HPDF(\theta)$s of DSCs of the segmentations generated by 5 semi-automatic segmentation algorithms using central interactions.

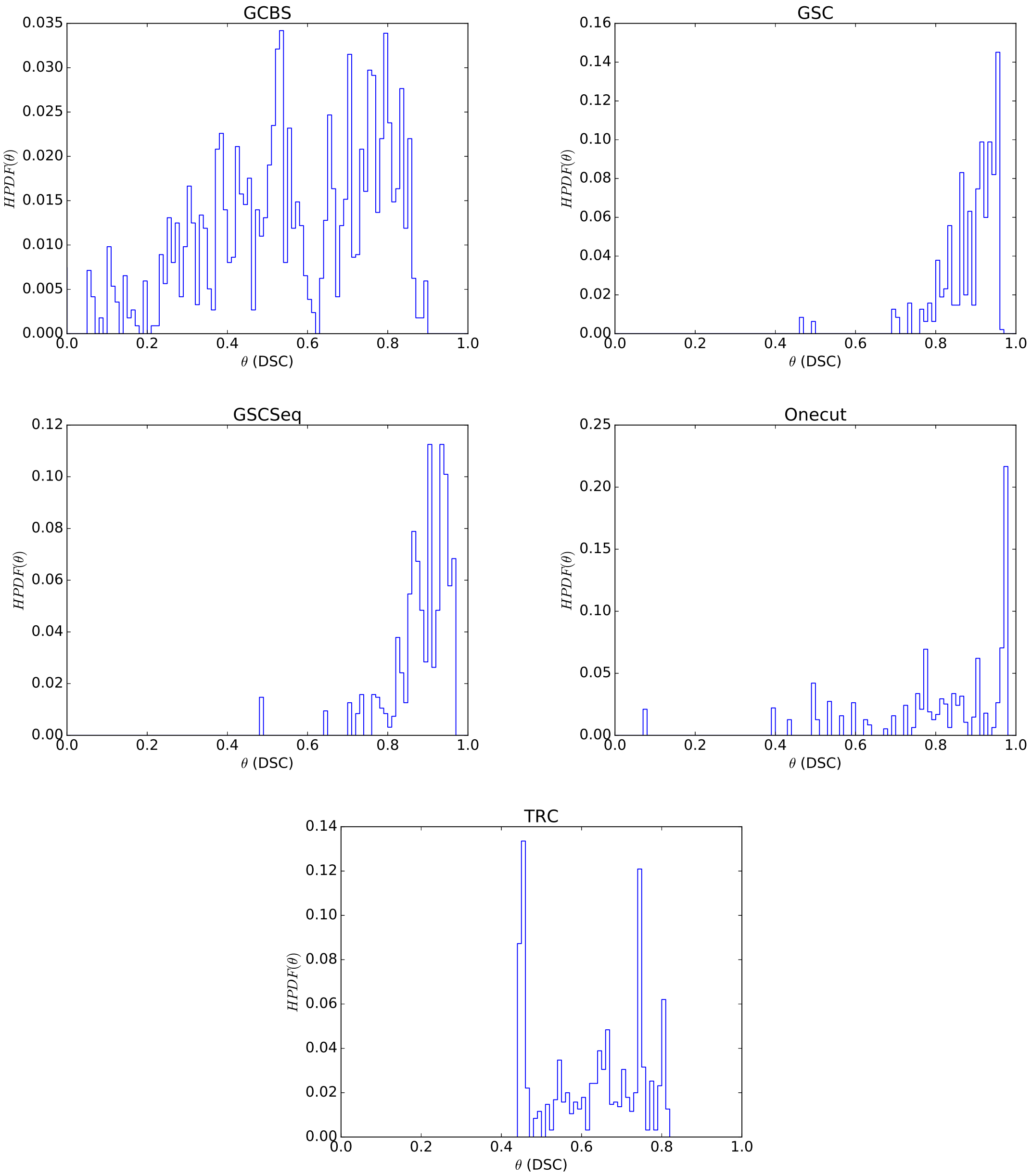


Figure 5.4: The $HPDF(\theta)$s of DSCs of the segmentations generated by 5 semi-automatic segmentation algorithms using intermediate interactions.

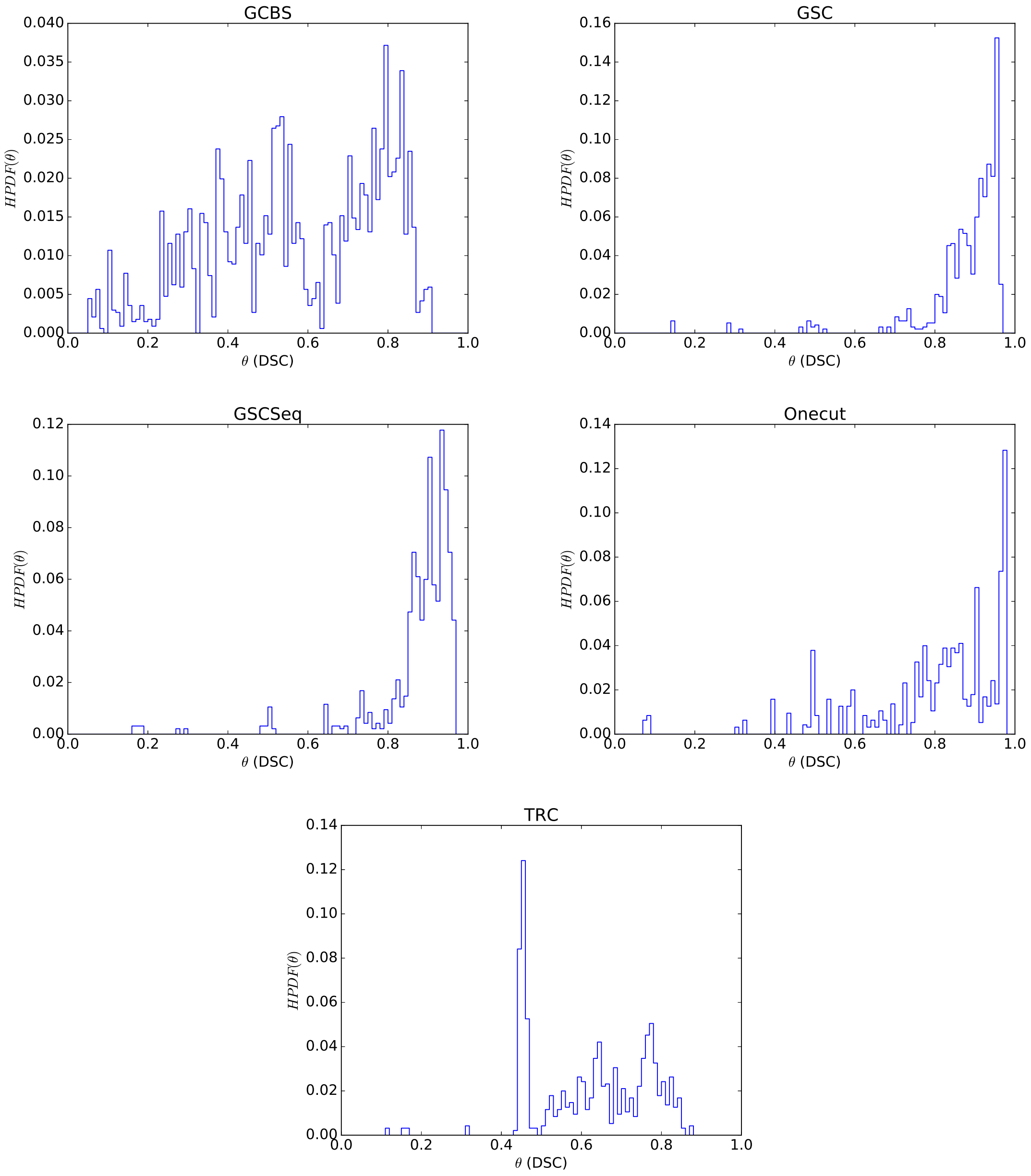


Figure 5.5: The $HPDF(\theta)$s of DSCs of the segmentations generated by 5 semi-automatic segmentation algorithms using peripheral interactions.

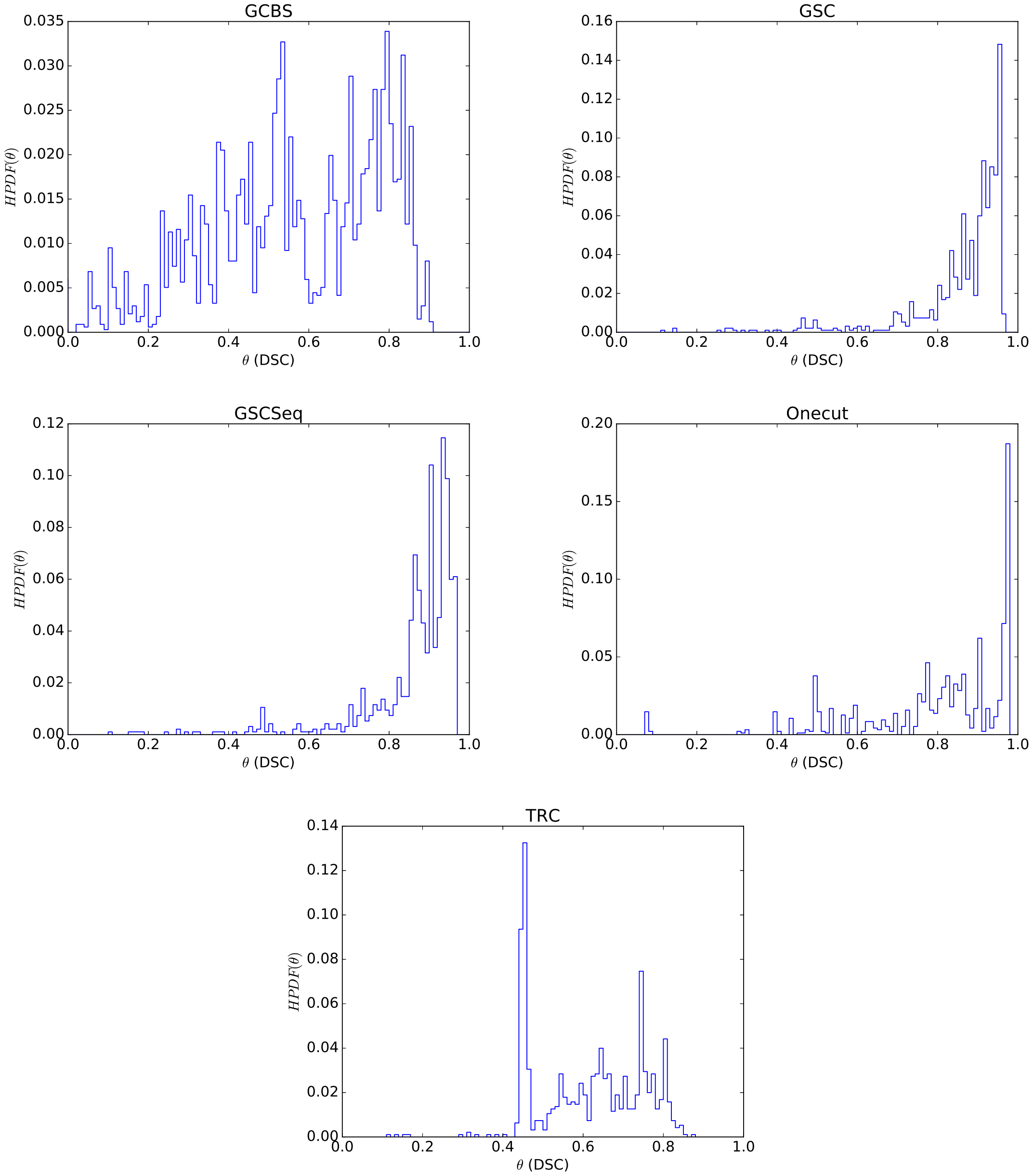


Figure 5.6: The $HPDF(\theta)$s of DSCs of the segmentations generated by 5 semi-automatic segmentation algorithms using interactions at all kinds of locations.

From Figure 5.3 to Figure 5.6, we can see for each semi-automatic segmentation algorithm, the $HPDF(\theta)$s of DSCs generated with different kinds of interactions have similar shapes. So we can refer to the Figure 5.6 when analyzing the shapes of $HPDF(\theta)$s of these segmentation algorithms.

In general, these five algorithms can be classified into 3 groups basing on the shapes of their $HPDF(\theta)$s. The first group of algorithm is the GCBS algorithm. Its $HPDF(\theta)$ spreads over the interval $[0, 1]$. The second group of algorithms are the GSC, the GSCSeq, and the Onecut algorithm. As θ increases from 0 to 1, the increment speeds of $HPDF(\theta)$s of these three algorithms are getting faster and faster. In addition, most of the DSCs of the segmentations generated by the GSC, the GSCSeq, and the Onecut algorithm are greater than 0.8. The third group of algorithm is the TRC algorithm. The DSCs of the segmentations generated by the TRC algorithm are mostly in the interval of $[0.4, 0.85]$. Intuitively, the second group of algorithms have better accuracy performances than the others, because most of their segmentations' DSC values are close to 1.

For each pair of algorithms, we applied Mann-Whitney U Tests on the accuracies of their segmentations and the results are shown in Table 5.2. Then, the mean of DSCs, the AUC of the $\overline{CDF}(\theta)$s, the θ_δs, and the final ranks of the summation of mean DSC, AUC, and θ_δs of these semi-automatic segmentations are calculated. It should be noted that we also ranked these algorithms basing on the summation of θ_δs. The results are shown in Table 5.3, Table 5.4, Table 5.5 and Table 5.6.

Table 5.2: The p-values of pairwise Mann-Whitney U Tests on the segmentation data generated by the GCBS, GSC, GSCSeq, Onecut, and TRC algorithm using different interactions. For each pair of algorithms, four p-values are calculated using the interactions at all locations (top-left), central interactions (top-right), intermediate interactions (bottom-left), peripheral interactions (bottom right). The p-values which are greater than 0.01 are marked in red.

	GCBS		GSC		GSCSeq		Onecut		TRC	
GCBS			0.0	7.1×10^{-260}	0.0	9.4×10^{-269}	4.8×10^{-193}	4.2×10^{-196}	1.7×10^{-6}	0.123
			0.0	0.0	0.0	0.0	2.2×10^{-196}	6.8×10^{-191}	1.2×10^{-11}	6.9×10^{-9}
GSC	0.0	7.1×10^{-260}			0.168	0.311	1.1×10^{-8}	0.283	5.9×10^{-232}	1.5×10^{-184}
	0.0	0.0			0.120	0.091	3.2×10^{-11}	3.9×10^{-24}	2.6×10^{-278}	3.9×10^{-253}
GSCSeq	0.0	9.4×10^{-269}	0.168	0.311			3.1×10^{-8}	0.295	6.0×10^{-233}	1.9×10^{-190}
	0.0	0.0	0.120	0.091			1.9×10^{-9}	1.8×10^{-23}	4.9×10^{-273}	1.3×10^{-249}
Onecut	4.8×10^{-193}	4.2×10^{-196}	1.1×10^{-8}	0.283	3.1×10^{-8}	0.295			8.8×10^{-128}	3.5×10^{-145}
	2.2×10^{-196}	6.8×10^{-191}	3.2×10^{-11}	3.9×10^{-24}	1.9×10^{-9}	1.8×10^{-23}			4.1×10^{-122}	5.4×10^{-121}
TRC	1.7×10^{-6}	0.123	5.9×10^{-232}	1.5×10^{-184}	6.0×10^{-233}	1.9×10^{-190}	8.8×10^{-128}	3.5×10^{-145}		
	1.2×10^{-11}	6.9×10^{-9}	2.6×10^{-278}	3.9×10^{-253}	4.9×10^{-273}	1.3×10^{-249}	4.1×10^{-122}	5.4×10^{-121}		

For each segmentation algorithm, its ranks in the Table 5.3, Table 5.4, Table 5.5 and Table 5.6 are the similar, except for the GSC algorithm and the GSCSeq algorithm. According to Table 5.2, when applying Mann-Whitney U tests on the segmentation data generated by the GSC algorithm and the GSCSeq algorithm, the results show that there are no statistical significant differences between the accuracies of segmentations generated with central interactions ($p = 0.311$). We can get the similar result when we test the GSC algorithm and the GSCSeq algorithm using the intermediate interactions ($p = 0.120$), peripheral interactions ($p = 0.091$) and the interactions at all locations ($p = 0.168$). Therefore, we can conclude that there are no significant difference between the accuracy performance of the GSC algorithm and the accuracy performance of the GSCSeq algorithm. Thus, we can attribute the reason why the ranks of the GSC algorithm and the GSCSeq algorithm in different tables are different to that it is so hard to rank these two segmentation algorithms because of their similar accuracies performances.

Table 5.3: The rank of the accuracy performances of 5 semi-automatic segmentation algorithms (using central interactions).

algorithm	GCBS	GSC	GSCSeq	Onecut	TRC
mean DSC	0.5722	0.8394	0.8413	0.8118	0.6024
rank	5	2	1	3	4
AUC	0.5616	0.8249	0.8283	0.7988	0.591
rank	5	2	1	3	4
$\theta_{5\%}$	0.85	0.96	0.97	0.98	0.81
rank	4	3	2	1	5
$\theta_{15\%}$	0.8	0.95	0.95	0.98	0.75
rank	4	2	2	1	5
$\theta_{25\%}$	0.76	0.94	0.94	0.97	0.71
rank	4	2	2	1	5
$\theta_{35\%}$	0.71	0.93	0.93	0.91	0.65
rank	4	1	1	3	5
$\theta_{45\%}$	0.64	0.91	0.91	0.87	0.63
rank	4	1	1	3	5
$\theta_{55\%}$	0.54	0.88	0.88	0.83	0.58
rank	5	1	1	3	4
$\theta_{65\%}$	0.5	0.84	0.85	0.79	0.52
rank	5	2	1	3	4
$\theta_{75\%}$	0.41	0.78	0.79	0.73	0.46
rank	5	2	1	3	4
$\theta_{85\%}$	0.31	0.7	0.71	0.59	0.45
rank	5	2	1	3	4
$\theta_{95\%}$	0.13	0.47	0.48	0.47	0.45
rank	5	2	1	2	4
summation (mean DSC, AUC, and θ_{δ}s)	55	22	15	29	53
rank	5	2	1	3	4
summation (θ_{δ}s only)	45	18	13	23	45
rank	4	2	1	3	4

Table 5.4: The rank of the accuracy performances of 5 semi-automatic segmentation algorithms (using intermediate interactions).

algorithm	GCBS	GSC	GSCSeq	Onecut	TRC
mean DSC	0.5702	0.8954	0.8928	0.8075	0.6348
rank	5	1	2	3	4
AUC	0.5596	0.8803	0.8796	0.7947	0.6234
rank	5	1	2	3	4
$\theta_{5\%}$	0.84	0.96	0.97	0.98	0.81
rank	4	3	2	1	5
$\theta_{15\%}$	0.8	0.95	0.95	0.98	0.76
rank	4	2	2	1	5
$\theta_{25\%}$	0.76	0.94	0.94	0.97	0.75
rank	4	2	2	1	5
$\theta_{35\%}$	0.7	0.93	0.93	0.91	0.71
rank	5	1	1	3	4
$\theta_{45\%}$	0.6	0.92	0.91	0.87	0.67
rank	5	1	2	3	4
$\theta_{55\%}$	0.54	0.91	0.9	0.82	0.63
rank	5	1	2	3	4
$\theta_{65\%}$	0.49	0.88	0.88	0.78	0.57
rank	5	1	1	3	4
$\theta_{75\%}$	0.41	0.86	0.86	0.75	0.49
rank	5	1	1	3	4
$\theta_{85\%}$	0.32	0.83	0.83	0.57	0.46
rank	5	1	1	3	4
$\theta_{95\%}$	0.17	0.74	0.74	0.44	0.45
rank	5	1	1	4	3
summation (mean DSC, AUC, and θ_δs)	57	16	19	31	50
rank	5	1	2	3	4
summation (θ_δs only)	47	14	15	25	42
rank	5	1	2	3	4

Table 5.5: The rank of the accuracy performances of 5 semi-automatic segmentation algorithms (using peripheral interactions).

algorithm	GCBS	GSC	GSCSeq	Onecut	TRC
mean DSC	0.5768	0.8887	0.8858	0.8063	0.6334
rank	5	1	2	3	4
AUC	0.5662	0.8742	0.8721	0.7934	0.6225
rank	5	1	2	3	4
$\theta_{5\%}$	0.86	0.96	0.96	0.98	0.83
rank	4	2	2	1	5
$\theta_{15\%}$	0.81	0.96	0.95	0.97	0.79
rank	4	2	3	1	5
$\theta_{25\%}$	0.77	0.95	0.94	0.94	0.76
rank	4	1	2	2	5
$\theta_{35\%}$	0.71	0.93	0.93	0.9	0.71
rank	4	1	1	3	4
$\theta_{45\%}$	0.62	0.92	0.91	0.86	0.66
rank	5	1	2	3	4
$\theta_{55\%}$	0.54	0.91	0.9	0.83	0.62
rank	5	1	2	3	4
$\theta_{65\%}$	0.49	0.88	0.88	0.79	0.56
rank	5	1	1	3	4
$\theta_{75\%}$	0.4	0.86	0.87	0.76	0.47
rank	5	2	1	3	4
$\theta_{85\%}$	0.32	0.84	0.84	0.6	0.46
rank	5	1	1	3	4
$\theta_{95\%}$	0.19	0.72	0.68	0.48	0.45
rank	5	1	2	3	4
summation (mean DSC, AUC, and θ_{δ}s)	56	15	21	31	51
rank	5	1	2	3	4
summation (θ_{δ}s only)	46	13	17	25	43
rank	5	1	2	3	4

Table 5.6: The rank of the accuracy performances of 5 semi-automatic segmentation algorithms (using interactions at all kinds of locations).

algorithm	GCBS	GSC	GSCSeq	Onecut	TRC
mean DSC	0.5733	0.8739	0.8731	0.8075	0.624
rank	5	1	2	3	4
AUC	0.5627	0.8592	0.8598	0.7946	0.6128
rank	5	2	1	3	4
$\theta_{5\%}$	0.85	0.96	0.97	0.98	0.81
rank	4	3	2	1	5
$\theta_{15\%}$	0.8	0.96	0.95	0.98	0.77
rank	4	2	3	1	5
$\theta_{25\%}$	0.77	0.94	0.94	0.97	0.75
rank	4	2	2	1	5
$\theta_{35\%}$	0.71	0.93	0.93	0.91	0.69
rank	4	1	1	3	5
$\theta_{45\%}$	0.62	0.92	0.91	0.87	0.65
rank	5	1	2	3	4
$\theta_{55\%}$	0.54	0.9	0.89	0.83	0.61
rank	5	1	2	3	4
$\theta_{65\%}$	0.49	0.87	0.87	0.78	0.55
rank	5	1	1	3	4
$\theta_{75\%}$	0.41	0.84	0.86	0.73	0.47
rank	5	2	1	3	4
$\theta_{85\%}$	0.32	0.79	0.79	0.59	0.46
rank	5	1	1	3	4
$\theta_{95\%}$	0.17	0.6	0.61	0.44	0.45
rank	5	2	1	4	3
summation (mean DSC, AUC, and θ_{δ}s)	56	19	19	31	51
rank	5	1	1	3	4
summation (θ_{δ}s only)	46	16	16	25	43
rank	5	1	1	3	4

For each table of Table 5.3, Table 5.4, Table 5.5, we can see that the ranks of mean DSCs of these segmentation algorithms are the same as the ranks that we re-rank the summations of the ranks calculated with mean DSCs, AUCs and θ_δs. This is because both of these two methods use the summation operations. If we just re-rank the summations of the ranks calculated with θ_δs, we can get the similar results as that we use the mean DSCs. In order to calculate the mean DSCs, we need sum up all of the DSCs and then divide it by the total number of segmentations. For the extensive method it sums up the ranks of calculated with mean DSCs, AUC, and θ_δs. When calculating the AUC of an algorithm, we need to sum up the area of all bins of the $\overline{CDF}\,(\theta)$. Besides the ranks calculated with mean DSCs and AUCs, the extensive method also sums up all the ranks calculated with $\theta_\delta = \max\left\{\theta|\overline{CDF}(\theta) \geq \delta\right\}$, where $\delta \in \{5\%, 15\%, 25\%, 35\%, 45\%, 55\%, 65\%, 75\%, 85\%, 95\%\}$. After the summation operations, the local properties of the distributions of DSCs are lost.

The extensive method can show more detailed information of the accuracy performances of segmentation algorithms than the method of using mean DSCs alone. Take Table 5.3 for example. When using mean DSCs to rank these algorithms, the rank of the DSCSeq algorithm is 1 and the rank of the Onecut algorithm is 3. We can get the same result when using the summation of different metrics to rank the DSCSeq algorithm and the Onecut algorithm. But when we use the extensive method to rank these algorithms, we can find that the GSCSeq algorithm is 25% likely to generate segmentations with accuracies of 0.94 or higher, and the Onecut algorithm is 25% likely to generate segmentations with accuracies of 0.97 or higher. Therefore, when using $\theta_{25\%}$ to rank these algorithms, the rank of the Onecut algorithm is smaller than that of the DSCSeq algorithm ($1 < 2$), which means the accuracy performance of the Onecut algorithm is better than the accuracy performance of the DSCSeq algorithm.

In practice, people may have different preferences when ranking segmentation algorithms. For example, those who prefer the segmentation algorithms which are more likely to generate high accuracy segmentations may be more concerned about the threshold that an algorithm is 5% likely to generate segmentations with this accuracies or higher, namely $\theta_{5\%}$. Those who can't tolerate segmentation algorithms to generate low accuracy segmentations may be more concerned about the threshold that the algorithms are 95% likely to generate segmentations with this accuracies or higher, namely $\theta_{95\%}$. Therefore, when summing up the ranks of different evaluation metrics, it is recommended that the weights of the ranks of mean DSC, AUC, and θ_δs should be different.

In summary, the extensive method can be used to rank the accuracy performances of segmentations. Comparing with the method that using mean accuracy alone to rank the accuracy performances of segmentation algorithms, the extensive method can show more detailed information. As the extensive method takes the distribution factors into account, it is more reasonable to use this method to rank segmentation algorithms whose accuracies of segmentations are not normal-distributed. At the same time, it can avoid the drawbacks of using $\overline{CDF}(\theta)$ to compare segmentation algorithms mentioned in chapter 4, such as the need of choosing θs manually.

6. Stability of Segmentation Reproducibility Measures

In chapter 4 and chapter 5, we illustrate how to evaluate, compare, and rank the accuracy performance of segmentation algorithms. Apart from accuracy, reproducibility of segmentation algorithms is also an important measure to characterize the performance of segmentation algorithms. This chapter focuses on what measures are suitable for characterizing the reproducibility of segmentation algorithms. As is mentioned in section 3.3, one objective of this thesis is to demonstrate GTC is the most insensitive measure among GTC, JTC, JDC, CV and ICC to the variation of segmentation group size. In this chapter, we will demonstrate it using real segmentation data.

6.1 Research Problem

Many measures were used to characterize the reproducibility of segmentation algorithms. However, only a few papers analyzed the properties of reproducibility measures. One major problem is that it is unclear when these measures are valid for characterizing the reproducibility of segmentation algorithms. It is known that the reproducibility of a segmentation algorithm can be calculated using the average of reproducibilities measured for a group of segmentations of each image in the dataset several groups of segmentations of a set of images. The group size of the segmentations may affect the value of reproducibility measures. However, few papers analyzed how the group size of segmentations would affect the values of reproducibility measures.

In [22], Eramian demonstrated that the group size of segmentations may affect the values of reproducibility measures and generalized Tanimoto coefficient (GTC) is the only measure which are insensitive to the variation of group size of segmentations among GTC, joint Dice coefficient (JDC), joint Tanimoto coefficient (JTC), coefficient of variation (CV), and intra-class correlation coefficient (ICC). The definition of GTC, JDC, JTC, CV, and ICC appear in section 2.3.2. The measures are often generalized from measures for pairs of images. However, as the existance of the built-in penalty, the generalization of these measures may contribute to some side effects, such as that the values of reproducibility measures may be affected by the number of segmentations. Intuitively GTC is fairer than other measures, because for each pair of segmentations, GTC is calculated by comparing the values of all pairs of pixels at the same location of that pair of segmentations. At first, he showed several images of segmentation groups of synthetic data and their reproducibility measures and compared the values of reproducibility measures and qualitative assessments. It is found that GTC and CV seems agree with the qualitative assessments. Then Eramian calculated the values of these reproducibility measures under different group size of synthetic segmentations. The result showed that GTC is the only measure which is stable under variations in segmentation group size, especially when the group size is less than five.

However, it is still unknown if the group size of segmentations will affect the reproducibility measures when applying real data. In [22], the parameters which are used to generate synthetic segmentations have a specific distribution. So the reproducibility measures are assumed to be affected by the group size of segmentations only. In practice, the segmentations are generated by different users, which means the reproducibility measures may be affected not only the group size, but also the difference between users.

The real segmentations may have different shape instead of the synthetic segmentations used in 6.3, which are all ellipsoids with different sizes and different positions. In this chapter, the effect of group size on the values of reproducibility measures will be studied using some real segmentation data.

6.2 Materials and Data collection

The material that used in this chapter is from Rau's project [62]. 25 images are selected from the BSDS500 dataset and were used as the trial images of Rau's experiment. All these images have only one foreground object. The foreground regions may be animals, humans, plants or vehicles. Example images and their corresponding ground-truth segmentations are shown in Figure 6.1.

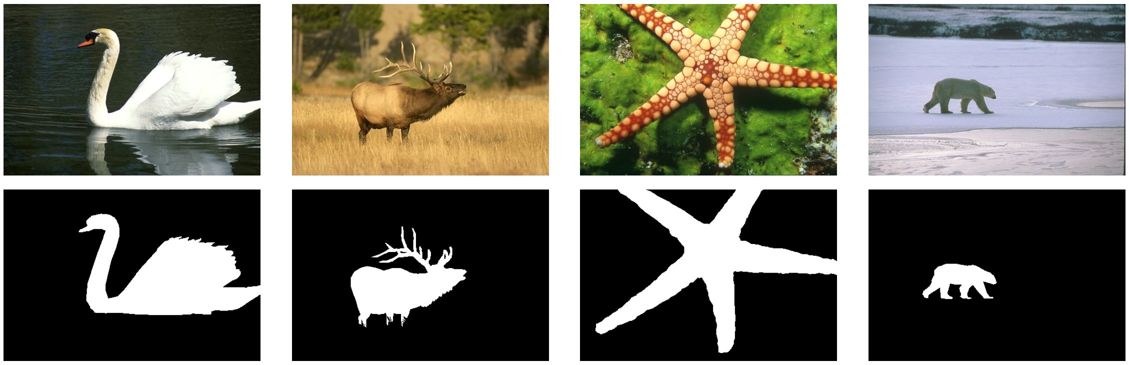


Figure 6.1: Example images of Rau's experiment. Top: original images, bottom: ground truths.

Rau used Boykov and Kolmogorov's Graph Cut algorithm [11] to segment images selected from the BSDS500 dataset. The theory of the Graph Cut algorithm can be found in the chapter 2. The implementation requires users to provide annotations on the images to be segmented.

Thirteen users participated in Rau's experiment. Each of them was asked to use two interaction methods to annotate these 25 images. One method is point-based interaction, which requires users to place seed points on images in different color to indicate foreground and background. The other method is stroke-based interaction, which requires users to draw strokes as in outlines on the images to be segmented. As a result, the semi-automatic implementation generated 650 images in total (375 segmentations using point-based interaction, and 375 segmentations using stroke-based interaction).

6.3 Data Analysis Methodology

In this chapter, the properties of GTC, JDC, JTC, CV, and ICC are analyzed using Rau's segmentations data. The results are compared with Eramian's results which were calculated using synthetic data [22].

At least 2 segmentations of an image are needed to calculate the reproducibility of an image. As there are 13 users in Rau's experiment, there are 13 segmentations generated by different users. Thus the maximum group size that can be considered with this data is 13.

The reproducibility of an algorithm can be characterized by the mean of the values of reproducibility of the different images. In section 6.1, it was mentioned that when applying real segmentation data generated by different users, the reproducibility measures may be affected by both the group size of segmentations and the difference between users. In order to minimize the effect of difference between users, we used all possible subsets of the 13 segmentations of each image from Rau's segmentations to calculate the mean reproducibility. Suppose we are going to use GTC to measure the reproducibility of the graph cut algorithm. When the group size of segmentation N is fixed, we can get different values of the same reproducibility measure if we use different combinations of segmentations. Because there may be more than one way to choose N segmentations from 13 segmentations of an image. For example, if the group size is 3, namely $N = 3$, there are $\binom{13}{3} = \frac{13!}{3! \times (13-3)!} = 286$ ways to select 3 segmentations from the 13 segmentations of an image randomly, and thus we can calculate 286 different GTCs. As there are 25 images in Rau's experiment, we can calculate $25 \times 286 = 7150$ GTCs. The mean of these 7150 GTCs can be regarded as the GTC of the graph cut segmentation algorithm when the group size is 3. In this thesis, N takes values from 2 to 13 and in general, we consider $\binom{13}{N}$ subsets of group size N.

6.4 Result and Discussion

Using different group sizes of segmentations, the reproducibility of the segmentation algorithm with point-based interaction and stroke-based interaction are evaluated with GTC, JDC, JTC, CV and ICC, which were introduced in chapter 2. The mean reproducibility measures averaged over N observations of them are shown in Figure 6.2. Each data point on those graphs is an average of many GTCs calculated from group size n.

For the point-based segmentations, the value of JDC, GTC, and JTC decrease with the increase of segmentation group size, and the value of ICC and CV increase with the increase of segmentation group size. The absolute values of the gradients of all these measures get smaller and smaller, which means when the segmentation group size is large enough, the values of these reproducibility measures are almost not affected by the variations of segmentation group size. It indicates that the group size of segmentations should be as large as possible in order to get an accurate measure of the reproducibility of segmentation algorithms.

Among these five kinds of reproducibility measures, GTC is the most stable one. When the group size increased from 2 to 13, the value of GTC changed from 0.569 to 0.525. JDC changed from 0.684 to 0.386. JTC changed from 0.569 to 0.198. CV changed from 0.220 to 0.499. ICC changed from 0.625 to 0.912. The value of GTC changed less than 0.05, while all of the others changed greater than 0.2. Therefore GTC is the only measure which is insensitive to group size.

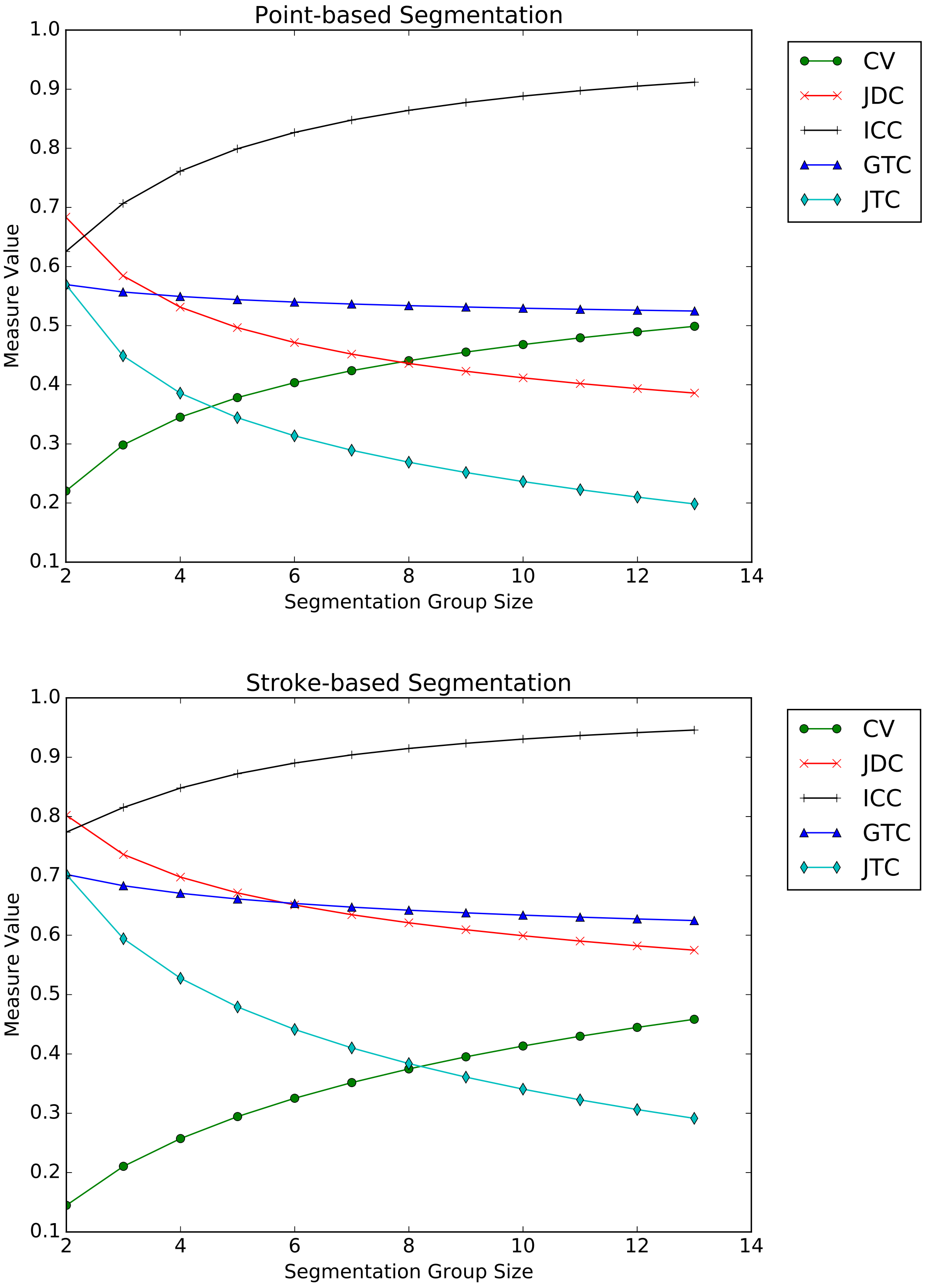


Figure 6.2: Mean reproducibility measures of the segmentations using point-based interaction (top) and using stroke-based interaction (bottom).

For stroke-based segmentations, all these reproducibility measures have behavior similar with that of the point-based segmentation. Under the condition of the same segmentation group size, all the values of the reproducibility measures of the stroke-based segmentations are greater than the values of the reproducibility measures of the point-based segmentations, except for the values of CV.

Basing on Figure 6.2, we can draw a conclusion that GTC is most insensitive to the segmentation group size. JTC, ICC, JDC and CV are sensitive to the group size of segmentations. In addition, the stroke-based segmentations have better reproducibility than the point-based segmentations.

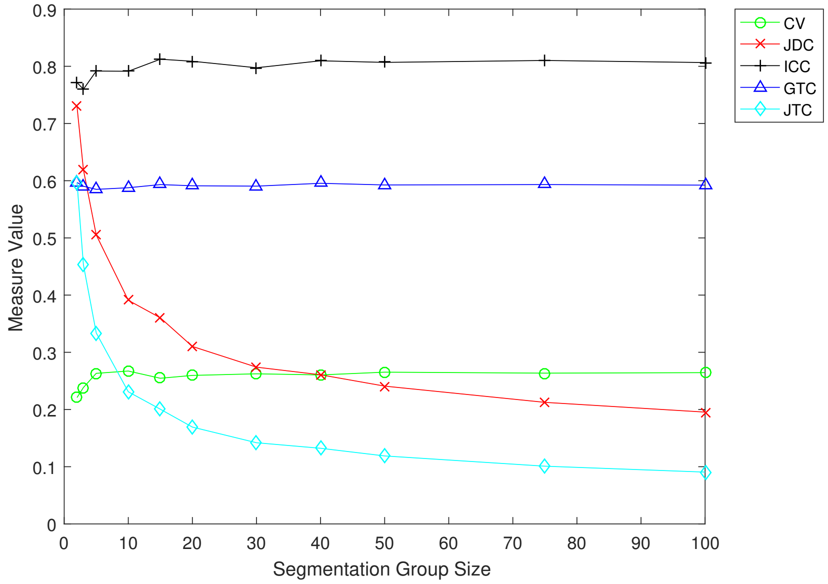


Figure 6.3: Mean reproducibility measures vs segmentation group size (Eramian's result basing on synthetic data [22], segmentation group size is in $\{2, 3, 5, 10, 20, 30, 40, 50, 75, 100\}$).

Eramian's result is shown in Figure 6.3. According to Figure 6.3, CV, ICC and GTC are insensitive to segmentation group size, while JTC and JDC are not. When the group size is no greater than 5, all of the reproducibility measures show some variation except GTC. If we focus on the interval $N \in [2, 13]$, the trends of these reproducibility measures of the synthetic data in Figure 6.3 are similar to the trends of real data in Figure 6.2.

It can be seen in Figure 6.3 that the GTC curve is very stable with the increase of segmentation group size, but in Figure 6.2, the GTC curves decline slowly. This is because the effect of the difference between users still exists. In [22], the synthetic data points are independent identically distributed, but in practical, the real data are not. Therefore, even though we choose the different combinations of segmentations of an image to calculate the GTCs and regard the mean of GTCs to be the reproducibility of the segmentation algorithm for that group size, the effect of difference between users can not be eliminated.

In summary, when the segmentation group size is small (≤ 10), GTC is most stable under the variations of segmentation group size. Therefore, GTC is better than JDC, JTC, CV and ICC for measuring the reproducibility of segmentation algorithms with small segmentation group size. In addition, in order to avoid the impact of segmentation group size on reproducibility measures, it is recommended that the group size of segmentations should be as large as possible.

7. Conclusion

This thesis has introduced a new method to evaluate and rank segmentation algorithms basing on their accuracy performances. At the mean time, the properties of five usually used reproducibility measures are analyzed. In this chapter, the contributions and future works are summarized.

7.1 Contributions

The goal of this thesis is to propose a methodology to evaluate, compare and rank segmentation algorithms basing on their accuracy performances. In addition, some suggestions are given for the evaluation of the reproducibility of semi-automatic segmentation algorithms. The details go as follow.

First, we illustrated how to use $\overline{CDF}(\theta)$ to evaluate the accuracy performances of segmentation algorithms and how to use $\overline{CDF}_{diff}(\theta)$ to compare segmentation algorithms basing on their accuracies. $\overline{CDF}(\theta)$ is a measure basing on the distribution of the accuracies of segmentations. It is the percentage of segmentations whose accuracies are no smaller than the given accuracy level θ. Therefore $\overline{CDF}(\theta)$ can provide local estimates of accuracy performances of segmentation algorithms at any accuracy level. Basing on $\overline{CDF}(\theta)$, $\overline{CDF}_{diff}(\theta)$ is proposed to compare two segmentation algorithms. It characterizes the difference of the $\overline{CDF}(\theta)$s of two segmentation algorithms. In other words, $\overline{CDF}_{diff}(\theta)$ measures the difference of probilities of segmentations whose accuracies are no smaller than the given accuracy level θ. Although $\overline{CDF}_{diff}(\theta)$ can be used to rank the accuracies of multiple segmentation algorithms, it is very complicated to choose accuracy levels θs to compare these algorithms pairwise and accordingly choosing the algorithm that we prefer.

Second, we developed an extensive method to rank the performance accuracy of segmentation algorithms. To begin with, the ranks, which are calculated using the mean of DSC, the AUC of the $\overline{CDF}(\theta)$, and the θ_δs of an algorithm, are summed up. Then the values of the summations of all segmentation algorithms are re-ranked. The new ranks are seemed as the ranks of these segmentation algorithms. Comparing with the method that using $\overline{CDF}_{diff}(\theta)$ to compare segmentation algorithms, this extensive method is concise and easy to operate. In addition, it can be used to rank segmentation algorithms no matter whether the accuracies of segmentations generated by these segmentation algorithms are normal-distributed or not.

Third, we demonstrated GTC is better than JDC, JTC, CV and ICC as a measure of reproducibility of segmentation algorithms. We calculated the JDCs, JTCs, CVs, ICCs and GTCs using real segmentation data. The result is consistent with Eramian's conclusion [22] that GTC is the only measure which is not sensitive to segmentation group size.

7.2 Future Work

There is still some work needed to be done. For example, the $\overline{CDF}(\theta)$ will be affected by the number of bins if we use equation 4.5. As the limitation of research time, we only used equation 4.5 to calculate $\overline{CDF}(\theta)$. It would be a better choice if we use the following equation to calculate $\overline{CDF}(\theta)$,

$$\overline{CDF}(\theta) = 1 - \frac{cnt\left(\{x \in X | x < \theta\}\right)}{N}, \tag{7.1}$$

where X is the finite set of accuracies and N is the number of these accuracies. In this way, we don't have to worry how the number of bins will affect the shape of $\overline{CDF}(\theta)$. In addition, even though the proposed extensive method for ranking segmentation algorithms can be easily operated, more work should be done to find reasonable weights when summing up the ranks calculated with mean DSCs, AUC, and θ_δs.

References

[1] Joel Akeret, Chihway Chang, Aurelien Lucchi, and Alexandre Refregier. Radio frequency interference mitigation using deep convolutional neural networks. *Astronomy and Computing*, 18:35–39, 2017.

[2] L. Alvarez, L. Baumela, P. Henriquez, and P. Marquez-Neila. Morphological snakes. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2197–2202, June 2010.

[3] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916, May 2011.

[4] Agus Zainal Arifin and Akira Asano. Image segmentation by histogram thresholding using hierarchical cluster analysis. *Pattern Recognition Letters*, 27(13):1515 – 1521, 2006.

[5] Christopher J. Armstrong, Brian L. Price, and William A. Barrett. Interactive segmentation of image volumes with Live Surface. *Computers & Graphics*, 31(2):212 – 229, 2007.

[6] Mohamad Awad. An unsupervised artificial neural network method for satellite image segmentation. *Int. Arab J. Inf. Technol.*, 7(2):199–205, 2010.

[7] Angela R. Baerwald, Gregg P. Adams, and Roger A. Pierson. Characterization of ovarian follicular wave dynamics in women. *Biology of Reproduction*, 69(3):1023–1031, 2003.

[8] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3169–3176, June 2010.

[9] Reinhard Beichel, Alexander Bornik, Christian Bauer, and Erich Sorantin. Liver segmentation in contrast enhanced CT data using graph cuts and interactive 3D segmentation refinement methods. *Medical Physics*, 39(3):1361–1373, 2012.

[10] Norman Biggs. Algebraic potential theory on graphs. *Bulletin of the London Mathematical Society*, 29(6):641–682, 1997.

[11] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, Sept 2004.

[12] Y. Y. Boykov and M. P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 1, pages 105–112 vol.1, 2001.

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

[14] T. Chai and R. R. Draxler. Root mean square error (RMSE) or mean absolute error (MAE)? - Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, 7(3):1247–1250, 2014.

[15] Heng-Da Cheng, X₋ H₋ Jiang, Ying Sun, and Jingli Wang. Color image segmentation: advances and prospects. *Pattern recognition*, 34(12):2259–2281, 2001.

[16] Keh-Shih Chuang, Hong-Long Tzeng, Sharon Chen, Jay Wu, and Tzong-Jer Chen. Fuzzy c-means clustering with spatial information for image segmentation. *Computerized Medical Imaging and Graphics*, 30(1):9 – 15, 2006.

[17] W. R. Crum, O. Camara, and D. L. G. Hill. Generalized overlap measures for evaluation and validation in medical image analysis. *IEEE Transactions on Medical Imaging*, 25(11):1451–1461, Nov 2006.

[18] P. Dastidar, T. Heinonen, T. Vahvelainen, I. Elovaara, and H. Eskola. Computerised volumetric analysis of lesions in multiple sclerosis using new semi-automatic segmentation software. *Medical & Biological Engineering & Computing*, 37(1):104–107, 1999.

[19] Delia Cabrera DeBuc, Gabor Márk Somfai, Sudarshan Ranganathan, Erika Tatrai, Mária Ferencz, and Carmen A Puliafito. Reliability and reproducibility of macular segmentation using a custom-built optical coherence tomography retinal image analysis software. *Journal of biomedical optics*, 14(6):064023, 2009.

[20] M. Egmont-Petersen, D. de Ridder, and H. Handels. Image processing with neural networksa review. *Pattern Recognition*, 35(10):2279 – 2301, 2002.

[21] AJ Einstein, J Gil, S Wallenstein, CA Bodian, M Sanchez, DE Burstein, H-S Wu, and Z Liu. Reproducibility and accuracy of interactive segmentation procedures for image analysis in cytology. *Journal of microscopy*, 188(2):136–148, 1997.

[22] M. Eramian. Worst-case local boundary precision in global measures of segmentation reproducibility. In *2013 International Conference on Computer and Robot Vision*, pages 59–66, May 2013.

[23] Courtenay L. Glisson, Hernan O. Altamar, S. Duke Herrell, Peter Clark, and Robert L. Galloway. Comparison and assessment of semi-automatic image segmentation in computed tomography scans for image-guided kidney surgery. *Medical Physics*, 38(11):6265–6274, 2011.

[24] Lena Gorelick, Olga Veksler, Yuri Boykov, and Claudia Nieuwenhuis. Convexity shape prior for segmentation. In *European Conference on Computer Vision*, pages 675–690. Springer, 2014.

[25] L. Grady. Random Walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1768–1783, Nov 2006.

[26] Houssem-Eddine Gueziri, Michael J. McGuffin, and Catherine Laporte. A generalized graph reduction framework for interactive segmentation of large images. *Computer Vision and Image Understanding*, 150:44 – 57, 2016.

[27] Varun Gulshan, Carsten Rother, Antonio Criminisi, Andrew Blake, and Andrew Zisserman. Geodesic star convexity for interactive image segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference*, pages 3129–3136. IEEE, 2010.

[28] S M Rafizul Haque. *Methodology for Extensive Evaluation of Semiautomatic and Interactive Segmentation Algorithms Using Simulated Interaction Method.* PhD thesis, University of Saskatchewan, 2016.

[29] M. Havaei, P. M. Jodoin, and H. Larochelle. Efficient interactive brain tumor segmentation as within-brain kNN classification. In *2014 22nd International Conference on Pattern Recognition*, pages 556–561, Aug 2014.

[30] M Heath, K Bowyer, D Kopans, R Moore, and P Kegelmeyer. The digital database for screening mammography. *Digital mammography*, pages 431–434, 2000.

[31] Michael Heath, Kevin Bowyer, Daniel Kopans, P Kegelmeyer, Richard Moore, Kyong Chang, and S Munishkumaran. Current status of the digital database for screening mammography. In *Digital mammography*, pages 457–460. Springer, 1998.

[32] R. Hebbalaguppe, K. McGuinness, J. Kuklyte, and G. Healy. How interaction methods affect image segmentation: User experience in the task. In *2013 1st IEEE Workshop on User-Centered Computer Vision (UCCV)*, pages 19–24, Jan 2013.

[33] Yrj Hme and Mika Pollari. Semi-automatic liver tumor segmentation with hidden Markov measure field model and non-parametric distribution estimation. *Medical Image Analysis*, 16(1):140 – 149, 2012.

[34] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

[35] Hiroshi Ishikawa, Daniel M Stein, Gadi Wollstein, Siobahn Beaton, James G Fujimoto, and Joel S Schuman. Macular segmentation with optical coherence tomography. *Investigative ophthalmology & visual science*, 46(6):2012–2017, 2005.

[36] Yangqing Jia and Changshui Zhang. Learning distance metric for semi-supervised image segmentation. In *2008 15th IEEE International Conference on Image Processing*, pages 3204–3207, Oct 2008.

[37] Yuan Jin and Hanif M. Ladak. Software for interactive segmentation of the carotid artery from 3D black blood magnetic resonance images. *Computer Methods and Programs in Biomedicine*, 75(1):31 – 43, 2004.

[38] Hyung Woo Kang and Sung Yong Shin. Enhanced lane: interactive image segmentation by incremental path map construction. *Graphical Models*, 64(5):282 – 303, 2002.

[39] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[40] Gary G Koch. Intraclass correlation coefficient. *Encyclopedia of statistical sciences*, 1982.

[41] Pushmeet Kohli, Hannes Nickisch, Carsten Rother, and Christoph Rhemann. User-centric learning and evaluation of interactive segmentation systems. *International Journal of Computer Vision*, 100(3):261–274, 2012.

[42] Teuvo Kohonen. The self-organizing map. *Neurocomputing*, 21(1-3):1–6, 1998.

[43] Terry K Koo and Mae Y Li. A guideline of selecting and reporting intraclass correlation coefficients for reliability research. *Journal of chiropractic medicine*, 15(2):155–163, 2016.

[44] Karl Krissian, Jose M. Carreira, Julio Esclarin, and Manuel Maynar. Semi-automatic segmentation and detection of aorta dissection wall in MDCT angiography. *Medical Image Analysis*, 18(1):83 – 102, 2014.

[45] Neeraj Kumar, Peter N. Belhumeur, Arijit Biswas, David W. Jacobs, W. John Kress, Ida C. Lopez, and João V. B. Soares. Leafsnap: A computer vision system for automatic plant species identification. In *Computer Vision – ECCV 2012*, pages 502–516, 2012.

[46] Binh Huy Le, Zhigang Deng, James Xia, Yu-Bing Chang, and Xiaobo Zhou. *An interactive geometric technique for upper and lower teeth segmentation*, pages 968–975. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[47] C. Li, C. Xu, C. Gui, and M. D. Fox. Distance regularized Level Set evolution and its application to image segmentation. *IEEE Transactions on Image Processing*, 19(12):3243–3254, Dec 2010.

[48] Yuanzhong Li, Wataru Ito, and Shingo Iwano. Interactive segmentation of lung nodules using AdaBoost and Graph Cuts. In *The Fourth International Workshop on Pulmonary Image Analysis, Medical Image Computing and Computer Assisted Intervention*, pages 125–133, 2011.

[49] Fangfang Lu, Zhouyu Fu, and Antonio Robles-Kelly. *Efficient Graph Cuts for Multiclass Interactive Image Segmentation*, pages 134–144. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[50] Zhen Ma, João Manuel RS Tavares, Renato Natal Jorge, and T Mascarenhas. A review of algorithms for medical image segmentation and their applications to the female pelvic cavity. *Computer Methods in Biomechanics and Biomedical Engineering*, 13(2):235–246, 2010.

[51] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[52] Sarah A. Mason, Tuathan P. O'Shea, Ingrid M. White, Susan Lalondrelle, Kate Downey, Mariwan Baker, Claus F. Behrens, Jeffrey C. Bamber, and Emma J. Harris. Towards ultrasound-guided adaptive radiotherapy for cervical cancer: evaluation of Elekta's semi-automated uterine segmentation method on 3D ultrasound images. *Medical Physics*, pages n/a–n/a.

[53] Kevin McGuinness and Noel E. OConnor. A comparative evaluation of interactive segmentation algorithms. *Pattern Recognition*, 43(2):434 – 444, 2010. Interactive Imaging and Vision.

[54] T. McInemey and D. Terzopoulos. Topology adaptive deformable surfaces for medical image volume segmentation. *IEEE Transactions on Medical Imaging*, 18(10):840–850, Oct 1999.

[55] T. McInerney. SketchSnakes: Sketch-line initialized Snakes for efficient interactive medical image segmentation. *Computerized Medical Imaging and Graphics*, 32(5):331 – 352, 2008.

[56] Emmanouil Moschidis and Jim Graham. Simulation of user interaction for performance evaluation of interactive image segmentation methods. In *Medical Image Understanding and Analysis*, pages 209–213, 2009.

[57] Charles Newey, Owain Jones, and Hannah Dee. Shadow Detection/Texture Segmentation Computer Vision Dataset, July 2016.

[58] Lszl G. Nyl, Alexandre X. Falco, and Jayaram K. Udupa. Fuzzy-connected 3D image segmentation at interactive speeds. *Graphical Models*, 64(5):259 – 281, 2002.

[59] S.D Olabarriaga and A.W.M Smeulders. Interaction in the segmentation of medical images: A survey. *Medical Image Analysis*, 5(2):127 – 142, 2001.

[60] Sang Hyun Park, Soochahn Lee, Il Dong Yun, and Sang Uk Lee. Structured patch model for a unified automatic and interactive segmentation framework. *Medical Image Analysis*, 24(1):297 – 312, 2015.

[61] M. Rajchl, J. Yuan, J. A. White, E. Ukwatta, J. Stirrat, C. M. S. Nambakhsh, F. P. Li, and T. M. Peters. Interactive hierarchical-flow segmentation of scar tissue from late-enhancement cardiac MR images. *IEEE Transactions on Medical Imaging*, 33(1):159–172, Jan 2014.

[62] Steven Rau. Analysis of user input methods for semi-automatic segmentation.

[63] Wilburn E Reddick, John O Glass, Edwin N Cook, T David Elkin, and Russell J Deaton. Automated segmentation and classification of multispectral magnetic resonance images of brain using artificial neural networks. *IEEE Transactions on medical imaging*, 16(6):911–918, 1997.

[64] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "GrabCut": Interactive foreground extraction using iterated Graph Cuts. *ACM Trans. Graph.*, 23(3):309–314, August 2004.

[65] Warren S Sarle. Neural networks and statistical models. 1994.

[66] Holger Scherl, Joachim Hornegger, Marcus Prümmer, and Michael Lell. Semi-automatic Level Set based segmentation and stenosis quantification of the internal carotid artery in 3D CTA data sets. *Medical image analysis*, 11(1):21–34, 2007.

[67] James A Sethian et al. Level Set methods and fast marching methods. *Journal of Computing and Information Technology*, 11(1):1–2, 2003.

[68] T. Shepherd, S. J. D. Prince, and D. C. Alexander. Interactive lesion segmentation with shape priors from offline and online learning. *IEEE Transactions on Medical Imaging*, 31(9):1698–1712, Sept 2012.

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

[70] Dirk Smeets, Dirk Loeckx, Bert Stijnen, Bart De Dobbelaer, Dirk Vandermeulen, and Paul Suetens. Semi-automatic Level Set segmentation of liver tumors combining a spiral-scanning technique with supervised fuzzy pixel classification. *Medical Image Analysis*, 14(1):13 – 20, 2010.

[71] J. Sourati, D. Erdogmus, J. G. Dy, and D. H. Brooks. Accelerated learning-based interactive image segmentation using pairwise constraints. *IEEE Transactions on Image Processing*, 23(7):3057–3070, July 2014.

[72] Herbert A Sturges. The choice of a class interval. *Journal of the american statistical association*, 21(153):65–66, 1926.

[73] H. Su, Z. Yin, S. Huh, T. Kanade, and J. Zhu. Interactive cell segmentation based on active and semi-supervised learning. *IEEE Transactions on Medical Imaging*, 35(3):762–777, March 2016.

[74] László Szilagyi, Zoltán Benyo, Sándor M Szilágyi, and HS Adam. MR brain image segmentation using an enhanced fuzzy c-means algorithm. In *Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE*, volume 1, pages 724–726. IEEE, 2003.

[75] Hui Tang, Robbert S van Onkelen, Theo van Walsum, Reinhard Hameeteman, Michiel Schaap, Fufa L Tori, Quirijn JA van den Bouwhuijsen, Jacqueline CM Witteman, Aad van der Lugt, Lucas J van Vliet, et al. A semi-automatic method for segmentation of the carotid bifurcation and bifurcation angle quantification on black blood MRA. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 97–104. Springer, 2010.

[76] Meng Tang, Lena Gorelick, Olga Veksler, and Yuri Boykov. Grabcut in one cut. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1769–1776. IEEE, 2013.

[77] Wenbing Tao and Xue-Cheng Tai. Multiple piecewise constant with geodesic active contours (MPC-GAC) framework for interactive image segmentation using graph cut optimization. *Image and Vision Computing*, 29(8):499 – 508, 2011.

[78] Andrew Top, Ghassan Hamarneh, and Rafeef Abugharbieh. *Active Learning for Interactive 3D Image Segmentation*, pages 603–610. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[79] Olga Veksler. Star shape prior for Graph-Cut image segmentation. In *European Conference on Computer Vision*, pages 454–467. Springer, 2008.

[80] Luminita A. Vese and Tony F. Chan. A multiphase Level Set framework for image segmentation using the Mumford and Shah model. *International Journal of Computer Vision*, 50(3):271–293, 2002.

[81] Henri A Vrooman, Chris A Cocosco, Fedde van der Lijn, Rik Stokking, M Arfan Ikram, Meike W Vernooij, Monique MB Breteler, and Wiro J Niessen. Multi-spectral brain tissue segmentation using automatically trained k-nearest-neighbor classification. *Neuroimage*, 37(1):71–81, 2007.

[82] Xiang-Yang Wang, Ting Wang, and Juan Bu. Color image segmentation using pixel wise support vector machine classification. *Pattern Recognition*, 44(4):777 – 787, 2011.

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

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

[85] L. Zhang and Q. Ji. A Bayesian Network model for automatic and interactive image segmentation. *IEEE Transactions on Image Processing*, 20(9):2582–2593, Sept 2011.

[86] Kelly H Zou, Simon K Warfield, Aditya Bharatha, Clare MC Tempany, Michael R Kaus, Steven J Haker, William M Wells, Ferenc A Jolesz, and Ron Kikinis. Statistical validation of image segmentation quality based on a spatial overlap index1: scientific reports. *Academic radiology*, 11(2):178–189, 2004.