Intelligent Medical Image Segmentation Using Evolving Fuzzy Sets

by

Ahmed Abdelrahman Othman

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2013

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Image segmentation is an important step in the image analysis process. Current image segmentation techniques, however, require that the user tune several parameters in order to obtain maximum segmentation accuracy, a computationally inefficient approach, especially when a large number of images must be processed sequentially in real time. Another major challenge, particularly with medical image analysis, is the discrepancy between objective measures for assessing and guiding the segmentation process, on the one hand, and the subjective perception of the end users (e.g., clinicians), on the other. Hence, the setting and adjustment of parameters for medical image segmentation should be performed in a manner that incorporates user feedback.

Despite the substantial number of techniques proposed in recent years, accurate segmentation of digital images remains a challenging task for automated computer algorithms. Approaches based on machine learning hold particular promise in this regard because, in many applications, including medical image analysis, frequent user intervention can be assumed as a means of correcting the results, thereby generating valuable feedback for algorithmic learning.

This thesis presents an investigation of the use of evolving fuzzy systems for designing a method that overcomes the problems associated with medical image segmentation. An evolving fuzzy system can be trained using a set of invariant features, along with their optimum parameters, which act as a target for the system. Evolving fuzzy systems are also capable of adjusting parameters based on online updates of their rule base. This thesis proposes three different approaches that employ an evolving fuzzy system for the continual adjustment of the parameters of any medical image segmentation technique.

The first proposed approach is based on evolving fuzzy image segmentation (EFIS). EFIS can adjust the parameters of existing segmentation methods and switch between them or fuse their results. The evolving rules have been applied for breast ultrasound images, with EFIS being used to adjust the parameters of three segmentation methods: global thresholding, region growing, and statistical region merging. The results for ten independent experiments for each of the three methods show average increases in accuracy of 5%, 12% and 9% respectively. A comparison of the EFIS results with those obtained using five other thresholding methods revealed improvements. On the other hand, EFIS has some weak points, such as some fixed parameters and an inefficient feature calculation process.

The second approach proposed as a means of overcoming the problems with EFIS is a new version of EFIS, called self-configuring EFIS (SC-EFIS). SC-EFIS uses the available data to estimate all of the parameters that are fixed in EFIS and has a feature selection process that selects suitable features based on current data. SC-EFIS was evaluated using the same three methods as for EFIS. The results show that SC-EFIS is competitive with EFIS but provides a higher level of automation.

In the third approach, SC-EFIS is used to dynamically adjust more than one parameter, for example, three parameters of the normalized cut (N-cut) segmentation technique. This method, called multi-parametric SC-EFIS (MSC-EFIS), was applied to magnetic resonance images (MRIs) of the bladder and to breast ultrasound images. The results show the ability of MSC-EFIS to adjust multiple parameters. For ten independent experiments for each of the bladder and the breast images, this approach produced average accuracies that are 8% and 16% higher respectively, compared with their default values.

The experimental results indicate that the proposed algorithms show significant promise in enhancing image segmentation, especially for medical applications.

Acknowledgements

All praise is due to God almighty who gave me the power and the knowledge to complete this work and The God's peace and blessings be upon prophet Muhammad and his family and companions.

I would like first to thank my supervisor, Prof. Hamid R. Tizhoosh, for his enormously beneficial help and support over more than four years. I very much appreciate his supervision and willingness to share his experience and knowledge. I am also grateful to Prof. Hamed Nassar for his continual encouragement and support.

My thanks extend as well to my committee members, Prof. Alireza Sadeghian, Prof. Jonathan Kofman, Prof. John Zelek, and Prof. George Freeman, for providing their valuable comments.

I would also like to express my deep appreciation and gratitude to all my friends and family who have sustained and helped me. Special thanks go to Mahmoud Khater, Ahmed Farahat, Mohammed Ismail, Mohamemd Hamouda, Ahmed Ghanem, and Amr Mostafa.

Additional credit goes to my father (may God shower his mercy on him). Although the period during which I lived with him before his death was short, I really learned a great deal from him, and his influence on my life has been immense. I wish to express my extreme gratitude to my mother, who, along with my uncle, faced endured numerous difficulties in raising me, my brother, and my sister. I am indebted to my uncle, who always assumed responsibility for taking care of me in such a way that I did not constantly miss my father. I give special thanks for and offer my deep appreciation to my beloved wife for her support and patience, and also convey sincere gratitude as well to my brother, sister, and cousins.

Dedication

To:

- My father Abdelrahman
- My mother Awatif
- My uncle Ahmed Abdellatif
- My wife Marwa Ali
- My lovely daughter Hedi
- My family

Table of Contents

List of Tables

List of Figures

Nomenclature

- A_C: Approximation matrix.
- C_L: Columns.
- D_S: SIFT descriptors.
- D_C: Discrete cosine transform.
- E_V: Edge variance parameter of N-cut
- F_F: Feature similarity feature selection technique.
- F_L: Laplacian feature selection technique.
- F_P: Spectral feature selection technique.
- F_M: Multi-cluster feature selection technique.
- F_G: Greedy feature selection technique.
- F_C: Correlation technique.
- F_S: Feature matrix for a test image.
- F_T: Feature matrix for a training image.
- F^*: The general feature matrix.
- G: Gold standard image.
- G_M: Gradient magnetite.
- I: Image.
- J: Jaccard index.
- L: Number of rows in general feature matrix.
- M: Input matrix.

- N_S: Number of segment parameter of N-cut.
- N_{S_O}: Optimum number of segment.
- N_{S_F}: Final optimum number of segment.
- N_E: Number of test images.
- N_R: Number of training images.
- N_I: Number of available images.
- N_A: All SIFT points.
- N_F: Final number of SIFT points.
- N_W: Number of rows.
- N_C: Number of columns.
- N_T: Number of all features.
- N_{T_1}: Number of features after discarding 99% correlated features.
- N_{T_2}: Number of features after discarding 90% correlated features.
- N_{T_3}: Number of features after ensemble.
- N_L: Final Number of features.
- O: Output matrix.
- p: Parameters.
- p^*: Best parameters.
- R_D: Radius parameter in N-cut technique.
- R_{D_O}: Optimum radius of N-cut technique.
- R_{D_F}: Final optimum radius.

- R_A: Rate parameter in N-cut technique.
- R_{A_O}: Optimum rate of N-cut technique.
- R_{A_F}: Final optimum rate of N-cut technique.
- R_O: Region of interest.
- R_F: Fuzzy rule.
- R_W: Rows.
- R_C: Rectangle around SIFT points.
- S: A segment.
- T_O: Output matrix for test image.
- T^*: Final single output threshold.
- T: All available best parameters matrix.
- T_B: Calculated best parameters for test image.
- T_R: Best parameters for training images.
- W: Weight.

Chapter 1

Introduction

1.1 Image Segmentation

An image is a two-dimensional function $f(x, y)$, where x and y are spatial coordinates. The amplitude of f at any coordinate (x, y) is the intensity or gray level of the image at that location. An image is called digital when x, y, and their values are all finite and discrete [1].

The task of marking or extracting a group of pixels that belongs to the same object is a major step in many image analysis applications [1]. This process, also known as image segmentation involves the assigning of a label to each individual pixel in an image so that all pixels with the same label share specific visual characteristics. The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image. The pixels in a given region may be similar with respect to some characteristics or to a computed property, such as colour, intensity, or texture. Adjacent regions may be significantly different with respect to the same characteristics [2]. Edges and texture are additional image attributes that are also useful for segmentation [3]. In other words, the essential feature of the analysis of objects in images is the ability to distinguish between the objects of interest and irrelevant data, such as the background of the image. The techniques used to determine the objects of interest are usually referred to as segmentation techniques: that is they separate the foreground from the background.

Image segmentation is a very important process because of its use in a number of practical

applications, such as medical imaging: the creation of images of the human body for clinical purposes and to help in the diagnosis and treatment of diseases such as cancer. Medical image segmentation is used for locating tumours and other pathologies, measuring tissue volumes, directing computer-guided surgery, diagnosing, planning treatment, and studying anatomical structures [4]. Medical images can be captured in a variety of ways such as X-ray, ultrasound, computed tomography, and magnetic resonance imaging [5].

The literature contains a large number of reports of segmentation algorithms based on a variety of methods, such as thresholding [6], region growing [1], watershed [7, 8, 9], active contours [10], graph cuts [11], and many clustering approaches [4, 12]. Techniques based on fuzzy set theory have also been proposed for segmentation tasks [13, 14, 15].

In spite of this broad exploration, segmentation remains highly challenging, especially if the larger goal is to design a universal segmentation algorithm capable of correctly, i.e., accurately, classifying any and all objects of interest in an image or series of images belonging to the same application. A main reason researchers have thus far failed to design such a universal algorithm is the intractably great diversity of digital images and their contents. Images can be expected to have varying resolutions and to contain objects depicted (digitized) with a huge variety of intensities and texture patterns. Hence, without a learning or dynamic adjustment technique, a generally applicable segmentation method, even for a specific class of images, appears to be beyond current capabilities. The challenge of designing a capable segmentation method becomes even more apparent in view of the impossibility of acquiring accurate results and meaningful segments even when a specific image category comprised of a very small subset of all digital images is targeted. For example, if the investigation is restricted to only medical imaging using ultrasound, and the segmentation target is limited to the narrow set of objects such as breast lesions, accurate segmentation results can still not be delivered. Thus, empirical knowledge indicates that generically designed algorithms are simply incapable of overcoming the challenges inherent in segmentation unless they are customized or fine-tuned. Acknowledging the desire for a general approach leads to the recognition that acquiring superior results is possible only if the methods are adjusted to the particularities of the imaging technique and to the domain knowledge of the imaged objects.

In the research presented in this thesis, the motivation has been the conviction that a superior segmentation framework ought to remain general but also be continually customized to fit the

problem at hand. To achieve this goal, the guiding maxim is the application of perpetual learning to the updating of the decision mechanisms for pixel classification. Using a type of learning technique for clustering and segmentation, however, is not an entirely new idea [16, 17, 18, 19]. The question has been why an approach should be chosen based on fuzzy segmentation when one could use more learning-oriented techniques such as neural nets, evolutionary algorithms, or swarm intelligence. Simply put, the answer is that in light of the introduction of evolving fuzzy systems [20, 21], a fuzzy platform offers a unique and flexible framework for knowledge representation, and thus a convenient strategy for domain-refined segmentation.

This thesis presents a proposed evolving fuzzy method for image segmentation based on expert user feedback. This method is designed to enable the adjustment of the parameters associated with any medical image segmentation technique. The central concept is to extract features from images and then associate them with optimal segmentation parameters via the formation of fuzzy rules based on past data (images plus their manually created segments) and then to continue the learning process (the evolution of the rules) through continual observation and the integration of user feedback in order to produce new images (images plus user-corrected segments). Three different approaches are proposed. The first, evolving fuzzy image segmentation (EIFS), is designed to adjust the parameters associated with any medical image segmentation technique. As an example, EFIS is used to estimate the parameters of three different segmentation techniques. The second approach is an advanced version of EFIS, called self-configuring EFIS (SC-EFIS), which incorporates a feature selection mechanism and produces a higher level of automation. The final approach involves a modification of SC-EFIS that enables the adjustment of multi-parametric segmentation techniques such as the normalized cut (N-cut) segmentation technique.

Medical images, specifically breast ultrasound and bladder MR images, were used in order to validate the proposed methods and to demonstrate how evolving rules lead to improved results.

1.2 Problem Formulation

Most medical image segmentation techniques have several parameters that must be tuned manually or experimentally in order to obtain the maximum degree of segmentation accuracy. Any change in the values of these parameters affects the accuracy of the segmentation for each image. The process of manually tuning these parameters is time consuming and may not produce

the required generalization. The primary focus of this research encompasses the threshold values in thresholding techniques, the similarity threshold that is part of region growing methods, the scale in statistical region merging technique and the set of parameters that are a component of the normalized cut (N-cut). These techniques represent areas in which the necessity of tuning parameters impedes the effectiveness of medical image segmentation. The other, more significant, problem addressed by this research is that, in medical image analysis, the expert user has the final say and always modifies the results of the segmentation to match his or her anatomical knowledge. Adjusting the segmentation parameters is thus effective only if based on user feedback. A learning or knowledge-based approach that puts the expert user (observer) at the centre of the calculations therefore has the highest chance of success.

An additional difficulty related to the application of these concepts for medical imaging is the problem associated with global image thresholding. Because of differences in the properties of the images, the use of a static thresholding technique such as Otsu [22] may provide accurate segmentation for some images but not for all. For example, as illustrated in Fig. 1.1, the top image segmented with Otsu is very close to the gold standard image, with a segmentation accuracy level of 95%. However, another image segmented using the same method generated a result very different from the gold standard image, with a segmentation accuracy level of only about 30%.

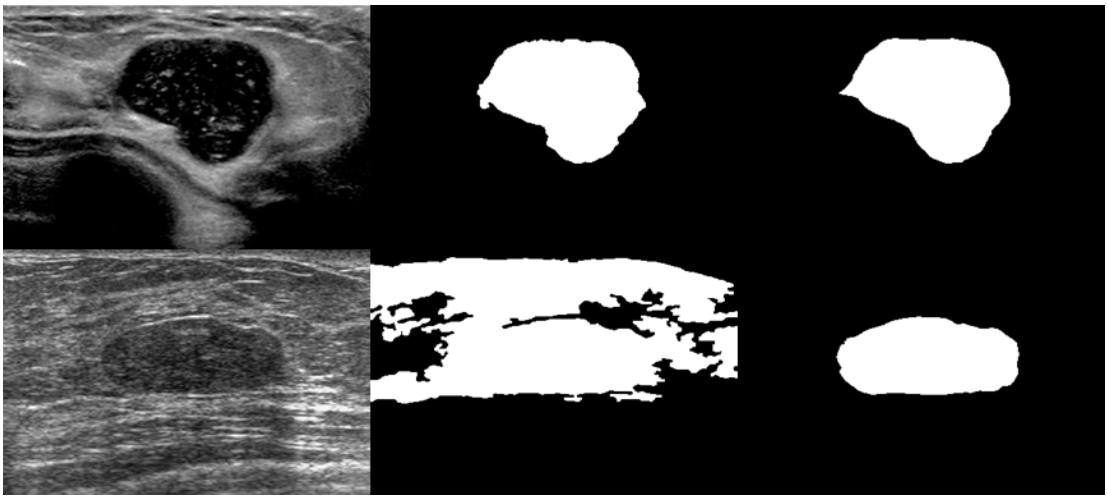

Figure 1.1: Global thresholding results from left to right: original image, segmented images using Otsu, and the gold standard image.

With the region growing technique, as shown in Fig. 1.2, the resulting segment for the first image, which was assigned a similarity threshold of 0.15, is very close to the gold standard image, with a segmentation accuracy of about 95.5%. However, segmenting the second image using the same threshod of 0.15 generated a segment well removed from the gold standard image, with a segmentation accuracy of about 83%. An additional threshold of 0.13 was used to segment the two images. For the first image, a segmentation accuracy of 95% was achieved: less than the 95.5% accuracy obtained when the threshold was 0.15. For the second image, a better result was achieved: a segmentation accuracy of about 91%.

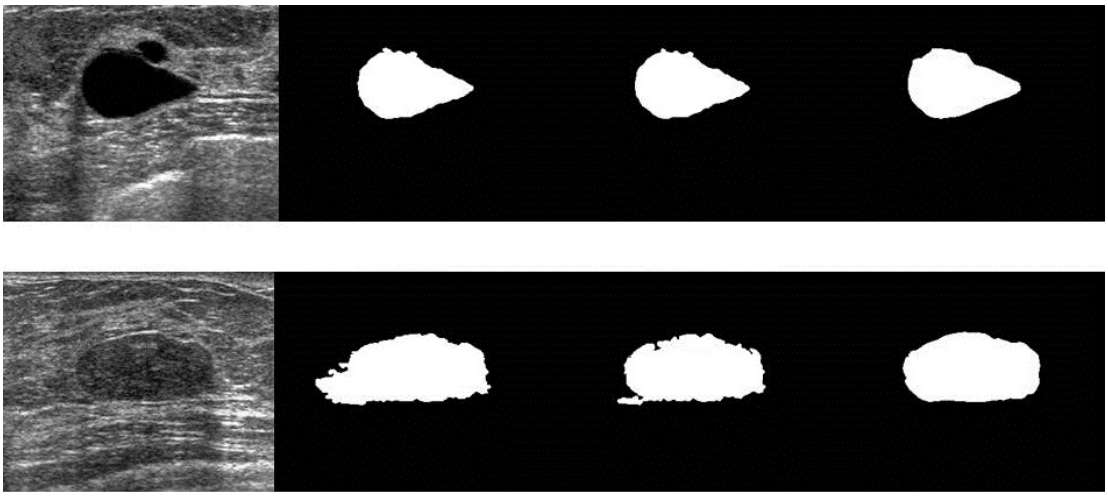

Figure 1.2: Region growing results from left to right: original image, segmented images using a threshold = 0.15, segmented images using a threshold = 0.13, and the gold standard image.

On the other hand, the N-cut segmentation technique has more than one parameter that affects the accuracy of the segmentation. These parameters are the number of segments N_S, the rate R_A, the radius R_D, and the edge variance E_V. Different values of these parameters thus affect segmentation accuracy. For example, as can be seen in Fig. 1.3, for the same image, different values of the parameters produce different segmentation accuracies relative to the gold standard image.

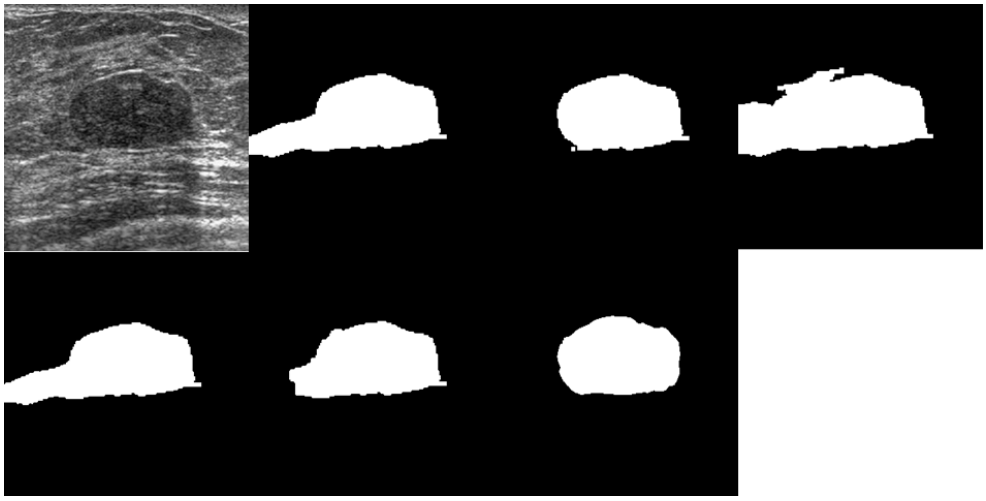

Figure 1.3: Different segments generated using different values of N-cut parameters (from top to bottom): Original image, segment using default values, segment using $N_S = 10$, segment using $R_D = 20$, segment using $R_A = 0.5$, and segment using $E_V = 0.2$.

1.3 Contribution

To enhance the segmentation of medical images, this research proposes a framework that updates an initial fuzzy system in order to adjust the parameters in segmentation methods through the online integration of user feedback. This work focused on solving two problems as examples of the observer-oriented adjustment of parameters with the goal of increasing the accuracy of the segmentation:

- Adjust a single parameter:
    1. The threshold in global binarization.
    2. The similarity threshold in region growing technique.
    3. The scale value in statistical region merging method.
- Adjust more than one parameter: The N-cut segmentation technique parameters.

An evolving fuzzy image segmentation technique is proposed that incorporates online learning as a means of solving these problems. The proposed algorithm uses a set of training data to extract

a set of features that are used to build an initial set of fuzzy rules. These rules are then employed to segment new images, which simultaneously serve as a new source of information for updating the rule base. The updating process results in the generation of a new set of rules. As long as new images are available, the process is repeated so that any incoming new image is segmented and then contributes to the process of regenerating the rule base, which results in a broader general knowledge base. This approach is designed not to increase the segmentation accuracy of the process involved in a specific segmentation task (comprising of several methods) but rather to be used as a general method of adjusting the segmentation parameters of a given segmentation technique in order to increase its accuracy. The previously mentioned techniques were used as examples to enable the evaluation of the proposed approach and to demonstrate its ability to fine-tune any medical image segmentation technique. This approach seems to be capable of achieving the goals of the study because it automatically assigns parameters and integrates expert feedback in order to provide intelligent guidance of the segmentation process.

1.4 Organization

The thesis is organized as follows. In Chapter 2, a survey of existing image segmentation techniques is presented along with a background review of fuzzy systems, the scale invariant feature transform (SIFT) technique, and evolving fuzzy systems. In Chapter 3, the proposed evolving fuzzy segmentation techniques are explained. Chapter 4 contains a description of the experiments and their results. Chapter 5 includes the conclusion and suggestions for future work.

Chapter 2

Segmentation, Features, and Fuzzy Systems

2.1 Image Segmentation

Segmentation refers to the process of partitioning a digital image into multiple regions, which are sets of pixels, also known as super pixels, in order to simplify and/or change the representation of an image into a format that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images [2]. If R denotes the input image, then image segmentation means that R is portioned into n regions, such that $\bigcup_{i=1}^{n} R_i = R$, where R_i is a connected set, and $R_i \cap R_j = \emptyset$ for $i \neq j$ [1].

A large number of medical image segmentation techniques are available. Pham et al. [4] divide medical image segmentation methods into eight categories: thresholding approaches, region growing approaches, classifiers, clustering approaches, Markov random field models, artificial neural networks, deformable models, atlas-guided approaches, and other approaches, such as the watershed method. Hu et al. [12] divide medical segmentation methods into four general categories: region-based approaches, boundary-based approaches, hybrid approaches, and atlas-based approaches. The following sections provide a brief review of selected relevant segmentation methods: thresholding, clustering, neural nets, deformable methods, watershed, and graph cut segmentation. Also described are the scale-invariant feature transform (SIFT), fuzzy

systems, and evolving fuzzy systems. Features that can subsequently be employed to train a fuzzy system can be extracted from images through the use of SIFT.

2.1.1 Thresholding Methods

Image thresholding can be viewed as the simplest technique for image segmentation because it separates an object from its background based on the gray levels in the image [6]. Thresholding is used in many applications because of its simplicity and speed. Let image $I = f(x, y)$, where $f(x, y)$ is the gray level of I at point (x, y), a thresholded image $g(x, y)$ using a threshold T is defined as

$$g(x,y) = \begin{cases} 0 & \text{if } f(x,y) > T, \\ 1 & \text{if } f(x,y) \leq T. \end{cases} \tag{2.1}$$

Within this technique are included a vast number of methods for thresholding gray-level images. Sezgin et al. [6] provide a survey of the most popular thresholding methods and categorize them based on the information used to differentiate six classes:

- **Histogram-shape-based methods**
  These methods are based on the determination of the threshold using the shape of the histogram whereby the peaks and valleys of the histogram are taken into consideration for the purpose of detecting the threshold. Many algorithms use the shape of the image histogram, but one of the popular methods is the Rosenfeld [23] method, in which the threshold is the deepest concavity identified by the calculation of the convex hull of the histogram.

- **Clustering-based methods**
  With these methods, the gray values of the image histogram are divided into two clusters: foreground and background. The most popular clustering algorithm is the Otsu method [22], in which the image histogram is divided into foreground and background clusters using a threshold that minimizes the variance between these two classes. Kittler et al. [24] also tried to obtain the minimum thresholding error by minimizing the error rate in the classification of object and background.

- **Entropy-based methods**
  In these methods, the detection of the threshold of the images is based on the level of entropy. The maximum sum of the entropy in the foreground and background regions is considered the threshold in [25], and the fuzzy membership of the gray level in the foreground and background is used in [26].

- **Object-attribute-based methods**
  These methods entail the calculation of the threshold based on a determination of similarities between the image and its binary form. The most popular algorithm based on this method is that proposed in [27], which uses fuzzy similarity to detect the threshold.

- **Spatial methods**
  These methods are based on the location of the pixels. According to [12], the most effective technique is to use a co-occurrence matrix to calculate the threshold [28].

- **Local methods**
  The algorithms used in these methods calculate the threshold for each pixel from statistics such as variance or the surface-fitting parameters of the pixel neighbourhood. The most popular method was proposed by Niblack [29], in which the mean and standard deviation of a window are used to calculate the threshold.

Most thresholding techniques are based primarily on the histogram of the image. Because the shape of the histogram is not identical for all images and because detecting an accurate threshold is difficult with a unimodal histogram, most of these methods work well only when the histogram is bimodal or multimodal. Other methods not dependent on histograms either use a static scheme for all images, that is, one without any ability to adjust to the features of the image, or are based on several parameters that must be tuned separately for each image in order to obtain maximum accuracy.

Numerous studies have been conducted with the goal of overcoming the limitations of thresholding techniques. Tizhoosh [30] proposed a new thresholding technique based on type II fuzzy sets. Liu et al. [31] used the entropy approach to calculate image thresholds. However, they assumed that the histogram of the image is multimodal. Bazi et al. [32] used an expectation-maximization algorithm to find the optimal global threshold of the image based on an estimation of the statistical parameters of the object and on the use of background classes that follow

generalized Gaussian distribution. This method has a lengthy computation time and produces very poor results when the histogram of the image is monomodal. Rather than relying on the optimization of one criterion, Nakib et al. [33] used the histogram of the image and a multi-objective optimization approach in order to determine the optimal thresholds of three criteria: a within-class criterion, a level of entropy criterion, and an overall probability of error criterion. This method generates good results compared with other popular thresholding techniques but is parameter-dependent. Kamal et al. [34] used a genetic algorithm in order to generate from the histogram a threshold that has been compressed using a wavelet transform. Although successful in some cases, this method is time-consuming. Wang et al. [35] proposed that the optimal threshold could be obtained by optimizing a criterion function created by applying the Parzen window technique to the image histogram in order to estimate the spatial probability distribution of the gray-level image values. The main difficulty with this method is the very long computation time required for calculating a global threshold. Nakib et al. [36] used a digital fractional differentiation of the image histogram in order to calculate the threshold of the image. This method depends on an assumed correlation of the probabilities of the gray levels in the image, which is not the case for all images. As well, the process of selecting the order of fractional differentiation is complicated and does not work well for all images. Li et al. [37] proposed a new statistical image thresholding method that takes into account not only the sum of the variance, as with the Otsu method, but also the discrepancies between the variances. The difficulty with this method is selecting the parameters used to control the weights assigned to the class variance sum and to the variance discrepancy. As extensions of the Otsu method and the Kittler method, Xue et al. [38] proposed two different approaches using median-based criteria. The problems with this method are the inability of the algorithm to detect an efficient threshold if the histogram is multimodal and the increase in the computional time resulting from the time required for calculating the median.

Othman and Tizhoosh [39, 40, 41] proposed methods using neural networks and evolving fuzzy rules for overcoming the problems associated with thresholding in medical images.

Although a great deal of effort has been dedicated to solving the problem of thresholding, many challenges remain. Most methods still depend on the shape of the image histogram, which prevents the generalization of the methods. On the other hand, methods that are not dependent on the shape of the image histogram require complicated operations that increase the computational

time required for the calculation of the threshold. Most methods also depend on the tuning of several parameters, all of which must be altered for each image in order to achieve a high degree of segmentation accuracy. In general, no method works well for a varied types of images because the lack of consideration of the features of the image itself and the disconnection from the visual perception of the user. A need therefore exists for a thresholding method that can dynamically adapt to a variety of images by taking into consideration both the features of the image and user feedback.

2.1.2 Region-Based Segmentation Methods

Region-based methods are derived from the suggestion that neighbouring pixels within the same region have similar intensity values and involve one of two different procedures: region growing, or split and merge.

The general region-growing procedure is to compare one pixel to its neighbours. If a similarity criterion is satisfied, the pixel is assumed to belong to the same region. The algorithm starts with a seed point selected manually inside the region of interest and begins to compare the intensity of this point to that of its neighbours. If a predefined threshold is satisfied, the new point can be added to the region. Several criteria have been suggested, such as the distance from the seed point or the average gray-level intensity. Region growing methods are popular techniques for image segmentation. However, these methods involve two major challenges: finding the seed point in the region of interest and determining the threshold value for stopping the growth process. Region growing can also be computationally expensive.

A method that does not require a predefined seed was suggested by Lin et al. [42]. They start with any point inside the image as a seed point and begin to grow from this point as long as the neighbours of the point satisfy a similarity criterion. If the algorithm encounters a point that does not satisfy the criteria, it considers it a new seed for a new region. This method increases the computational time because it requires the scanning of the whole image, and the similarity threshold problem remains unsolved.

Stewart et al. [43] combined a pulse-coupled neural network with a region growing algorithm in order to solve the problem of selecting seed points. The difficulty with this method is that both the similarity function and the similarity threshold must be tuned for each image. Lu et

al. [44] proposed a simplified version of the algorithm presented in [43]. They modified the linking channel function and decreased the complexity of the adjusting parameters. However, the parameters remain the same and must still be tuned to accommodate different images.

Malek et al. [45] proposed a method of segmenting microcalcifications in breast images by means of the automatic selection of the initial seed points, which represent image regions based on mathematical morphology. Their method uses a similarity criterion that calculates the difference between the intensity of each pixel in the image and the mean intensity of the region. The pixel with the smallest difference is then allocated to the corresponding region. This method is case-based because it depends primarily on the brightness of the microcalcifications for the selection of both the seeds and the similarity criteria. Park et al. [46] proposed a new skull-stripping method for analyzing magnetic resonance brain images. Rather than selecting seed points, they suggested beginning from two seed regions that represent the brain and non-brain regions and then using a mask produced through morphological operations in order to detect these two regions. A region growing method is then employed to provide expansion from these two regions. Their method is customized for information related to brain anatomy, and their experimental results show that the similarity threshold must be tuned for different images in order to obtain a satisfactory degree of segmentation accuracy.

The researchers mentioned as well as many others have tried to solve the two major problems related to region growing. Most of the approaches described have been directed at finding a solution to the selection of a seed point, but the suggested solutions are limited by their lack of generalization or by the need for tuning parameters. A need thus still exists for a satisfactory general method of detecting the seed point and of adjusting the similarity threshold used in region growing.

In contrast to region growing, the split and merge technique begins with the whole image, which is divided into disjointed parts so that similar regions can be merged based on similarity criteria. The primary difficulty with this method is the challenge of selecting both the similarity function and the similarity threshold. A recent technique [47] uses statistically based region merging. In this technique, the image segmentation problem is viewed as an inference problem, which then leads to a simple merging predicate and a simple ordering of the merges. However, to obtain maximum accuracy, this method has several parameters that must be tuned for each image such as the scale of the segmentation.

2.1.3 Classification and Clustering

Classifiers are supervised methods based on training images that are segmented manually by an expert. This category included numerous techniques, such as nearest-neighbour, K-nearest-neighbour, the Parzen window, the maximum-likelihood, and the Bayes rule. Clustering methods, on the other hand, are unsupervised methods that depend on the clustering of the available image into different groups. The following are the two commonly used clustering techniques [4, 12]:

- K-means is a clustering technique whose operation requires the number of clusters so that a centre can be assigned for each cluster. All image pixels are classified as belonging (= 1) or not belonging (= 0) to a cluster, based on a selected criterion related to the distance between the pixel and the centre of the cluster.
- Fuzzy C-means is similar to K-means but with a major difference: any image pixel can belong to more than one cluster based on a selected membership function.

2.1.4 Segmentation Using Neural Networks

Due to the large number of differing neural architectures on the one hand, and the proliferation of varying learning schemes on the other, providing an in-depth review of all the neural approaches to image segmentation is difficult. Using a constraint satisfaction neural network, Faith et al. [48] have provided a survey of image segmentation based on relaxation. Glass et al. [49] used a self-organizing map for the process of segmenting brain images into different regions. They also trained a multilayer backpropagation neural network to classify and label the segmented region of the brain. Chaun et al. [16] segmented medical images using a three-dimensional Hopfield neural network architecture based on contextual constraint. Nunez et al. [50] introduced an algorithm for segmenting astronomical images based on self-organizing neural networks and wavelets. They used the second plane of the wavelet decomposition to separate stars and other prominent objects, and the network was then employed to segment the rest of the image. Lee et al. [51] used artificial neural networks for the segmentation of swallow from healthy participants. The network is trained to accept signals from both single and combined sources, and the

accuracy of the segmentation is at its highest level when the network accepts input from combined signal sources. Victor et al. [17] used two different artificial neural networks to segment micro-structures from metallo-graphic images. The challenge with such methods that depend on a self-organizing map is the large number of parameters that must be tuned before the training, such as the number of classes and the size of the neigbourhood function. As well, the latter three methods depend primarily on prior information about the image, which renders their generalization very difficult.

In [18], Nadir et al. extended their incremental neural network proposed in [52] in order to segment tissues in ultrasound images using a discrete Fourier transform and a discrete cosine transform (DCT) for feature extraction. Using DCT and vector quantization, Dokur [53] employed an incremental self-organizing map to compress and segment medical images obtained by magnetic resonance, computed tomography, and ultrasound. Iscan et al. [54] presented a method that uses an incremental supervised neural network for segmenting tissues in medical images. They employed a continuous wavelet transform and the moments of the gray-level histogram in order to compute a features vector for medical images. Although these methods are incremental in the sense that the number of nodes is not predefined, a large number of parameters still must be tuned in order to accommodate different images. A threshold value that controls the number of nodes, and the gain constraint of the network when DCT is used must both be tuned as appropriate for different images. As well, the process of selecting the region of interest used to extract the features is time consuming because it is performed manually by an expert.

Fu et al. [55] used a pulse-coupled neural network along with a statistical expectation-maximization (EM) model in order to develop adaptive techniques for the segmentation of brain images. The EM technique is applied to the image histogram in order to estimate initial thresholds for segmenting different regions of the brain and for adjusting the parameters of the neural network. The difficulty with this method is the large number of parameters that must be tuned, such as the watershed threshold used in the enhancing stage. In addition, the threshold used to segment portions of the brain is generated from the image histogram, which is not to be trusted when the shape of the histogram is unclear.

2.1.5 Deformable Methods

He et al. [56] have provided a comparative study of snake and level set methods. Snake methods included the balloon snake, the topology snake, the distance snake, and the gradient vector flow snake. For the level set methods, they reviewed the original level set method, the geodesic active contour method, the area and length active contour method, and the constrained optimization method. Due to their popularity, only level set methods are briefly reviewed in the following subsection.

Level Set Segmentation

The level set method is a tool for modeling time-variant objects because it uses numerical techniques for tracking shapes. The method is based on the initialization of a contour inside or outside the region of interest, which then evolves toward the optimum segment. The curvature of the contour and the features of the image are used to calculate the speed of the deformation of the contour [56].

In traditional level set methods [57], to overcome the problems associated with a sharp and/or flat shape generated during the evolution, a common numerical scheme initializes a function as a signed distance function prior to any evolution. This scheme periodically reshapes (or re-initializes) the function so that it is a signed distance function during the evolution. The re-initialization process is therefore crucial, cannot be avoided in traditional level set methods. Chan et al. [58] proposed a new level set method, in which the re-initialization process is optional. This important method can detect objects whose boundaries are not defined by a gradient. Li et al. [59, 60] completely eliminated the need for the re-initialization procedure by forcing the level set function to be close to a signed distance function and presenting a new variational formulation for geometric active contours. Their formulation consists of an internal and an external energy term. The internal energy term penalizes the deviation of the level set function from a signed distance function. The external energy term drives the motion of the zero level set toward the desired image features, such as object boundaries. They managed to minimize the overall energy function using the gradient flow generated from the evolution of the level set function. Lankton et al. [61] proposed a new method that reformulated any region-based segmentation energy in a local way, and used local information to cause the contour to evolve. Shi et al.

[62] proposed a new level set algorithm suitable for real-time implementations. They eliminated the need for solving partial differential equations by using a two-cycle algorithm: one cycle for the data-dependent term, and a second cycle for the smoothness regularization. Bernard et al. [63] proposed a fast level set method that relies on a continuous parametric implicit function expressed as a B-spline rather than on the use of a discrete representation of the implicit function. Yan et al. [64] used a level set framework to develop a new hybrid medical image segmentation method, which achieves robust and accurate results through the use of the boundary of the object as a means of detecting the precise location of the target object, and information about the region as a means of preventing boundary leakage. Bo et al. [10] proposed a novel model based on a level set for segmenting breast ultrasound images. They used the differences between the actual and estimated probability densities of the intensities in different regions to develop an energy function. The actual probability densities are calculated directly, and a level set approach is used to formulate the energy function, along with a partial differential equation approximated by the central difference and non-re-initialization approach for finding the minimum of the energy function. Chen [65] used statistical decision theories such as conditional average risk and Bayesian risk in order to derive a level set evolution function. The author classified pixels and applied the losses in pixel classification to form the Bayesian risk, which is minimized and utilized to generate the evolution function. Achuthan et al. [66] proposed a level set method for segmenting similar regions in computed tomography images. They used a wavelet transform in order to adapt a set of features to represent different regions inside the image. The features are then used in a level set segmentation technique in order to segment the regions.

Although several variations in level set methods have been proposed for medical image segmentation, they all entail two main challenges: defining the initial segment/polygon, and adjusting multiple parameters (e.g., for smoothness regulation, balloon force, penalty term, and time step). Moreover, a set of parameters are associated with specific methods, such as propagation parameters in [57], the curvature parameter in [58], curvature and radius parameters in [61], the scale parameter in [63], and the Gaussian filter and weight evolution parameters in [62].

In an attempt to solve the problem of determining the initial contour, Li et al. [67] proposed a method that incorporates the results of the fuzzy C-means (FCM) segmentation as an initial contour for segmenting liver images. They extended their method in [19] by using the FCM to provide a guess about the set of parameters that control the level set. The difficulty with this

method is that it relies mainly on an FCM method for clustering the image and dividing it into regions, a method that could be efficient for a carotid artery ultrasound or liver images that have clear edges and small number of objects but not for medical images that involve a large number of objects or unclear edges.

2.1.6 Graph Cut Segmentation

A graph cut (GC) is a recent technique for image segmentation. The general idea of GC techniques [11] is to treat the image as a graph $G = (V, E)$ whereby each pixel of the image is represented as a graph node $v_i \in V$, and the connection between two pixels is represented as the graph edges $(v_i, v_j) \in E$. Each edge in the graph is associated with a weighted function $w(v_i, v_j)$ of the similarity between pixels. This weighted function becomes the basis for partitioning V into a set of disjoint regions based on thresholding. Weiss [68] proposed a review of GC segmentation methods that are based on the eigenvectors of the affinity matrix. The most efficient technique involving eigenvectors is the normalized cut (N-cut) technique proposed in [69]. These methods are very slow and involve the tuning of numerous parameters that affect the accuracy of the segmentation, such as the weight function and the threshold used in the weighted function.

As an alternative category of GC segmentation, Boykov et al. [70, 71, 72, 73] proposed an interactive GC method that asks the user to identify a set of object seeds and a set of background seeds. They suggested adding two additional terminal nodes, a source node and a sinking node, which are connected with an edge-weighted function to object seed nodes and background seeds, respectively. This weighted function calculates how close in intensity each pixel is to the source and to the sink terminals. Each pixel node is also connected to its neighbours with a weight function so that every pixel node in the graph thus has three edges: two for the two terminals and one with its neighbours. The authors suggested cutting the graph and separating the two terminals by minimizing a cost function of the boundary $B(A)$ and region $R(A)$ properties of segment A using a min-cut max-flow algorithm:

$$E(A) = \lambda R(A) + B(A), \tag{2.2}$$

where the constant λ determines the importance of the region properties term $R(A)$ versus the

boundary properties term $B(A)$ [72].

Unfortunately, this technique presents problems, especially for use with medical images. First, it requires intensive user interaction in the marking of multiple objects and background seeds, and after the results have been obtained, even greater user interaction is required in order to correct the segment with additional marking of the object and background. Second, medical images lack strong edges between objects and contain multiple objects with very similar intensities, both of which features lead to errors in the results generated by this technique. Because the user marks the object and background seeds, when multiple objects have similar intensities, a multi-object segment is generated, which is considered a bad segment. Third, the energy and the weighted functions contain parameters such as λ in equation 2.2, which must be tuned for each different image. Finally, the min-cut max-flow algorithm necessitates the construction of large graphs that consume a great deal of memory.

Many researchers have proposed solutions that overcome these problems, including combining a pre-segment technique such as a watershed with a GC technique [74], or blending watershed and iterative region-merging methods [75], or using an iterative GC [76], or employing the prior shape of the object to address the problem of similar objects [77], or relying on a recursive minimum description length to represent the image [78] or to reduce the graph size [79]. These techniques may not overcome all of the problems associated with the GC technique because they still require user interaction and parameter adjustment, but this method remains relatively popular due to its underlying sophisticated theory [80, 81, 82].

In [11], the authors proposed an efficient GC technique that depends on a predicate weight function for detecting whether a boundary exists between two regions in the image. The boundary is considered not to exist, and the two regions are merged if the weight of the edge between these two regions is smaller than the internal differences in both regions; otherwise, the two regions remain separate. The weight function of the edges is calculated based on the distance between pixels, and the internal difference in a region is taken as the maximum weight inside the region. This technique does not require the initialization of seeds in the foreground and background, and clusters the image into different regions. An extension of this work that makes use of the statistical information about each sub-region was proposed in [83]. The drawbacks of this method are that it employs a Gaussian filter for the image using different values of the standard deviation (σ) and that the predicate is controlled by a threshold that determines the degree of

difference between the boundary weight and the internal weight. The accuracy of the resulting segment is therefore affected by the tuning of these parameters.

Normalized Cut (N-cut)

A review of the GC segmentation methods that are based on the eigenvectors of the affinity matrix is presented in [68] . However, the most efficient technique involving eigenvectors is the N-cut technique proposed in [69]: the authors suggest thresholding a generalized eigenvector of a normalized affinity matrix rather than using the first eigenvector of the affinity matrix as a means of cutting the image into segments. In this method, the image is treated as a graph that is segmented using an N-cut criterion, which calculates the similarity within the group as well as the dissimilarity between different groups and is optimized using a technique based on a general eigenvalue problem. A weighted undirected graph $G = (V, E)$ is constructed from a set of points (graph nodes) in an arbitrary feature space, and a weight function of similarity $w(i, j)$ is built on the edges between node i and node j. The graph is segmented based on a grouping approach, in which the set of vertices $V_1, V_2, \cdots V_m$ is divided into different sets in which the similarity between the vertices within the same set is high, and the similarity between different sets is low. The image is then divided using eigen vectors. In general, the graph $G = (V, E)$ is divided into two sets: A, B where $A \cup B = V$, and $A \cap B = \emptyset$ by removing the edges connecting the two parts. The total weight or degree of dissimilarity, called the cut between the two pieces, can be computed as follow:

$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v). \tag{2.3}$$

This cut value should be minimized in order to achieve the best partition of the graph. In [69], a new measure of disassociation, or N-cut, is proposed, which computes the cost of the cut as a fraction of the total edge connections to all nodes in the graph:

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)}, \tag{2.4}$$

where $assoc(A, V) = \sum_{u \in A, t \in V} w(u, t)$.
In the same manner, a measure for total normalized association within the groups for a given

partition could be defined as follows:

$$Nassoc(A, B) = \frac{assoc(A, A)}{assoc(A, V)} + \frac{assoc(B, B)}{assoc(B, V)}, \tag{2.5}$$

where $assoc(A, A)$ and $assoc(B, B)$ are the total weights of the edges connecting nodes within A and B. In the grouping algorithm, the two criteria are therefore that the association within the groups should be maximized and that the disassociation between the groups should be minimized. In the N-cut algorithm, this normalized cut is used as the partition criterion. In general, as stated in [69], the grouping algorithm consists of the following steps:

- Construct a weighted graph $G = (V, E)$ for every image, with the weight on the edges between nodes being the measure of similarity.
- Solve $(D - W)x = \lambda Dx$ for eigenvectors with the smallest eigenvalues, where D is an $N \times N$ diagonal matrix with $d(i) = \sum_j w(i, j)$ on its diagonal, W is an $N \times N$ symmetrical matrix with $W(i, j) = w_{ij}$ and $N = |V|$, and x is an indicator vector.
- Use the eigenvector with the second smallest eigenvalue to divide the graph by finding the splitting point such that the N-cut is minimized.
- Determine whether the current partition should be divided by checking the stability of the cut and ensuring that the N-cut is below the prespecified value.
- Recursively repartition the segmented parts if necessary.

This method entails the tuning of numerous parameters that affect the accuracy of the segmentation, such as the number of segments N_S, the rate R_A, the radius R_D, and the edge variance E_V.

2.1.7 Watershed Segmentation

In geography, the ridge that divides the areas between different rivers is called a watershed. Watershed segmentation incorporates this concept in the segmentation of regions into different gray scale images. Watershed segmentation techniques include those based on a watershed

transformation, a watershed that uses a distance transform, a watershed with gradients, and a marker-controlled watershed [1]. Letteboer et al. [84] present a method for segmenting tumours in brain images using an interactive multiscale watershed segmentation technique. Jung et al. [8] incorporated watershed segmentation as the second stage of a new segmentation scheme after a stage involving multiresolution wavelet decomposition. Cristoforetti et al. [9] used a marker-controlled watershed segmentation technique to segment the left atrial surface in multi-detector cardiac images. Kuo et al. [85] combined watershed segmentation with a competitive Hopfield clustering network for the segmentation of brain images. Hamarneh [86] improved the watershed algorithm used in medical image segmentation by employing prior shape knowledge. Hsu [87] employed vector quantization and a competitive Hopfield neural network in order to compress the tumour and breast regions in mammogram images that had been previously segmented by means of an improved watershed transform.

2.2 Scale-Invariant Feature Transform

The scale-invariant feature transform (SIFT) is an approach developed by Lowe [88] as a method of object recognition. It is based on the detection of a set of key points for an object and the subsequent calculation of a set of descriptors, or features, for these points. These features, invariant to image scale and rotation, can enable the differentiation of one object from a group of different objects. The process of detecting key points and calculating the descriptors consists of the following four stages:

- **Scale-space extrema detection**
  In this stage, the points of interest called key points are detected. First, Gaussian filters at different scales are generated and then convolved with the image at every scale. The difference of Gaussians (DoGs) blurred images are determined, and the candidate key points are assigned as the maxima and minima of the DoGs $D\left(x, y, \sigma\right)$ at multiple scales. To create the scale space of an image, the convolution of a Gaussian filter $G(x, y, \sigma)$, with an input image $I(x, y)$, is determined and defined as a function $L(x, y, \sigma)$, where

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y), \tag{2.6}$$

and (DoG) is given by

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y). \tag{2.7}$$

The maxima and minima of the DoGs are calculated through a comparison of each point with its eight neighbours in the current image and nine neighbours in the scale above and below (Fig. 2.1 [88]). The point is selected to be a key point if it is larger or smaller than all of these neighbours.

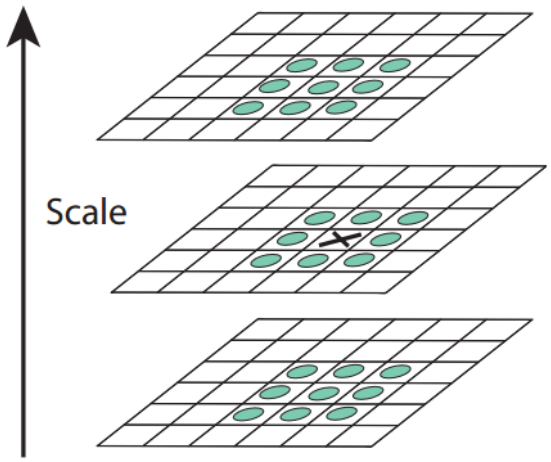

Figure 2.1: Process for detecting key points: The pixel indicated by the cross is compared to all of the circled pixels.

- **Key point localization**
  A substantial number of candidate points are detected in the previous step; however, many of these points are useless due to either low contrast or to poor localization across the edges. Therefore, for each candidate point, a measure of stability is calculated so that points with low contrast or poor localization along the edges can be discarded.

- **Orientation assignment**
  Local image gradient directions are used as a means of assigning one or more orientations

to each keypoint location in order to ensure that the candidate points are invariant to orientation. The orientation of the keypoint is calculated based on the orientation and magnitude of the pixels in a region around the keypoint.

- **Key point descriptor**
  After the candidate points have been selected, the local image gradients are measured in the region around each key point. Orientation histograms are created from 4×4 pixel neighbourhoods with eight bins each, and the magnitude and orientation values of the samples in a 16×16 region around the key points are calculated. When $4 \times 4 = 16$ histograms exist, each with eight bins, the feature vector of each key point with 128 elements is calculated.

SIFT has been used in numerous applications, such as object recognition, image stretching, 3D modeling, gesture recognition, video tracking, match moving, and face authentication [89] as well as to provide a self-calibration strategy for estimating intrinsic and extrinsic camera parameters [90]. A number of published reports [91, 92, 93, 94] help provide an understanding of the SIFT technique. Details about the SIFT technique are provided in Appendix A.

2.3 Fuzzy Image Segmentation

Relatively popular methods of image segmentation, fuzzy methods are based on the interpretation of the image, its histogram, or features, all considered as fuzzy sets. This section provides a review of samples of these methods, which are similar to the approach presented in this thesis. The following are general categories of fuzzy image segmentation methods [30]:

- **Fuzzy clustering methods**
  These methods cluster (i.e., classify) all image pixels into different segments, or in the specific case of thresholding, into foreground and background pixels. The fuzzy clustering method most commonly used is fuzzy c-means (FCM) [95, 96, 97, 98, 99]. An additional recent development is collaborative fuzzy clustering [100].

- **Rule-based methods**
  These methods use fuzzy if-then rules to determine a threshold. The rules may be defined by an expert or generated from data with continual updates [101, 39].

- **Fuzzy-geometrical methods**
  These methods minimize or maximize fuzzy geometrical measures, such as compactness, in order to binarize the image. In contrast to other fuzzy techniques, fuzzy geometry is generally focused on local image information [102, 103].

- **Information-theoretical methods**
  These algorithms minimize or maximize fuzzy information measures such as fuzzy entropy, index of fuzziness, and fuzzy divergence. Implemented primarily globally, operating on image histograms, these methods are relatively fast [104, 27, 105].

- **Type II thresholding methods**
  These methods interpret image information as Type II, or interval-valued, fuzzy sets and may use information-theoretical measures such as fuzzy entropy or the index of ultrafuzziness to locate a global threshold [106, 107].

With the goal of solving the problem associated with the manual selection of the parameters of the fuzzy membership, which was encountered in previous fuzzy segmentation techniques, Karmakar et al. [13] proposed a segmentation technique based on a generic fuzzy rule. Their proposed technique is application independent and incorporates the spatial relationships between pixels. Cinque et al. [15] suggested a modification to the existing fuzzy approach for image segmentation: the adaptive reasoning theory (ART) model. They maintain that their proposed approach solves the problems inherent in the existing ART model, such as overfitting, complexity, and the inability to measure the extent of the match between any two patterns. However, this method still does not address the difficulty of tuning network parameters, such as the learning and vigilance parameters, which affect the results for different images.

Fuzzy C-means (FCM) has been used as a popular clustering algorithm for image segmentation. Zhang et al. [108] suggested a new algorithm that uses a kernel-induced distance metric rather than the original Euclidean distance in the FCM and adds a spatial penalty to the membership functions in order to modify the objective function in the FCM. They use their approach for the segmentation of magnetic resonance (MR) imaging data. Chuang et al. [109] provided a new version of the FCM that uses the spatial information about the image for segmenting medical images. They created a spatial function by combining the spatial information with the membership function. However, tuning is required for some of their parameters such as the parameters

that control the importance of the functions. Yang et al. [110] proposed a new fuzzy clustering algorithm for managing noisy images, which is a modification of the existing FCM segmentation algorithm. Zhuge et al. [111] developed image segmentation based on fuzzy connections in order to segment vectorial images. Their technique extends previous work on scalar images to include vectorial images, and they applied their method to arbitrary dimensional vectorial MR medical images. However, several of their parameters must also be tuned, such as the percentile value used for estimating the parameters. Zhao et al. [112] proposed a method that makes use of the spatial information about the image in order to overcome the sensitivity of the FCM to noisy images.

Methods that depend on an FCM technique have failed to address the main problems associated with FCM: the setting of the threshold so that cause the method converges for different images as well as the determination of the number of clusters and the cluster radius, all of which must be tuned in order to enhance the clustering. The setting of these parameters generates different results for different images.

Dou et al. [113] proposed a fuzzy fusion technique for automatically segmenting tumour areas of MR images of the human brain. They suggested building a separate fuzzy model for each type of MR image about which prior information has been obtained. These separate models are then fused into one model. However, this method seems to be specific to the segmentation of brain images because it depends on prior information provided by an expert in the nature of the images. It also uses a region growing method with a threshold that must be tuned for different images. Hata et al. [114] proposed a fuzzy technique for the segmentation of endorrhachis in MR images. Their algorithm checks the possibility of the existence of endorrhachis regions in the image and assigns a high fuzzy degree to a higher possibility. The researchers also described a fuzzy maximum intensity projection technique that maps higher fuzzy membership degrees to brighter values in the data set. This method seems to be a case-based customization for segmenting only endorrhachis in MR images. Foo et al. [115] proposed an image segmentation method based on a fuzzy rule in order to segment tumours in three-dimensional computed tomography (CT) data. Their method relies on the initial selection of a region of interest by the user, and a set of parameters must also be tuned, such as the similarity threshold used to segment the tumours. Khotanlou et al. [116] proposed an automated method for segmenting tumours in brain images. They employed fuzzy classification for the initial segmentation of the tumours, followed by the

use of a parametric deformable model controlled by fuzzy relations for the final segmentation. However, this method is based on two assumptions: that the brain is symmetrical in shape and that the tumour has a specific gray level.

Many fuzzy segmentation techniques use fuzzy rules in order to segment images or to combine fuzzy rules with other techniques such as the principal component analysis approach. In addition to the previously described challenges that still must be addressed, a method is also required for updating fuzzy rules to enable online segmentation. Such an approach might help improve the accuracy of the segmentation and might also reduce the time required for updating the rules manually.

2.4 Evolving Fuzzy Systems

The possibility of designing an optimal fuzzy system that could be used in any application has been investigated in numerous studies. Developing an efficient fuzzy system requires effective methods of determining three factors: the selection of the fuzzy rule base, the definitions of the membership functions, and the structure of the fuzzy system (number of rules and membership functions) [117]. Many researchers have attempted to address these issues automatically in order to generate efficient fuzzy systems, and the following processes have been investigated as possibilities: the swarm optimization technique [118] and the application of evolutionary computation directly to the data structure of the fuzzy logic system rather than using binary representation [117]. Genetic algorithms have been employed as a means of enabling the type and the shape of membership functions as well as the number of rules to evolve [119], for example, in forecasting financial time series data for stocks [120]. The Pittsburgh and Michigan schemes are well-known approaches that facilitate the evolution of the rule set using genetic algorithms. The Pittsburgh approach enables the rule set to evolves as a whole whereas the Michigan method deals with individual rules [121, 122]. A hybrid approach combining the Pittsburgh and the Michigan techniques has also been proposed for classifying tissue in MR images [123], and more recently, a hybrid model of self-organization maps as well as a genetic algorithm and a fuzzy rule-based system have appeared in the literature [124].

These methods are focused on the automatic generation of fuzzy parameters from the input data in order to design a fuzzy system. However, they still represent a static paradigm because

they do not make use of new data that becomes available online in order to update or expand the knowledge contained in the rule base, which would increase the accuracy of a fuzzy system because the membership functions would be extended to cover larger areas.

Attempts to make use of online data could involve the evolution of fuzzy rules through a variety of methods that either combine them with neural networks or incorporate genetic algorithms or other techniques for updating fuzzy rules online. Linden et al. [125] used evolving fuzzy rules and a genetic algorithm in which the rules are represented as an expression in reverse Polish notation in order to extract rules from microarray data.

Kasabov [126, 127] proposed evolving fuzzy neural networks (EFNN) that can update their structures online as new data arrive. This network can learn and perform reasoning, rule extraction, and aggregation online. EFNN uses locally tuned elements to perform the process of incremental online learning. When more data are available, the network must be updated by pruning or by the aggregation of the nodes. For use in prediction applications, Lin et al. [128] proposed a self-adaptive neural fuzzy network with a group-based symbiotic evolution process. Their approach contains an online self-clustering algorithm that addresses the internal structure in which the number of clusters (fuzzy rules) is equal to the true number of clusters in a given training data set. A group-based symbiotic evolutionary learning method is used to adjust the parameters in order to produce the desired output. Juang et al. [129] have provided a dynamic system processing approach that uses a recurrent self-evolving fuzzy neural network with local feedback. Their method is an online technique that involves a learning process for determining the structures (generating the fuzzy rules online) and the parameters. A Kalman filter is used to enable the system to learn the consequent parameters, and a gradient descent algorithm determines the antecedent parameters. Rong et al. [130] applied the functional equivalence between a radial basis function network and a fuzzy inference system in order to develop a sequential adaptive fuzzy inference system. Their proposed algorithm is an adaptive technique that depends on the data input in order for a determination to be made about whether to add or remove a fuzzy rule to or from the fuzzy rule base.

EFNNs have shown promise in a variety of applications, ranging from the extraction of fuzzy rules associated with cancer-related genes [131] to the forecasting of sales of printed circuit boards [132].

Previous methods attempted to make use of online input data in order to increase the range

of a system. However, all of these methods relied on neural networks that update their structure online as new data arrive, and they still involve a number of parameters that must be assigned so that decisions can be made with respect to adding a new node or updating or replacing an existing one in the neural network. This process seems to be infeasible for image segmentation because tuning the parameters requires substantial time and effort, especially when a large number of images are involved.

Evolving fuzzy systems are a rather recent development that has been used for a variety of applications: fault detection, intelligent sensors, robotics, and internet-based information mining. They are especially helpful when new data can be used to expand and improve core capabilities, such as in the segmentation of medical images. According to [20, 21], an evolving fuzzy system is an unsupervised method of updating the rule-base structure of a fuzzy inference system in a non-iterative way. As additional information becomes available, the rule base evolves by adding more rules to the basic ones that formed the initial fuzzy model. The existing rules might also be replaced with new rules if the new ones are more descriptive. The evolution of rules in a fuzzy inference system can occur in a variety of ways. As long as the initial fuzzy model (set of rules) is continually updated, no preference exists with respect to how the actual update is performed. The only reason for favouring one evolutionary strategy over others is that the preferred strategy has demonstrated superior memory and speed characteristics. The term "evolving" may imply that rather than retraining the initial system, the system should be updated online when new data becomes available. The learning process can start from scratch: the first point in an online dataset can be treated as the centre point of the cluster, and the sum of the contributions of the Euclidean distances between that point and all other data points is called the potential of that point. When data is captured online, its potential is calculated and the potentials of existing cluster centres are recursively updated. The potential of the new data is compared with the existing centres and one of the following decisions is made:

- Replace one or more existing clusters with the new data point if its potential is higher than a specific threshold and if the new data point is close to an old centre.
- Add the new data as a new cluster if its potential is higher than a specific threshold.

In [133], a Takagi-Sugeno (TS) type of fuzzy controller with an evolving structure was proposed as a new technique for the online identification of TS systems. The fuzzy rules of the structure

of the controller are expanded through the use of data collected during the control process. The fuzzy controller is thus trained in a non-iterative, recursive way. The interpretability of evolving fuzzy systems is discussed in [134]. Recent approaches based on evolving fuzzy systems have also been proposed in [135, 136], and in general, the idea of evolving fuzzy systems is finding its way into a new range of applications, such as high dimensional problems [137], fault detection [138], the modeling of non-linear dynamic systems [139] and image classification [140].

2.5 Feature Selection

Feature selection is an important step in any machine learning process. Providing relevant features to the system will increase its ability to learn and elevate its performance. Feature selection is the process of selecting the most relevant features out of a larger group of features so that either redundant or irrelevant features are removed [141]. Redundant features add no new information to the system, and irrelevant features may confuse the system and decrease its ability to learn efficiently. In mathematical formulation, feature selection could be defined as follows: if Y is a set of all features, then feature selection is the process of calculating a weight w_i for each feature $y_i \in Y$ in order of relevance. Feature selection may be conducted according to one of four schemes [142]:

- **Filter feature selection**
  Filter feature selection methods work directly on the available data and select features based on the data properties. They are independent of any learning methods [143, 144].
- **Wrapper feature selection**
  Unlike filter methods, in wrapper methods, learning methods may evaluate features but without consideration of the structure of the classifier [144].
- **Embedded feature selection**
  In embedded methods, the learning and feature selection aspects are related and work together.
- **Hybrid systems**
  These systems may be a combination of wrapper and filter approaches [145].

Feature selection may also be categorized into three main branches: supervised, semi-supervised, and unsupervised, as explained in the following subsections.

2.5.1 Supervised Feature Selection

In supervised feature selection, the selection of a set of features from a larger number of features is based on one of three characteristics [142]:

- Features of a size that optimizes an evaluation measure.
- Features that meet a specific condition in the evaluation measure.
- Features that best match a size and evaluation measure.

Supervised feature selection methods deal primarily with the classification problem, in which the class labels are known in advance [146]. Numerous studies have investigated supervised feature selection using the measures of the information theoretic [147], Hilbert-Schmidt independence criterion [148]. Unfortunately, no class labels are available in medical image segmentation, so, the supervised feature selection was unsuitable for the research presented in this thesis.

2.5.2 Semi-Supervised Feature Selection

The concept of semi-supervised feature selection has emerged recently as a means of addressing situations in which insufficient labels are available to cover the entire training data [149] or in which a substantial portion of the data are unlabeled. Traditional supervised feature selection techniques are ineffective under such circumstance. The concept of semi-supervised feature selection is therefore employed for the selection of features when not enough labels are available.

A semi-supervised feature selection constraint score that takes into account the unlabeled data is proposed in [150]. The literature also contains proposals for numerous semi-supervised techniques based on spectral analysis [149], a Bayesian network [151], a combination of a traditional technique with feature importance measure [152], or the use of a Laplacian score [153]. Although semi-supervised selection does not require a complete set of class labels, it does need

some. Because no labels are available in the case of this research, semi-supervised feature selection was not an option.

2.5.3 Unsupervised Feature Selection

Unsupervised feature selection is the process of selecting the most relevant non-redundant features from a larger number of features without the use of class labels. Mitra et al. [154] proposed an unsupervised feature selection algorithm based on feature similarity. They used a maximum information compression index to measure the similarities between features so that similar features could be discarded. He et al. [155] proposed an unsupervised feature selection technique that relies on the Laplacian score to indicate the significance of the features. Zhao et al. [156] used spectral graph theory to develop a new algorithm that unifies both supervised and unsupervised feature selection in one algorithm. They applied the spectrum of the graph that contains the information about the structure of the graph in order to measure the relevance of the features. Cai et al. [157] proposed a new unsupervised feature selection algorithm called Multi-Cluster Feature Selection, in which the features selected are those that maintain the multi-cluster structure of the data. Farahat et al. [158] present a novel filter unsupervised greedy feature selection algorithm. Their method consists of two parts: a recursive technique for calculating the reconstruction error of the matrix of features selected along with a greedy algorithm for feature selection. The method was tested on six different benchmark data sets, and the results show an improvement over state-of-the-art unsupervised feature selection techniques.

For the research presented in this thesis, because of the unavailability of class labels, the previous five popular unsupervised feature selection algorithms were applied for the selection of suitable features for training the evolving fuzzy system. These five methods, along with an additional correlation-based method, were combined to produce an ensemble of final relevant features that could be used for training. In the remaining of the thesis, the output matrices of these techniques are denoted as follows:

- Mitra et al. [154]- F_F (feature similarity).
- He et al. [155]- F_L (Laplacian score).
- Zhao et al. [156]- F_P (spectral graph).

- Cai et al. [157]- F_M (multi-cluster).
- Farahat et al. [158]- F_G (greedy algorithm).
- F_C (correlation method).

Chapter 3

Evolving Rules for Image Segmentation

3.1 Introduction

The potential of evolving fuzzy systems for user-oriented image segmentation in general and for medical imaging in particular seems not to have been intensively explored. The general approach proposed in this thesis is based on the evolving of fuzzy rules as a continual function of user feedback, which entails the use of manually segmented images to serve as "gold standard images" or "ground-truth images". These function as the baseline for benchmarking the segmentation algorithm: the more the algorithmically generated segments overlap with the gold standard images, the higher the accuracy of the algorithm. In this work, "evolving" is understood to be an iterative regeneration of rules as more and more images are processed, for example, in applications such as medical image analysis. Compared to other methods, segmentation using fuzzy rules offers the benefits of transparency and interpretability [13, 14, 15]. Fuzzy inference systems generally consist of a set of if-then rules of the following form:
IF x_1 is A_1 AND x_2 is A_2 AND $\cdots$ AND x_n is A_n,
THEN y is B,
where $x_i, y \in X$ are variables defined in corresponding universes of discourse X_i and Y, respectively, and A_i and B are fuzzy subsets. For a Takagi-Sugeno fuzzy inference system [133], a fuzzy rule is of the following general form:

IF x_1 is A_1 AND x_2 is A_2 AND $\cdots$ AND x_n is A_n,
THEN $y = f_j(x_1, x_2, \ldots, x_n), j = 1, 2, \cdots, N$,
where x_i and y are variables defined in corresponding universes of discourse X_i and Y, respectively, and A_i is a fuzzy subset. The function $f_j(x_1, x_2, \ldots, x_n)$ is a crisp (non-fuzzy) function of x_i. In general, the function f_j is defined as the weighted combination of all variables:

$$f_j(x_1, x_2, \ldots, x_n) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n. \tag{3.1}$$

The output is then calculated by

$$y = \frac{\sum_{j=1}^{N} f_j(x_1, x_2, \ldots, x_n) T_{i=1}^{m_j} \mu_j(x_i)}{\sum_{j=1}^{N} T_{i=1}^{m_j} \mu_j(x_i)}, \tag{3.2}$$

where N is the number of fuzzy rules, n is the number of inputs (features), $\mu_j(x_i)$ is the membership value of the i-th input x_i for the j-th rule, $1 \leq m_j \leq n$, and T is a T-norm representing the logical conjunction.

Assuming that a clustering or optimization approach is used to generate the rules from a dataset (features plus the desired outputs), the general idea of an evolving fuzzy technique for image segmentation can be defined as follows:

Definition [159]– The evolving fuzzy technique for image segmentation is a fuzzy rule-based approach to image segmentation that starts with an initial rule base, developed from the classification of existing input/output data, and that continually adds input data from user feedback in order to reclassify the data and to regenerate the rule base using both the initial and the new data. The input to the system consists of image features that characterize the image content. The output of system consists of either parameters for directly segmenting the image (e.g., a threshold for binarization) or adjusted parameters for a parent algorithm (e.g., a similarity threshold for region growing). The output can also be a set of weights for quantifying the quality of multiple segments and thereby determining whether to switch between different algorithms or to fuse segments generated by different algorithms.

Definition of Rule evolving [159] – The evolving fuzzy technique for image segmentation is based on rule evolving in which the rule base adapts to incoming data so as to reclassify and capture the overall dataset with greater efficiency and accuracy.

Rule evolving depends, of course, on the availability of new data that is structured by the application based on user feedback. "User" here indicates an experienced user: an "expert". For the purposes of this research, segmented images (gold standards) were prepared manually in advance by an expert and introduced incrementally in order to simulate the process of user feedback. This method neglects the effect of intra-observer variability, (which is beyond the scope of this thesis). Relying on proper (correct) feedback is generally safe if substantial feedback from the same user is available, since the likelihood of mistakes need not affect the performance of the learning system. Alternatively, feedback from multiple users could be applied.

Three proposed approaches are presented in this chapter: evolving fuzzy image segmentation (EFIS), self-configuring EFIS (SC-EFIS), and multi-parametric SC-EFIS (MSC-EFIS).

3.2 EFIS - Evolving Fuzzy Image Segmentation

In this section, the proposed evolving fuzzy image segmentation (EFIS) framework is presented. For any segmentation technique, EFIS can be trained and used as a means of adjusting the parameters of that technique in order to increase its segmentation accuracy. EFIS is not a single algorithm but rather a set of algorithms. An EFIS approach typically entails three stages: an offline phase (Algorithm 1), a training phase (Algorithm 1), and an online and evolving phase (Algorithm 2).

3.2.1 Offline Phase

In the offline phase, the available images are scanned so that the best parameters for each image can be calculated, and so that a region of interest (ROI) around each image segment can be detected. The process is as follows:

- Select one or more parent algorithms to perform segmentation (in case the objective is to

use EFIS to adjust the parameters of existing methods) and to determine the parameters to be adjusted (Algorithm 1, DPA).

- Collect images and corresponding gold standard images (segmented by an expert user) (Algorithm 1, RTI, and RGI).
- Determine a procedure for finding the optimal parameters for training (e.g., via brute force search prior to training) (Algorithm 1, BSP).
- Use domain knowledge to customize a detection algorithm for finding the seed points that are assumed to be the centre of the ROI in order to calculate the features (Algorithm 1, DRI, and SRI).

ROI Detection

The detection algorithm should find the ROI, which is defined as a rectangle that contains the given segment. In this research, a simple approach was developed that starts with the detection of the position of a point $(x^c_{\text{ROI}}, y^c_{\text{ROI}})$ assumed to be the centre of the ROI (i.e., region to be segmented). This point is detected by tracing an $n \times n$ mask (e.g., 10×10 pixels) over the image in order to calculate the sum and the standard deviation of the intensities within the mask as well as the correlation between the mask and its neighbour. Based on empirical knowledge about the test images (section 4.1), the ROI will be dark (most breast lesions in ultrasound images are hypo-echoic, meaning that they are darker than surrounding tissue). The segments also exhibit relatively low standard deviations. The mask with the minimum sum and standard deviation is designated to contain the seed point. If the minimum sum and standard deviation are found in more than one mask, the correlation coefficients between those masks and the masks preceding them are used as a basis for the selection of a single mask. For every $n \times n$ window, the minimum sum, standard deviation, and in some cases, correlation, are thus considered in the designation of the position of the ROI centre $(x^c_{\text{ROI}}, y^c_{\text{ROI}})$. The algorithm begins by defining an ROI around this point and then enlarges it in small steps. This process stops when the standard deviation of one region becomes greater than or less than the standard deviation of the previous region by a specified percentage (e.g., 2%). The window at its next-to-last dimension is then considered to be the ROI. Of course, ROIs are different sizes for different images. Fig. 3.1 shows the detected ROI R_O (large rectangle).

Algorithm 1 Offline and Training Phases Algorithm

1: **———— Offline phase ————**
2: DPA – Determine the parent algorithm(s) (e.g., global thresholding) and its parameters $p_1, p_2, \cdots, p_k$.
3: RTI – Read the available images $I_1, I_2, \cdots, I_{N_I}$.
4: RGI – Read the gold standard images $G_1, G_2, \cdots, G_{N_I}$.
5: BSP – Via exhaustive or trial-and-error comparisons with gold standard images, determine the best segments $S_1, S_2, \cdots, S_{N_I}$ and the best parameters $p_1^*, p_2^*, \cdots, p_k^*$ that generate the best segments and store them in matrix T.
6: DRI – Determine ROIs R_O around each segment.
7: SRI – Save ROIs for each image.
8: **———— Training phase ————**
9: DTI Determine the available training images $I_1, I_2, \cdots, I_r$ with ROIs $R_{O_1}, R_{O_2}, \cdots, R_{O_{N_R}}$.
10: INI – Create two empty matrices: M for input and O for output.
11: **for** each ROI **do**
12: SNS – Determine the number of SIFT points N_F inside R_{O_i} (e.g. $N_F = 10$).
13: **for** each SIFT point **do**
14: EXF – Extract features $f_1, f_2, \cdots, f_{N_T}$ from the SIFT point's neighbourhood and store them in F_T.
15: AFM – Append matrix F_T to M.
16: AOM – Append the optimum parameters that belong to R_{O_i} to O.
17: **end for**
18: GFR – Generate fuzzy rules $R_{F_1}, R_{F_2}, \cdots$ from the input matrix M and the output matrix O (e.g., using clustering).
19: **end for**

3.2.2 Training Phase

In this phase, the set of training images is selected, and the features are extracted from their detected ROIs and used to train the fuzzy system as follows:

- Determine the training images (Algorithm 1, DTI).
- Determine the number and the locations of scale invariant feature transform (SIFT) points inside the detected ROI (Algorithm 1, SNS). SIFT is used to locate 10 points within the

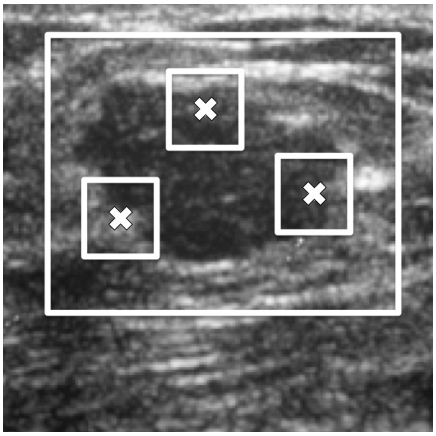

Figure 3.1: ROI detected (large rectangle), with three SIFT points indicated crosses within the smaller rectangle for feature calculation.

ROI (Fig. 3.2). Fig. 3.1 depicts multiple seed points (white crosses). Fig. 3.2 shows a sample case for seed point detection inside an ROI. This process is as follows:

1. Pass the image to the SIFT, which detects a set of N_A points inside R_O, along with their SIFT descriptors D_S.
2. Sort the N_A points in descending order according to the mean value of their SIFT descriptors D_S.
3. Return the first N_F points (e.g., $N_F = 10$) that are separated from each other by 20 pixels in each direction.

- Establish a procedure for extracting features from the ROIs detected (Algorithm 1, EXF).

Feature Extraction

To characterize the image, multiple features inside the detected ROI are calculated (Fig. 3.1). For each seed point delivered by the SIFT, a neighbourhood is constructed around the seed point, e.g., 40×40 pixels (small rectangles centered around the crosses in Fig. 3.1) and a set of features are extracted in those neighbourhoods.

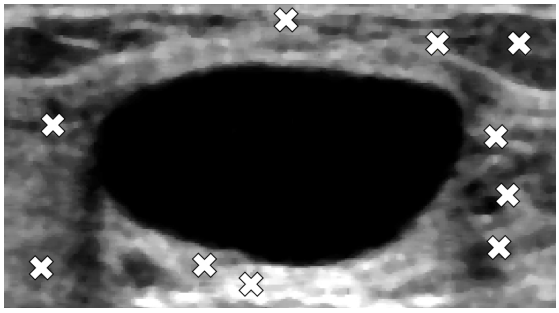

Figure 3.2: Example of seed point detection inside an ROI: 10 SIFT points are detected inside the cropped ROI.

EFIS hence extracts N_F (e.g., $N_F = 10$) different data points from the same image. Each data point is represented as a single row consisting of the following features:

- The mean graylevel (an indication of darkness)
- The standard deviation (intensity variation)
- A set of texture features from the gray level co-occurrence matrix (GLCM) in the directions of $0°, 45°, 90°$, and $135°$.

The texture features are as follows:

- Contrast (CON): a measure of the difference in intensity between a pixel and that of its neighbours.
- Correlation (COR): a measure of how correlated a pixel is to its neighbours.
- Energy (EN): the sum of squared elements in the GLCM.
- Homogeneity (HOM): a value that measures the closeness of the distribution of the GLCM elements to the GLCM diagonal.

- In conjunction with the optimum parameters, the extracted features are used in order to generate the initial fuzzy rules (Algorithm 1, GFR). For the generation and implementation of the fuzzy inference system, the Matlab function "genfis3" was used with a Sugeno inference mechanism [160] and FCM for clustering. Gaussian membership functions were used for input, and linear membership functions for the output.

A general example of the rules generated is as follows:
IF (μ IS μ_1) AND (σ IS σ_1) AND ($CON0$ IS low) AND ($COR0$ IS high) AND ($EN0$ IS medium) AND ($HOM0$ IS high) AND ($CON45$ IS low) AND ($COR45$ IS low) AND ($EN45$ IS medium) AND ($HOM45$ IS medium) AND ($CON90$ IS high) AND ($COR90$ IS high) AND ($EN90$ IS low) AND ($HOM90$ IS low) AND ($CON135$ IS high) AND ($COR135$ IS medium) AND ($EN135$ IS medium) AND ($HOM135$ IS high) THEN (T^* IS T_1), where $CON0$ is the contrast at 0°, etc.

3.2.3 Online and Evolving Phase

In this phase, the testing and the evolving process are performed for each testing image as follows (Fig. 3.3):

- Fill a matrix F_S with extracted features (Algorithm 2, EXF).
- Perform fuzzy inference using F_S, and a parameter vector T_O (size 1×10, parameter for each row in F_S) is returned and the final output parameter T^* is calculated (Algorithm 2, PFI and GSO).
- Apply T^* to segment the image (Algorithm 2, APS).
- Store the resulting segment and then display it to the user for review and eventual correction (Algorithm 2, FED).
- Calculate the best parameter for the current image using the user-corrected segment and store it in T_B (Algorithm 2, DPA).
- Following pruning, append a revised version of features F_S, thresholds T_O to M and O (Algorithm 2, PRU and AIS).

Pruning

Compared to streamed data, the number of images a clinical expert can process in daily practice is quite small. It may take the expert several weeks or months just to process

several thousand images that lead to the addition of several thousand rows to the M matrix and the generation of a significant number of rules. It seems that "memory" is not a critical aspect; nonetheless, a "pre-pruning" stage for EFIS was implemented and tested. Once a new image is processed, a new row is added to the M matrix only if the features and corresponding output parameters have not been previously observed. The difference between an input row (features plus output) and all rows in M, result in the addition of the information from the new image only if it is not identical or very similar. The Euclidean distance is used as a parameter for the application of the pruning process.

- Regenerate the current fuzzy system using the updated M, and O matrices (Algorithm 2, GFR).

- Repeat the process as long as new images are available.

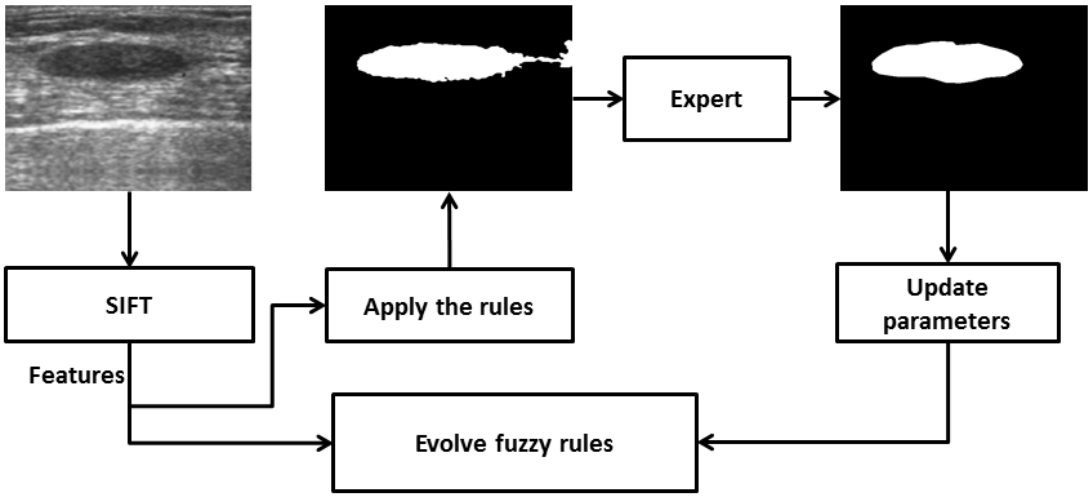

Figure 3.3: The online and evolving phase emphasizing the expert feedback loop.

EFIS must be designed and trained for specific algorithms and image categories. For the experiment presented in this thesis, the following component was designated:

- Parent algorithms: global thresholding (with a static/fixed rule base), region growing, and statistical region merging

Algorithm 2 Online and Evolving Phase Algorithm

———— Online phase ————
LFR – Load the fuzzy rules $R_{F_1}, R_{F_2}, \cdots$ and the matrices M and O.
DTI – Determine the available testing images $I_1, I_2, \cdots, I_{N_E}$ with ROIs $R_{O_1}, R_{O_2}, \cdots, R_{O_{N_E}}$.
for each ROI **do**
  SNS – Determine the number of SIFT points N_F inside R_{O_i} (e.g. $N_F = 10$).
  for each SIFT point **do**
    EXF – Extract features $f_1, f_2, \cdots, f_m$ from the SIFT point's neighbourhood and store them in F_S.
  end for
  F_S of $N_F \times N_T$.
  PFI – Perform fuzzy inference to generate output:
  T_O = FUZZY-INFERENCE($R_{F_1}, R_{F_2}, \cdots, F_S$).
  T_O of $1 \times N_F$.
  GSO – Generate a single output T^* from T_O using the mean of T_O (μ_{T_O}), the median of T_O (M_{T_O}), the fuzzy membership (m_{T_O}) of the standard deviation of T_O (σ_{T_O}) using a Z-shaped function (zmf)
  $m_{T_O} = zmf(\sigma_{T_O}, [10, 30])$, and
  $T^* = m_{T_O} * \mu_{T_O} + (1 - m_{T_O}) * M_{T_O}$
  APS – Apply the parameters to segment I_i
  FED – Display the segment S and wait for the user feedback (user generates a gold standard image G by editing S)
  ————Evolving phase——
  DPA – Determine the best output(s) $p_1^*, p_2^*, \cdots, p_k^*$ (via comparison of S with G) and store it in T_B.
  PRU – Pruning step: Discard row from F_S, T_B that are similar to rows in M and O, respectively.
  AIS – Append the updated matrices F_S and T_B to M and O respectively.
  GFR – Generate fuzzy rules $R_{F_1}, R_{F_2}, \cdots$ from the updated matrices M and O (e.g., using clustering).
end for

- Parameters: thresholds (section 4.3.1)
- Images and corresponding gold standard images: breast ultrasound images collected online
- Procedure for finding optimal parameters: brute force (or trial-and-error) via comparison

with gold standard images

3.2.4 EFIS for Single Parameter

EFIS can be used for the learning of a single output parameter, which can be used to perform segmentation directly (e.g., for image thresholding) or as an adjustable parameter of a parent algorithm. If the parent algorithm has only one parameter, the operating rules of the following general form:
IF x_1 is A_1 AND x_2 is A_2 AND $\cdots$ AND x_n is A_n,
THEN y is B.
With n features to adjust a threshold τ,
IF f_1 is F_1 AND f_2 is F_2 AND $\cdots$ AND f_n is F_n
THEN τ is T.
Subsection 4.3.1 explains the use of global thresholding (calculating a threshold based on the image histogram), region growing (RG), and statistical region merging as examples of parent algorithms in which a single parameter is adjusted.

3.2.5 EFIS for Fusion and Switching

This subsection explains the use of EFIS for enabling the learning of the weights for three methods. Because weight quantifies how accurately individual methods perform for given images, an evolving switch can be designed to select the method with the highest weight. The weights for the same image for all methods are determined at the same time and under the same conditions, i.e., using the same preprocessing. A fusion algorithm, called STAPLE [161] is employed to incorporate the EFIS output as fusion weights. The output parameters can therefore serve as the mechanism for a given algorithm with respect to the selection of the parent algorithms to be used, or to fuse the results from different parent algorithms. The EFIS approach builds two new configurations or versions:

- Switching EFIS (EFIS-S) results in switches among three different thresholding methods based on a fixed (or unadjusted) parameter. The three methods selected were the Niblack

method [29], statistical region merging (SRM) [47], and region growing RG. The results of the method with the highest weight is selected as the output.

- Fusion EFIS (EFIS-F) fuses the results of the three different segmentation methods using the weights generated and the STAPLE algorithm [161]. The weights are first mapped into a specific number of copies (one copy per each 20% of the weight (Algorithm 3)) to be fused by the STAPLE algorithm, after which, weights below 50% are ignored. The STAPLE algorithm, which is an optimization method, can extract a contour from multiple binary segments in order to establish a "consensus segment."

Algorithm 3 Conversion Weights into Number of Copies for Each Segment to be Fused by STAPLE

1: Get the weights: W_1, W_2, W_3 // e.g., $(0.90, 0.80, 0.60)$
2: Normalize: $W_i = \frac{W_i}{W_{min}} - 1$ // e.g., (0.5, 0.3, 0)
3: Copies$(W_i) = round(\frac{W_i}{0.2}) + 1$ // e.g., (4, 3, 1)

Assuming N_T features, weights w_i can be generated so that they quantify the level of confidence in the results of a given method for a given image, as follows:
IF f_1 is F_1 AND f_2 is F_2 AND $\cdots$ AND f_n is F_n,
THEN w_1 is W_1 AND w_2 is W_2 AND w_3 is W_3,
or
IF f_1 is F_1 AND f_2 is F_2 AND $\cdots$ AND f_n is F_n,
THEN $\mathbf{W} = [W_1 \;\; W_2 \;\; W_3]$.

The Jaccard index J (equation 4.1, section 4.2) for each image (during training) can be used as the weight for that method. For new images, EFIS-F infers a weight for each method. More images were used to train EFIS-S and EFIS-F than to train EFIS because extra parameters must be learned.

3.3 Disadvantages of EFIS

EFIS has some disadvantages that prevent its use as a general approach. The features are calculated from a rectangles around a set of points assigned by the SIFT inside an ROI. This process

requires an ROI to be detected for each image, and EFIS uses a customized algorithm for detecting the ROI, which is not appropriate for all images. The number of SIFT points detected and used inside the ROI is fixed and may or may not be sufficient to have an efficient feature extraction process. The area around each SIFT point is also a fixed size, which may not be efficient for feature extraction if other types of images are processed. EFIS also extracts a fixed number of features from each rectangle around each SIFT point, which may not be effective for different categories of images. No pruning process is applied in the EFIS training phase. This process occurs only in its online phase and only with fixed pruning parameters that remain unchanged regardless of the current segmentation technique. In calculating the final threshold from a threshold matrix, EFIS uses a fusion process but with a fixed parameter range that may work well for one segmentation technique but be useless for others.

3.4 Self-Configuring EFIS (SC-EFIS)

This section introduces a new version of EFIS, called self configuring evolving fuzzy image segmentation (SC-EFIS) representing a higher level of automation than that offered by EFIS. SC-EFIS was developed in an attempt to provide solutions for the problems associated with EFIS, as explained in section 3.3. A detailed explanation of the proposed SC-EFIS algorithm is presented in this section. It consists of three phases: the preprocessing phase, the training phase, and the online and evolving phase.

3.4.1 Preprocessing Phase

In the preprocessing phase, all available images are processed in order to determine two crucial elements: the size of the area around the SIFT points, and the final number of features to be used for the current type of image.

The $Z \times Z$ rectangle around each SIFT point (Fig. 3.4) to be used for feature calculation is determined based on the sizes of all available images (algorithm 4, RSI and DSI). Following this step, the set of features that should be used for the available images is selected from a large number of features which are calculated for each image from a rectangle around the SIFT points

located in the entire image (since there is no longer an ROI) (Fig. 3.5). This process starts with a determination of the number of SIFT points N_F that should be used in the current image (algorithm 4, DSP). This step is identical to the procedure used in the EFIS training phase, as previously explained in subsection 3.2.2, with three exceptions: the SIFT points are detected inside the entire image, the final number of SIFT points N_F is not fixed, and the final N_F points returned are separated from each other by Z in each direction.

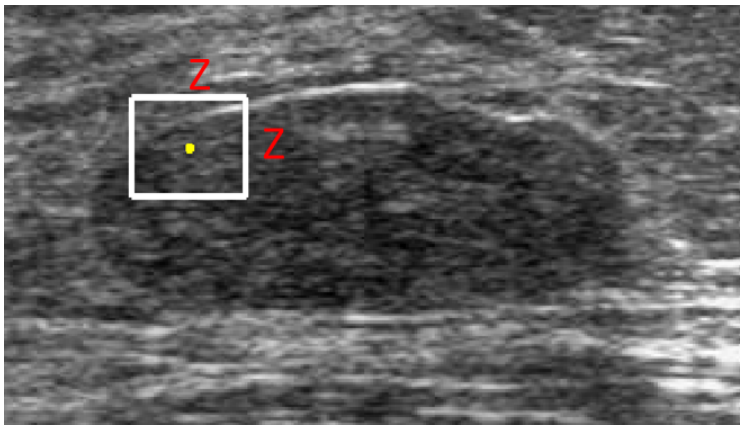

Figure 3.4: Size of the rectangle around a SIFT point

For all N_F points, features are extracted from a rectangle R_C around each point, based on the discrete cosine transform (D_C) of R_C, the gradient magnitude (G_M) of R_C, the approximation coefficient matrix A_C of R_C (computed using the wavelet decomposition of R_C), and the SIFT descriptors D_S. The following set of features is extracted (Algorithm 4, EXF and EIF):

1. The mean, median, standard deviation, co-variance, mode, range, minimum, and maximum of R_C, $D_{C_{R_C}}$, and $A_{C_{R_C}}$, and $G_{M_{R_C}}$ (32 features)

2. The mean, median, standard deviation, co-variance, range, minimum, maximum, and zero population of D_S (eight features) with the minimum of D_S changed to be the minimum number after zero

3. The contrast, correlation, energy, and homogeneity of the gray level co-occurrence matrices (computed in four directions $0\,^\circ$, $45\,^\circ$, $90\,^\circ$, and $135\,^\circ$) of R_C, $D_{C_{R_C}}$, and $A_{C_{R_C}}$, and $G_{M_{R_C}}$ (64 features)

4. The contrast, correlation, energy, and homogeneity of the gray level co-occurrence matrices (computed in only one directions of $0\,^{\circ}$) of D_S (four features)

5. A feature matrix F_1 of size $N_F \times N_T$ generated for I (in this case $N_T = 108$)

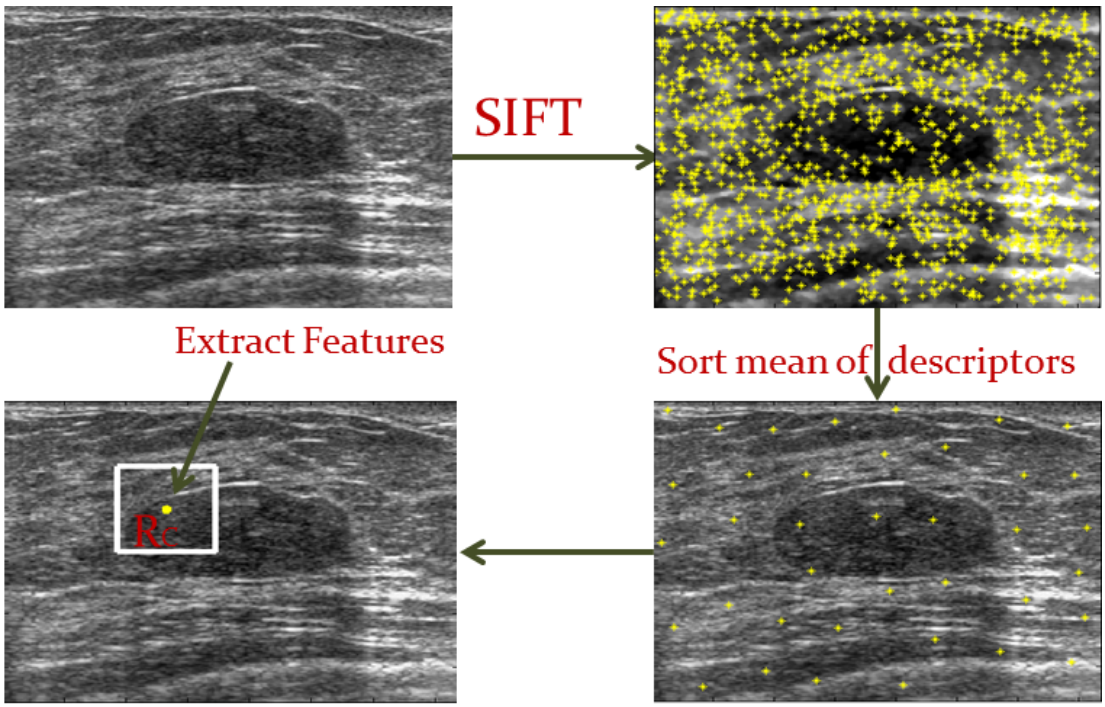

Figure 3.5: Feature extraction process.

The next step is to calculate eight different statistical measures (mean, median, mode, standard deviation, co-variance, range, minimum, and maximum) from F_1. The resulting matrix F_2 (size $8 \times N_T$) is returned, in which each row represents a statistical measure (Algorithm 4, CSF). F_2 is then appended to the feature matrix F_3 (Algorithm 4, AF I). After all images are processed, the feature matrix F_3 is formed from the features of all images, with each image being represented by eight rows.

In the last step, the final set of features that should be used in the current type of image are selected from F_3 (Fig. 3.6). This process starts with the removal of very similar features in F_3 based on the calculation of the correlations between all features in F_3. Hence, if two features

are correlated with a correlation coefficient of at least 99%, then one is kept and the other is discarded. The output of this process is a matrix F_4 (Algorithm 4, RSF).

Algorithm 4 Preprocessing Phase Algorithm

RTI – Read the available images $I_1, I_2, \cdots, I_{N_I}$.
RSI – Read the size of the images $R_{W_1}, R_{W_2}, \cdots, R_{W_{N_W}}, C_{L_1}, C_{L_2}, \cdots, C_{L_{N_C}}$.
DSI – Determine the size of the rectangle
Z = $max(median(R_{W_1}, R_{W_2}, \cdots, R_{W_{N_W}}), median(C_{L_1}, C_{L_2}, \cdots, C_{L_{N_C}})) * 10\%$.
CFI – Create an empty initial matrix F_1 and an empty final matrix F^*.
for each image **do**
  DSP – Determine the number of SIFT points N_F that should be used in image I_i.
  for each SIFT point **do**
    EXF – Extract features $f_1, f_2, \cdots, f_{N_T}$ (e.g., $N_T = 108$) from the $Z \times Z$ rectangle around each SIFT point (Fig. 3.4).
    AIF – Append the row to the initial matrix F_1, which becomes of size $N_F \times N_T$.
  end for
  CSF – Calculate S_T (e.g., $S_T = 8$) different statistics from F_1 and assigned in F_2.
  AFI – Append F_2 of the current image of size $S_T \times N_T$ to the feature matrix F_3.
  FMS – The feature matrix F_3 becomes of size $L \times N_T$, $L = S_T * N_I$.
end for
RSF – Remove very similar features from F_3 (e.g., at least 99% correlated). F_4 is a reduced matrix of F_3 of size $L \times N_{T_1}$, $N_{T_1} \leq N_T$.
DNF – Determine the number of features by discarding similar ones from F_4 (e.g., at least 90% corelated). F_C is a feature matrix generated from F_4 of size $L \times N_{T_2}$, $N_{T_2} \leq N_{T_1}$.
UFS – Use k (e.g., $k = 5$) different unsupervised feature selection methods to generate k different feature matrices in addition to F_C: F_P, F_M, F_F, F_G, and F_L. All of these matrices are of size $L \times N_{T_2}$.
SFF – Select any features found in at least half of the matrices to form F_5 of size $L \times N_{T_3}$, $N_{T_3} \leq N_{T_2}$.
FFM – Generate a final feature matrix F^* from F_5 by removing similar features (e.g., at least 90% correlated). F^* is of size $L \times N_L$, $N_L \leq N_{T_3}$.

For any unsupervised feature selection technique, the number of features N_{T_2} that should be returned must be established in advance. A correlation with a threshold of 90% is used in order to determine the number of features that should be returned from F_4 (Algorithm 4, DNF). Following this process, F_C is the resulting feature matrix. In addition to F_C, five different unsupervised

feature selection methods are also used for feature selection. The matrix F_4 and the variable N_{T_2} are passed to the methods, and each method returns a different matrix with its selected features. The resulting matrices are F_G [158], F_L [155], F_F [154], F_P [156], and F_M [157] (Algorithm 4, UFS). For all features in the six matrices, any found in at least three of the six methods are selected and appended to a matrix F_5 (Algorithm 4, SEF). The final matrix F^* is generated based on the discarding of features from F_5 that are at least 90% correlated (Algorithm 4, FFM).

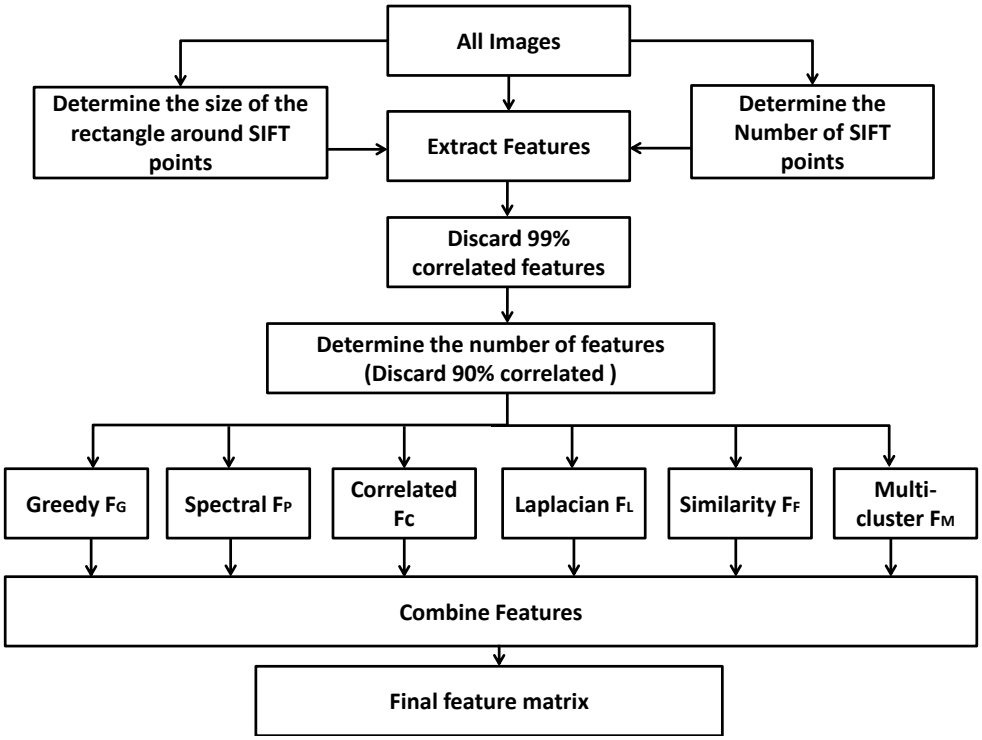

Figure 3.6: SC-EFIS preprocessing phase.

3.4.2 Offline Phase

In the offline phase, the best parameters for segmenting each image are calculated through an exhaustive search and then stored in matrix T (Algorithm 5, BSP). The process is performed as explained in subsection 3.2.1.

3.4.3 Training Phase

In this phase, the features selected for the training images are used for the training of the fuzzy system. A set of images are randomly selected for training (Algorithm 5, DTI). A matrix M is created and filled with the rows from F^* that belong to the training images (Algorithm 5, FRM and AIT). A matrix O is created and filled with the rows from T that belong to the training images (Algorithm 5, FRO and AIT). A pruning step is performed starting from the second training image in order to ensure that M and O do not contain similar rows (Algorithm 5, PRU). The pruned matrices M and O are used for the generation of the initial fuzzy rules (Algorithm 5, GFR). The initial fuzzy system is built through the creation of a set of rules using the Takagi-Sugeno approach to describe the input and output matrices. Based on N_L different features from the input and one optimal parameters from the output, a set of rules is generated whereby the features are in the antecedent part and the optimal parameters are in the consequent part of the rules.

3.4.4 Online and Evolving Phase

In the online and evolving phase, the testing process is performed, and the results are saved. The evolving process is also performed in order to increase the capabilities of the proposed system. For each test image, a matrix F_S is filled with the rows from F^* that belongs to the test image (Algorithm 6, FSM). A fuzzy inference using F_S is applied, and a parameter vector T_O is returned (size 1×8) and the final output parameter T^* is calculated (Algorithm 6, PFI and GSO). The resulting parameter is used for the segmentation of the image (Algorithm 6, APS), and the resulting segment is stored and then displayed to the user for review and eventual correction (Algorithm 6, FED). The best parameter for the current image is then calculated based on the user-corrected segment and is stored in T_B (Algorithm 6, DPA). A pruning of F_S and T_B is performed as described in subsection 3.2.3, with the exception that the Euclidean distance thresholds are different for different techniques. After pruning, a revised versions of F_S and T_B are appended to M and O (Algorithm 6, PRU and AIS). In the final step, the current fuzzy system is regenerated using the updated M and O matrices (Algorithm 6, GFR), and the process is repeated as long as new images are available.

Algorithm 5 Offline and Training Phases Algorithm

———— Offline phase ————
DPA – Determine the parent algorithm(s) (e.g., global thresholding) and their parameters $p_1, p_2, \cdots, p_k$.
RGI – Read the gold standard images $G_1, G_2, \cdots, G_n$.
BSP – Via exhaustive or trial-and-error comparisons with gold standard images, determine the best segments $S_1, S_2, \cdots, S_n$ and the best parameters $p_1^*, p_2^*, \cdots, p_k^*$ that generate the best segments and store them in matrix T.
———— Training phase ————
DTI – Determine the available training images $I_1, I_2, \cdots, I_{N_R}$.
INI – Create two empty matrices M for input and O for output.
for i=1:N_R **do**
    FRM – Fill matrix F_T with rows from matrix F^* that belong to the training image I_i ($F_T = F^*(I_i)$).
    FRO – Fill matrix T_R with rows from matrix T that belong to the training image I_i ($T_R = T(I_i)$).
    if i=1 **then**
        AIT – Append F_R to M, and T_R to O.
    else
        PRU – Pruning step: Discard rows from F_R and T_R that are similar to rows in M and O, respectively.
        AIT – Append the updated matrices F_R and T_R to M and O respectively.
    end if
end for
GFR – Generate fuzzy rules $R_{F_1}, R_{F_2}, \cdots$ from the input matrix M and the output matrix O (e.g., using clustering).

3.5 Multi-Parametric SC-EFIS

SC-EFIS can also applied for multiple output parameters, which can be used to perform segmentation directly (e.g., multi-thresholding), or as the adjustable parameters of a parent algorithm. In such cases, the system is called multi-parametric SC-EFIS (MSC-EFIS) with operating rules such as the following:

IF x_1 is A_1 AND x_2 is A_2 AND $\cdots$ AND x_n is A_n,

THEN y_1 is B_1 AND y_2 is B_2 AND $\cdots$.

With n features to adjust the parameters $p_1, p_2, \cdots$,

Algorithm 6 Online and Evolving Phase Algorithm

1: LFR – Load the fuzzy rules $R_{F_1}, R_{F_2}, \cdots$ and the matrices M, O, and F^*.
2: DSI – Determine the available testing images $I_1, I_2, \cdots, I_{N_E}$.
3: **for** i=1:N_E **do**
4: FSM – Fill matrix F_S with the rows from matrix F^* that belong to the testing image I_i ($F_S = F^*(I_i)$).
5: PFI – Perform fuzzy inference to generate output: T_O = FUZZY-INFERENCE($R_{F_1}, R_{F_2}, \cdots$).
6: GSO – Generate a single output T^* from T_O using the mean of T_O (μ_{T_O}), the median of T_O (M_{T_O}), the fuzzy membership (m_{T_O}) of the standard deviation of T_O (σ_{T_O}) using a Z-shaped function (zmf) $m_{T_O} = zmf(\sigma_{T_O}, [(\mu_{T_O} * 0.10) \ \ (\mu_{T_O} * 0.20)])$, and $T^* = m_{T_O} * \mu_{T_O} + (1 - m_{T_O}) * M_{T_O}$.
7: APS – Apply the parameters to segment I_i.
8: FED – Display segment S and wait for user feedback (user generates a gold standard image G by editing S)
9: **——— *Rule Evolution - Invisible to User* ———**
10: DPA – Determine the best output $p_1^*, p_2^*, \cdots, p_k^*$ (via comparison of S with G) and store it in T_B.
11: PRU – Pruning step: Discard rows from F_S, T_B that are similar to rows in M and O, respectively.
12: AIS – Append the updated matrices F_S and T_B to M and O, respectively.
13: GFR – Generate fuzzy rules $R_{F_1}, R_{F_2}, \cdots$ from the updated matrices M and O (e.g., using clustering).
14: **end for**

IF f_1 is F_1 AND f_2 is F_2 AND $\cdots$ AND f_n is F_n,
THEN p_1 is P_1 AND p_2 is P_2 AND $\cdots$.

As an example, this section describes the use of SC-EFIS for adjusting more than one parameter in a normalized cut (N-cut) image segmentation technique [69]. An N-cut technique involves multiple parameters that should be tuned manually in order to achieve maximum segmentation accuracy. From those available, four different N-cut parameters are selected to be adjusted using the proposed algorithm:

- The number of segments (N_S): This parameter indicates the number of different regions into which the image will be divided. It has a default value of 5 and can range from 5 to 20.

- The radius (R_D): This parameter has a default value of 10 and a range from 8 to 20.
- The rate (R_A): This parameter has a default value of 0.3 and a range from 0.2 to 0.5.
- The edge variance (E_V): This parameter has a default value of 0.1 and a range from 0.1 to 0.5.

The last three parameters, R_D, R_A, and E_V, are used to calculate the similarity matrix. The specified ranges are determined experimentally based on the downloaded code [1]. This code is subject to a number of problems and the specified ranges are the ranges that do not cause problems in the code. Regarding the N_S parameter, three different default values are found in the code: 5, 8, and 10. However, the default number 5 is contained in the start demo file. For this reason, all operations performed in this research, such as the process for determining the optimum values explained in subsection 3.5.2, are based on the use of 5 as the default N_S value. However, in the result section, the proposed system is compared with three different results for the default values using $N_S = 5, N_S = 8$, and $N_S = 10$.

Different values of these parameters will produce different segmentation results, even for the same image. To achieve maximum segmentation accuracy, these parameters may therefore be fine-tuned manually for every set of images. To achieve maximum segmentation accuracy for each image, SC-EFIS is thus used to tune these parameters for each image based on its features. However, in some image data sets, keeping the default value of one parameter may be more efficient than training and adjusting that parameter. Different types of images may hence require the adjustment of four parameters while others may require three, two, or even only one parameter adjustment. The number of parameters that should be adjusted is determined based on the available data. The first step in MSC-EFIS is therefore to decide how many parameters should be adjusted for the available data set. SC-EFIS for the N-cut entails two main stages:

- Determining the number of parameters to be adjusted.
- Adjusting the selected parameters.

[1] http://timotheecour.com/software/ncut/ncut.html

3.5.1 Determining the Number of Parameters to Be Adjusted

In this stage, the number of parameters that should be adjusted for the available data is determined. First, SC-EFIS is used to train and adjust each parameter alone. Based on a comparison of the results with the default values, the number of parameters that should be adjusted from the four possible parameters is determined. The aim of this process is to establish, based on the data available, whether a parameter is worth adjusting or whether it is preferable to leave it at the default value.

SC-EFIS is applied to adjust each parameter in an independent system, and the final number of parameters that should be adjusted for the current data is determined based on the results. The results for each parameter are compared with the default results. The number of parameters that should be adjusted for the data available is determined based on this comparison. SC-EFIS will therefore have four versions: one for each parameter. For each parameter, the process consists of four phases: preprocessing, offline, training, and online and evolving. Each phase is the same as described for the overall SC-EFIS process (section 3.4).

3.5.2 Adjusting the Selected Parameters

In this stage, SC-EFIS is used to adjust the parameters selected in the previous stage. This stage is comprised of three phases:

- **Offline phase**
  In this phase, the optimum values for the combination of the selected parameters are calculated.
- **Training phase**
  In this phase, features are extracted, and an initial fuzzy system (rule base) is generated in order to train the set of parameters selected.
- **Online and evolving phase**
  In this phase, each incoming image is segmented using the parameters estimated by the fuzzy system. The proposed method is also used to update the N-cut parameters assigned by the system based on the segment provided by the expert. These updated parameters,

along with the features extracted from the test image, are used to evolve the existing fuzzy system. Each image encountered online is thus not only segmented by the system but also helps improve the system by facilitating the enhanced accuracy of the segmentation of future images (Fig. 3.7). Of course, the expert feedback, in the form of the provision of the ideal segmentation, is a crucial factor that can be assumed as a given in the medical field.

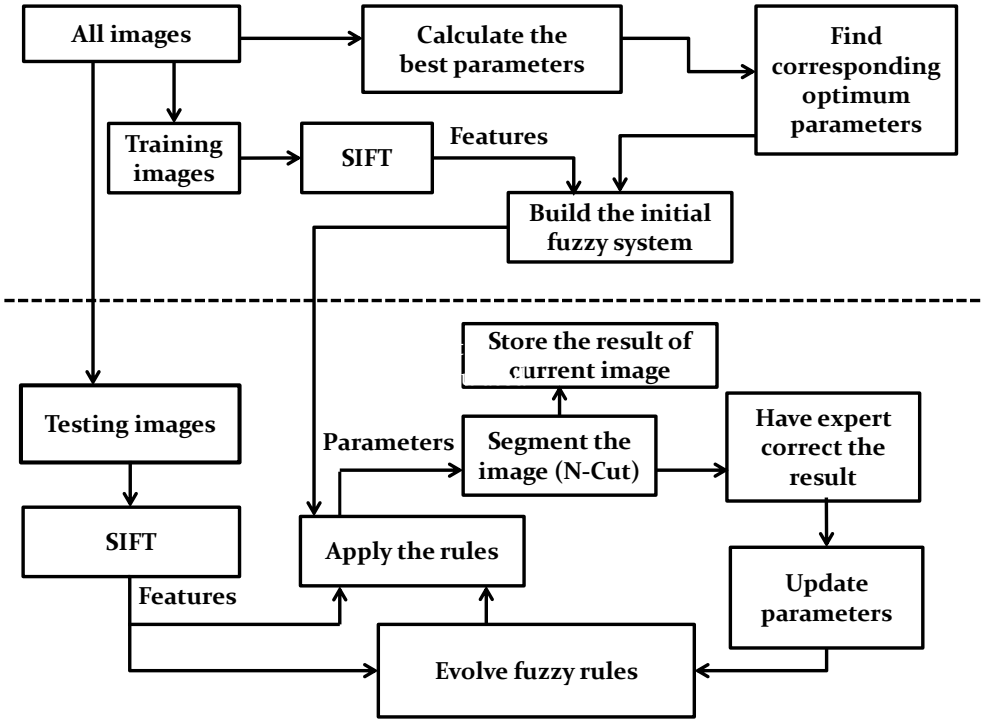

Figure 3.7: Proposed approach for the parameter adjustment of N-cut segmentation using evolving fuzzy rules.

The Offline Phase

After the optimum values are calculated for each parameter (subsection 3.5.1), the best values for the combination of selected parameters are calculated. An optimization technique is used for the calculation of the best N-cut parameters using the optimum values calculated for each parameter (subsection 3.5.1) as the initial input, as follows:

- Simulated annealing (SA) (with 10 iterations) [162, 163] is used to calculate the optimum parameters.
- N_{S_O}, R_{D_O} , and R_{A_O} are provided as initial input for the SA.
- The optimization function is represented by

$$\text{Minimize } f = 100 - \frac{|B \bigcap G|}{|B \bigcup G|} \tag{3.3}$$

  subject to:
  $5 \leq N_S \leq 20$,
  $0.2 \leq R_A \leq 0.5$, and
  $8 \leq R_D \leq 20$.

- The output of the SA are N_{S_F}, R_{A_F}, and R_{D_F} which are stored to be used in the training phase.

Training Phase

The training phase is identical to the SC-EFIS training phase for a single parameter (subsection 3.4.3), with the exception that the set of rules is generated based on three optimal parameters from the output rather than one.

Online and Evolving Phase

In the online and evolving phase, the remaining images are used for testing and evolving the system. For every incoming image, the parameters of the N-cut algorithm are estimated by the fuzzy system. The segmented image is then provided to an expert for correction. The parameters assigned by the system are updated using an optimization technique based on a comparison with the segment provided by the expert. The features extracted from the image, and the updated parameters are used for updating the existing system with the addition of the new information. The parameters of the next image are thus estimated using the updated system. Therefore, as long as new images are provided, and the results can be inspected by an expert, the scope and

vision of the system is continually increased through new information. This process enhances the ability of the system to estimate parameters for subsequent images. This phase is identical to the SC-EFIS online and evolving phase explained in subsection 3.4.4, with the following exceptions:

- The matrix T_E that results from the fuzzy inference is of size 8×3, with each column representing the results of one parameter. The trained fuzzy rules estimate a value for each parameter for each row in the feature matrix F_S. Because F_S consists of eight rows, the size of the returned matrix T_E is 8×3.

- The final output N_S^*, R_D^* and R_A^* are calculated for each parameter in the same way as SC-EFIS (Algorithm 6, GSO).

  Prior to their use in the N-cut algorithm, N_S^* and R_D^* are rounded whereas R_A^* is rounded and normalized to values less than one.

- If the value of any parameter is greater than or less than its range, it is assigned to the upper or the lower value of its range, respectively.

The final parameters calculated are used to segment the current image using the N-cut technique (Algorithm 7, APS). Most of the parameters generated by the initial fuzzy system may not the best ones, especially if a small number of images have been used to generate the first set of rules. For its accuracy to increase, the system therefore requires continual updating. This process starts with the resulted segment S being provided to an expert for correction (assuming that the corrected segment is the available gold standard image G previously segmented by an expert) (Algorithm 7, FED). The updating process is based on the segment G provided by the expert. In other words, the updating technique searches for values of the parameters N_S, R_D and R_A that generate a segment more closely resembling to G. The corrected segment is then used for updating the current parameters, as explained in subsection 3.5.2, with the following exceptions (Algorithm 7 OUP):

- The input to the SA techniques are the parameters N_S^*, R_D^* , and R_A^* (Algorithm 7, OUP)

- The output of the SA technique (N_{S_B}, R_{A_B}, and R_{D_B}) is appended to the T_B matrix and used for the evolving/retraining of the existing fuzzy system

Following the pruning process, the final input and output matrices are used for the evolution of the existing fuzzy rules (Algorithm 7, PRU, AIT, and RFR). The N-cut parameters of the next image are then estimated using the evolved fuzzy system.

Algorithm 7 Online and Evolving Phase Algorithm

1: LFR – Load the fuzzy rules $R_{F_1}, R_{F_2}, \cdots$ and the matrices M, O, and F^*.
2: DSI – Determine the available testing images $I_1, I_2, \cdots, I_s$.
3: **for** i=1:N_E **do**
4: FSM – Fill matrix F_S with the rows from matrix F^* that belong to the test image I_i ($F_S = F^*(I_i)$).
5: PFI – Perform fuzzy inference to generate output(s): T_E = FUZZY-INFERENCE(F_S). $T_E = [N_{S_i}, R_{A_i}, R_{D_i}]$
6: The size of each parameters $N_{S_i}, R_{A_i}, R_{D_i} = 1 \times 8$.
7: COP – A single output N_S^*, R_D^* , and R_A^* is calculated for each parameters from $N_{S_i}, R_{D_i}, R_{A_i}$ using fusion (Algorithm 6, GSO).
8: APS – Use N-cut with the parameters N_S^*, R_D^* , and R_A^* to segment I_i.
9: FED – Display segment S and wait for user feedback (user generates a gold standard image G by editing S).
10: **———— *Rule Evolution - Invisible to User* ————**
11: OUP – Input the parameters N_S^*, R_D^* , and R_A^* with G to the simulated annealing optimization technique to estimate the best output N_{S_B}, R_{A_B}, and R_{D_B} (that generate a segment close to G), and store it in T_B.
12: PRU – Pruning step : Discard rows from F_S and T_B that are similar to rows in M and O respectively.
13: AIT – Append the updated matrices F_S and T_B to M and O, respectively.
14: RFR – Regenerate fuzzy rules $R_{F_1}, R_{F_2}, \cdots$ from the updated matrices M and O (e.g., using clustering).
15: **end for**

3.6 Summary

This chapter has presented a detailed explanation of the proposed approaches: EFIS, SC-EFIS, and MSC-EFIS. The basic EFIS approach is designed for the adjustment of a single parametric segmentation techniques. EFIS can be used to improve the performance of any segmentation technique by adjusting its parameter based on the image features. Along with the optimum values

of the parameter, these features are employed in order to train a fuzzy system and create fuzzy rules. The fuzzy rules then become increasingly evolved online as additional images become available. EFIS is also used for switching between the results of more than one segmentation technique and for fusing the individual results. Although EFIS is an efficient approach, it seems to be a technique customized for breast ultrasound images. The several fixed parameters in EFIS limits its suitability for use as a general approach.

An improved, highly automated version of EFIS called SC-EFIS is proposed as a means of overcoming the problems associated with EFIS. SC-EFIS is a general approach that depends on the available images for determination of the set of parameters that would be fixed in EFIS. SC-EFIS has also includes a multistage feature selection approach that enables the selection of the valuable features for the current type of images. In the third proposed approach, a modified version of SC-EFIS called MSC-EFIS allows the adjustment of more than one parameter. As an example, the N-cut segmentation technique was used for testing the proposed MSC-EFIS.

Chapter 4

Experimentation and Results

This chapter describes the experiments conducted in order to test evolving fuzzy image segmentation (EFIS), self configuring EFIS (SC-EFIS), and multi-parametric SC-EFIS (MSC-EFIS). A set of 35 breast ultrasound images and 33 bladder magenetic reasoning (MR) images were employed to train and test the approaches. Segmentation of the breast ultrasound or bladder MR images is challenging, especially with a simple approach such as thresholding. A large number of tests were run in order to verify the performance of the algorithms.

To build the initial fuzzy system, for each training set, a set of five randomly selected images from the data set were used for the extraction of the features along with the optimum parameters as output. This initial fuzzy system was then used to test the proposed methods using the remaining images. The initial fuzzy system evolves as long as new, previously unseen images are input to the system and as long as the segmentation results produced by the algorithms are corrected by an expert user in order to generate optimal parameter values. This process drives the evolution of the fuzzy rules for different segmentation techniques (e.g. region growing, global thresholding, statistical region merging, and normalized cut). During the experimentation, the training-testing cycle was repeated many times. The results of ten different training sets for each segmentation technique in each approach are presented in order to verify the efficiency of the proposed approaches. The number of rules was monitored during the evolution process in order to provide additional information about the algorithm. As well, to verify the convergence of the algorithm, one training set was chosen from the ten training sets and used to illustrate for each

image the improvement from the test set after multiple evolving iterations. All experiments are performed using Matlab. EFIS was developed using Matlab 32-bit, while other experiments used Matlab 64-bit. The experimental results using an image dataset for four different segmentation techniques (region growing, global thresholding, statistical region merging, and normalized cut) are presented.

4.1 Image Data

The first target dataset was developed from 35 breast ultrasound scans that were segmented by an expert. The images were of different dimensions, ranging from 230×390 to 580×760 pixels. Ultrasound images are generally difficult to segment, primarily due to the presence of speckle noise and low level of local contrast. An additional set of 33 bladder MR images were also used as a means of evaluating the N-cut approach. It should be noted that the segmentation of ultrasound and MR images actually does require a complete processing chain, (including proper preprocessing and postprocessing steps). However, the purpose of using these images was solely to demonstrate that the accuracy of the binarization can be increased with the application of EFIS, SC-EFIS, or MSC-EFIS.

4.2 Evaluation Measures

For consideration of two segments S (generated by an algorithm) and G (the gold standard image manually created by an expert), the following measures were applied in order to evaluate the performance of the proposed methods:

- The average of the Jaccard index[1] J (area overlap) [164]:

$$J(S,G) = \frac{|S \cap G|}{|S \cup G|}. \tag{4.1}$$

[1] In medical image segmentation both the Dice similarity coefficient and Jaccard index are widely used. However, Dice values are generally higher than Jaccard values. Jaccard was chosen because it is a more conservative measure.

- The standard deviation σ_J of the Jaccard index
- 95% confidence interval (CI) of the Jaccard index CI_J
- The null hypothesis (t-test) for comparing the results of a parent algorithm and its evolved version in order to verify whether any increase in accuracy is statistically significant

Ground-truth images G were created so that the objects of interest (i.e., lesions and tumours) could be labeled as white (1) and the background as black (0). All thresholding techniques were used consistently to label object pixels in this way.

4.3 Evolving Fuzzy Image Segmentation - EFIS

In this section, the results of testing the EFIS approach are presented for EFIS, EFIS-S, and EFIS-F (described in subsections 3.2.4, 3.2.5).

4.3.1 EFIS for Binarization

Image binarization is the simplest form of segmentation yet remains a somewhat difficult task. Three methods with one parameter p have been chosen to serve as parent algorithms:

1. Region growing (RG) with $p = \epsilon$ (the similarity threshold)
2. Global thresholding (THR) with $p = T$ (the threshold): In this case, the initial fuzzy rules are used (generated during the training phase) in order to estimate the threshold. The results are then compared to the EFIS technique whereby the initial rules are added/modified as additional images are processed.
3. Statistical region merging (SRM) with $p = s$ (the scale)

The results are discussed with respect to visual inspection, accuracy verification, rule evolution, and the analysis of the EFIS results.

Visual Inspection – The results from the parent algorithms (RG, THR, and SRM) and their

evolved versions (EFIS-RG, EFIS-THR, and EFIS-SRM) were visually compared to the corresponding gold standard images in order to confirm the improvement in the accuracy. Sample results are illustrated in Fig. 4.1 shows clear improvement in segmentation.

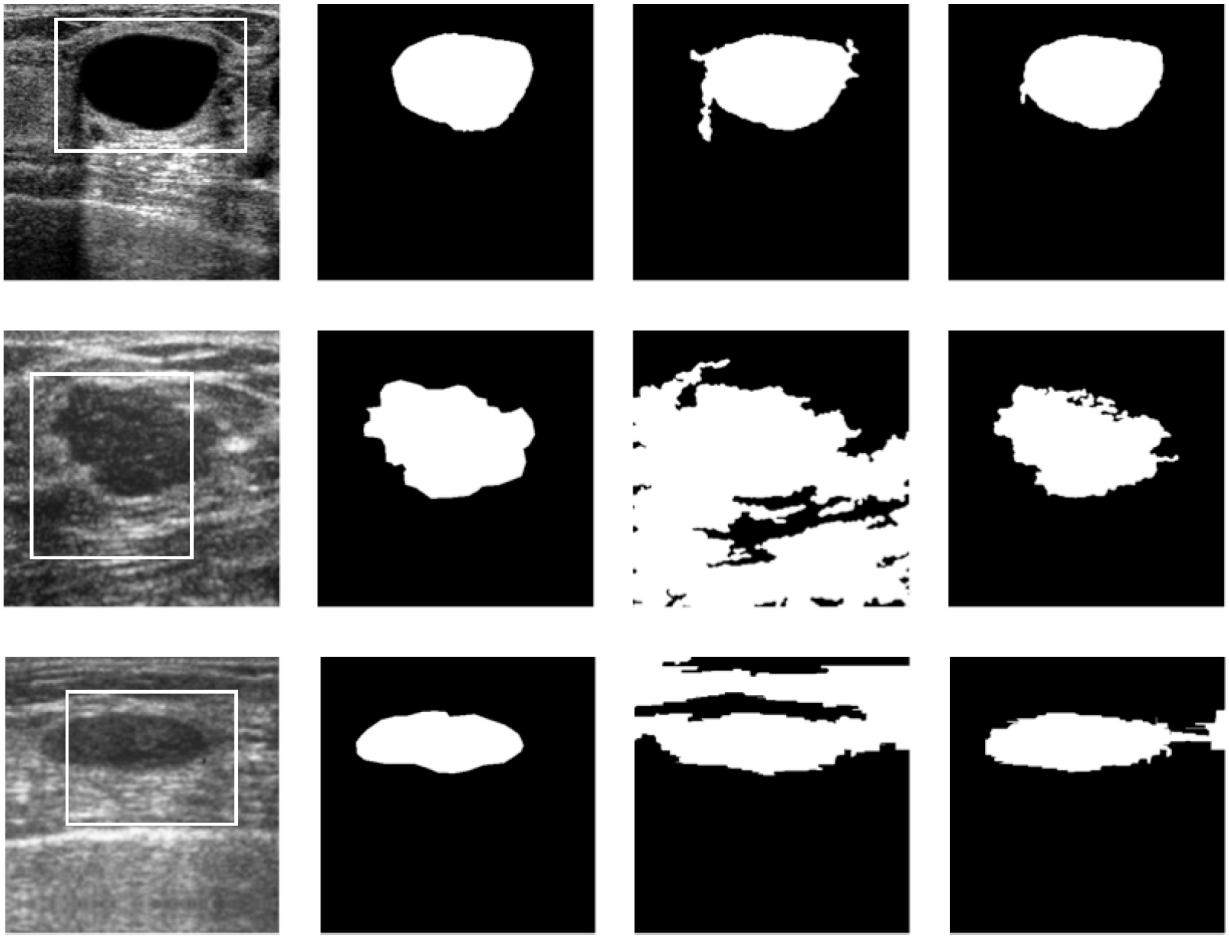

Figure 4.1: Sample results: The first two columns show the original images (along with the detected ROIs) and the corresponding gold standard images (manual segmentation). From top to bottom (the last two columns) indicate THR versus EFIS-THR, RG vesus EFIS-RG, and SRM versus EFIS-SRM.

Accuracy Verification – The results of the ten different runs (each run is an independent experiment trained and tested with different images) are presented in Tables 4.1, 4.2 and 4.3. In all experiments, the accuracy of the EFIS version of the parent algorithm was greater than that

of the parent algorithm alone, and the increase in accuracy was confirmed by the t-test to be statistically significant.

Training	Method	J	σ_J	CI_J
1st run	RG	54%	30%	43%-65%
	EFIS-RG	68%	21%	**60%-76%**
2st run	RG	52%	30%	41%-64%
	EFIS-RG	63%	24%	**54%-72%**
3rd run	RG	54%	30%	43%-65%
	EFIS-RG	65%	25%	**55%-74%**
4th run	RG	54%	31%	42%-65%
	EFIS-RG	64%	23%	**56%-73%**
5th run	RG	54%	29%	43%-65%
	EFIS-RG	66%	21%	**58%-74%**
6th run	RG	52%	30%	41%-63%
	EFIS-RG	64%	23%	**55%-73%**
7th run	RG	57%	29%	46%-68%
	EFIS-RG	67%	24%	**58%-75%**
8th run	RG	53%	30%	42%-64%
	EFIS-RG	64%	25%	**55%-73%**
9th run	RG	53%	31%	41%-64%
	EFIS-RG	66%	23%	**58%-75%**
10th run	RG	57%	29%	46%-67%
	EFIS-RG	68%	21%	**60%-76%**

Table 4.1: Sample results for region growing (RG) and evolved region growing (EFIS-RG) based on training with 5 randomly selected images and testing with 30 in each run, with each run a separate experiment. The default similarity threshold for RG was set to 0.17. The null hypothesis was rejected for 10/10 runs.

Rule Evolution – For any new image being segmented, the rule matrix M may grow, but the number of rules may not necessarily increase. The general tendency seems to be an initial increase in the number of rules followed by a decrease after additional images are processed. Fig. 4.2 provides an example of rule generation over time for EFIS-THR; similar behaviour was observed for EFIS-RG and EFIS-SRM. To verify the effectiveness of the rule evolution, the same

image was processed sequentially; i.e., the same image was repeatedly used as input. As Fig. 4.3 illustrates, for some representative cases, the segmented images reach their highest achievable level of accuracy after an image has been observed multiple times. The highest achievable accuracy for an image is the maximum Jaccard index that can be obtained for that image using a specific technique (e.g., global thresholding).

Training	Method	J	σ_J	CI_J
1rd run	THR	62%	24%	53%-71%
	EFIS-THR	62%	25%	**53%-71%**
2rd run	THR	59%	27%	48%-69%
	EFIS-THR	61%	24%	**52%-70%**
3rd run	THR	54%	31%	42%-65%
	EFIS-THR	63%	25%	**54%-73%**
4rd run	THR	57%	29%	46%-68%
	EFIS-THR	63%	22%	**55%-71%**
5th run	THR	59%	25%	49%-68%
	EFIS-THR	62%	24%	**53%-71%**
6th run	THR	61%	25%	52%-71%
	EFIS-THR	63%	23%	**55%-72%**
7th run	THR	52%	28%	42%-63%
	EFIS-THR	60%	24%	**51%-69%**
8th run	THR	62%	24%	53%-71%
	EFIS-THR	62%	21%	**54%-70%**
9th run	THR	60%	25%	50%-69%
	EFIS-THR	63%	23%	**54%-71%**
10th run	THR	60%	27%	50%-70%
	EFIS-THR	58%	26%	**48%-68%**

Table 4.2: Sample results for global thresholding (THR) and evolved global thresholding (EFIS-THR) training with 5 randomly selected images and testing with 30 in each run with each run a separate experiment. The null hypothesis was rejected for 1/10 run.

Training	Method	J	σ_J	CI_J
1st run	SRM	60%	28%	50%-71%
	EFIS-SRM	71%	19%	**64%-78%**
2nd run	SRM	60%	27%	50%-70%
	EFIS-SRM	69%	22%	**61%-77%**
3rd run	SRM	61%	28%	50%-71%
	EFIS-SRM	67%	24%	**58%-76%**
4th run	SRM	59%	29%	48%-70%
	EFIS-SRM	71%	21%	**63%-79%**
5th run	SRM	59%	29%	49%-70%
	EFIS-SRM	67%	23%	**59%-76%**
6th run	SRM	60%	28%	49%-70%
	EFIS-SRM	69%	21%	**61%-76%**
7th run	SRM	61%	29%	50%-72%
	EFIS-SRM	71%	22%	**62%-79%**
8th run	SRM	59%	28%	48%-69%
	EFIS-SRM	68%	22%	**60%-76%**
9th run	SRM	59%	30%	47%-70%
	EFIS-SRM	71%	22%	**63%-79%**
10th run	SRM	61%	29%	51%-72%
	EFIS-SRM	68%	23%	**60%-77%**

Table 4.3: Sample results for statistical region merging (SRM) and evolved SRM (EFIS-SRM) based on training with 5 randomly selected images and testing with 30 in each run, with each run a separate experiment. The default scale for SRM was set to 32. The null hypothesis was rejected for 8/10 runs.

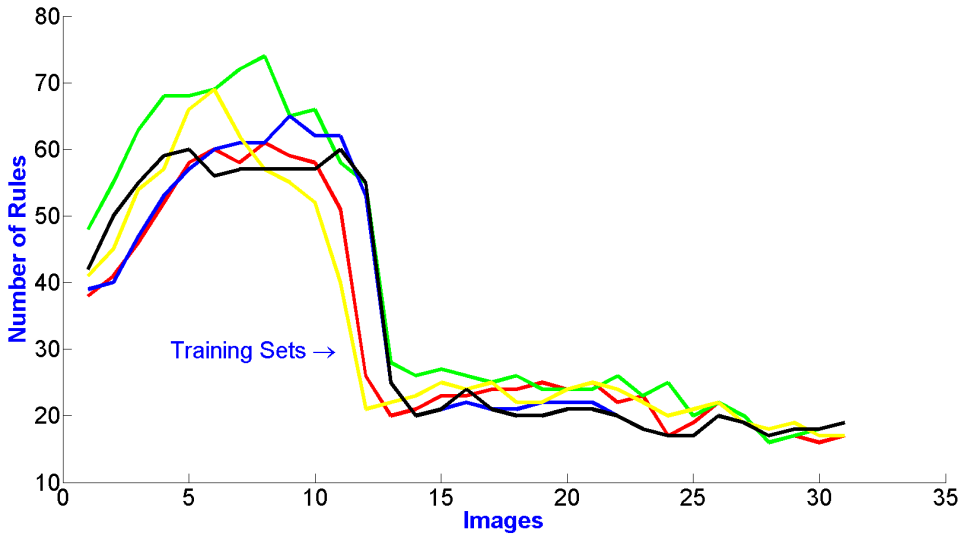

Figure 4.2: Rule evolution for EFIS-THR: The number of rules changes as more images are processed but then converges toward a lower number. Each curve shows the number of rules for a separate run (trained with 5 different images and tested with 30 new images).

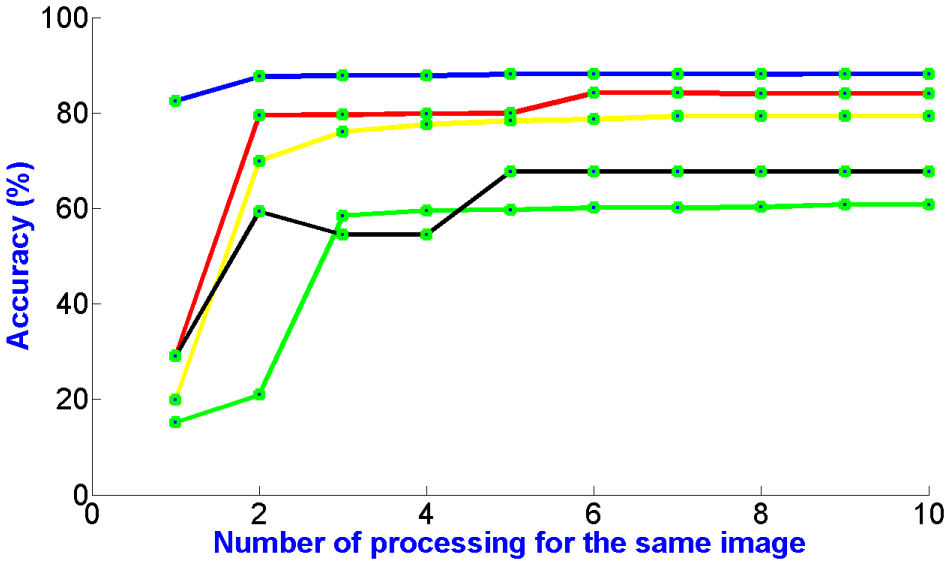

Figure 4.3: Convergence to the maximum degree of accuracy achievable: Sample results, with each curve representing a different image, show that if the same image is processed many times, the accuracy converges toward a maximum degree of accuracy achievable.

Fig. 4.4 show that if the same image is processed many times, the accuracy converges toward a maximum degree of accuracy achievable.

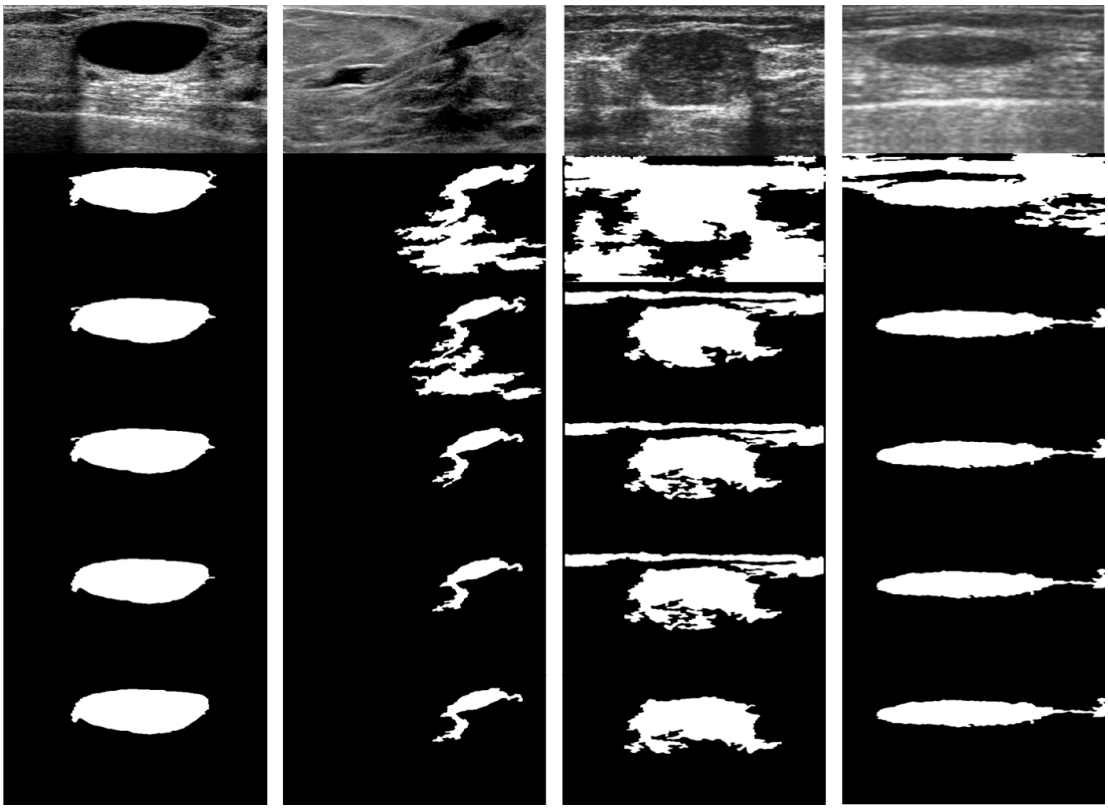

Figure 4.4: Sample results: To demonstrate the positive effect of rule evolution, the same image was processed multiple times (in this example, 5 times).

Analysis of EFIS Switching and Fusion Results – As can be seen in Fig. 4.5 and Table 4.4, SRM delivers better results than Niblack or RG (an average accuracy of $m = 64.7\%$). EFIS-S produces the best results (an average accuracy of 66.5%). EFIS-F produces an accuracy level almost identical to that of the best individual method, namely SRM, but with slightly lower variability ($64.3\% \pm 14\%$).

Figure 4.5: Sample results: (from left to right) results for Niblack, SRM, and RG and for all three fused by STAPLE. Note that the result of the fusion correctly reflects the fact that SRM is the best overall method based on the experiments.

Dataset	Niblack	SRM	RG	EFIS-S	EFIS-F
1	76%	68%	50%	77%	77%
2	52%	55%	48%	53%	53%
3	77%	74%	72%	80%	72%
4	74%	57%	55%	54%	56%
5	43%	33%	33%	36%	34%
6	59%	59%	62%	62%	62%
7	55%	82%	80%	81%	78%
8	62%	62%	58%	66%	66%
9	68%	64%	63%	76%	70%
10	59%	90%	89%	79%	76%
m	62.3%	64.7%	61.0%	66.5%	64.3%
σ	11%	16%	16%	15%	14%

Table 4.4: Accuracy of switching (EFIS-S) and fusion (EFIS-F) for three methods: Niblack, SRM (scale = 32), and region growing (similarity = 0.17). Each dataset had 30 images for training and 5 images for testing.

4.3.2 Comparison of EFIS Results

To compare the results of EFIS for image thresholding (EFIS-THR) with those produced by other methods, four global thresholding methods were used: two fuzzy and two non-fuzzy. A "local" thresholding method was also used: local methods are generally expected to outperform global methods. Specifically, EFIS-THR was compared with the following:

- Huang-Wang method – based on optimization of fuzzy entropy [165]
- Tizhoosh method – fuzzy thresholding using interval-valued fuzzy sets and optimization of the index of ultrafuzziness [30]
- Otsu method – one of the most popular methods, based on bimodality and variance [22]
- Kittler method – based on error minimization [24], reportedly one of the best thresholding algorithms [6]
- Niblack method – a local method based on the calculation of the local mean and standard deviation [29]

Table 4.5 shows the results for three different (representative) runs in which the average and standard deviation of the Jaccard index $J \pm \sigma_J$ as well as the 95% confidence interval CI_J of all accuracy numbers were calculated for each method and for each run/experiment. Five images were randomly selected for thresholding in each run (with EFIS-THR using the remaining 30 for training). To achieve good benchmarking, the maximum achievable accuracy (MAA) was also calculated for each experiment. MAA is the highest accuracy level (via global thresholding) when the optimal threshold is selected through exhaustive search and comparison with gold standard images. MAA is hence the upper bound of the degree of accuracy, and no global thresholding technique can exceed it for the images used in each experiment. When the comparative experiments were analyzed, in all experiments, EFIS-THR produced the highest level of accuracy. Other global thresholding methods achieved predictably poor results since the test images are extremely difficult to binarize via global thresholding, and these methods are static and have no learning capability. The local Niblack method produced the second best results, but this method was extremely slow compared to the others (≈ 10 times slower).

Run	Method	Type	$J \pm \sigma_J$	CI_J
1	MAA	–	79%±12%	[75% 84%]
	EFIS-THR	global	**62%±25%**	**[53% 71%]**
	Niblack	local	56%±24%	[47% 65%]
	Huang	global	45%±27%	[35% 55%]
	Kittler	global	39%±32%	[27% 51%]
	Tizhoosh	global	35%±32%	[23% 47%]
	Otsu	global	28%±25%	[18% 37%]
2	MAA	–	79%±11%	[75% 83%]
	EFIS-THR	global	**60%±24%**	**[51% 69%]**
	Niblack	local	57%±25%	[48% 66%]
	Huang	global	44%±29%	[34% 55%]
	Kittler	global	41%±31%	[29% 52%]
	Tizhoosh	global	38%±32%	[26% 50%]
	Otsu	global	29%±25%	[19% 38%]
3	MAA	–	79%±12%	[74% 83%]
	EFIS-THR	global	**63%±23%**	**[54% 71%]**
	Niblack	local	59%±24%	[49% 68%]
	Huang	global	46%±27%	[35% 56%]
	Kittler	global	41%±33%	[29% 53%]
	Tizhoosh	global	35%±33%	[23% 48%]
	Otsu	global	28%±23%	[20% 37%]

Table 4.5: Comparison of the results of EFIS-THR (EFIS for global thresholding) and four other global thresholding as well as one local thresholding method: Average and standard deviation of the Jaccard index $J \pm \sigma_J$ and 95% confidence interval CI_J. The MAA indicates the maximum achievable accuracy determined via exhaustive search and through comparison with gold standard images; no global thresholding method can achieve higher accuracies than the MAA.

4.4 Testing SC-EFIS

In this section, the SC-EFIS results for RG, global thresholding, and SRM are presented. The results are discussed with respect to visual inspection, accuracy verification, rule evolution, and the analysis of the Jaccard results.

Visual Inspection - A visual inspection of Fig. 4.6 shows that the results produced by the proposed SC-EFIS-RG represent a substantial improvement over those obtained with the FRG

(fuzzy RG – the initial fuzzy rules are used in order to estimate the similarity threshold). A visual inspection of Fig. 4.7 revails a significant improvement in the SC-EFIS-SRM images over the SRM ones.

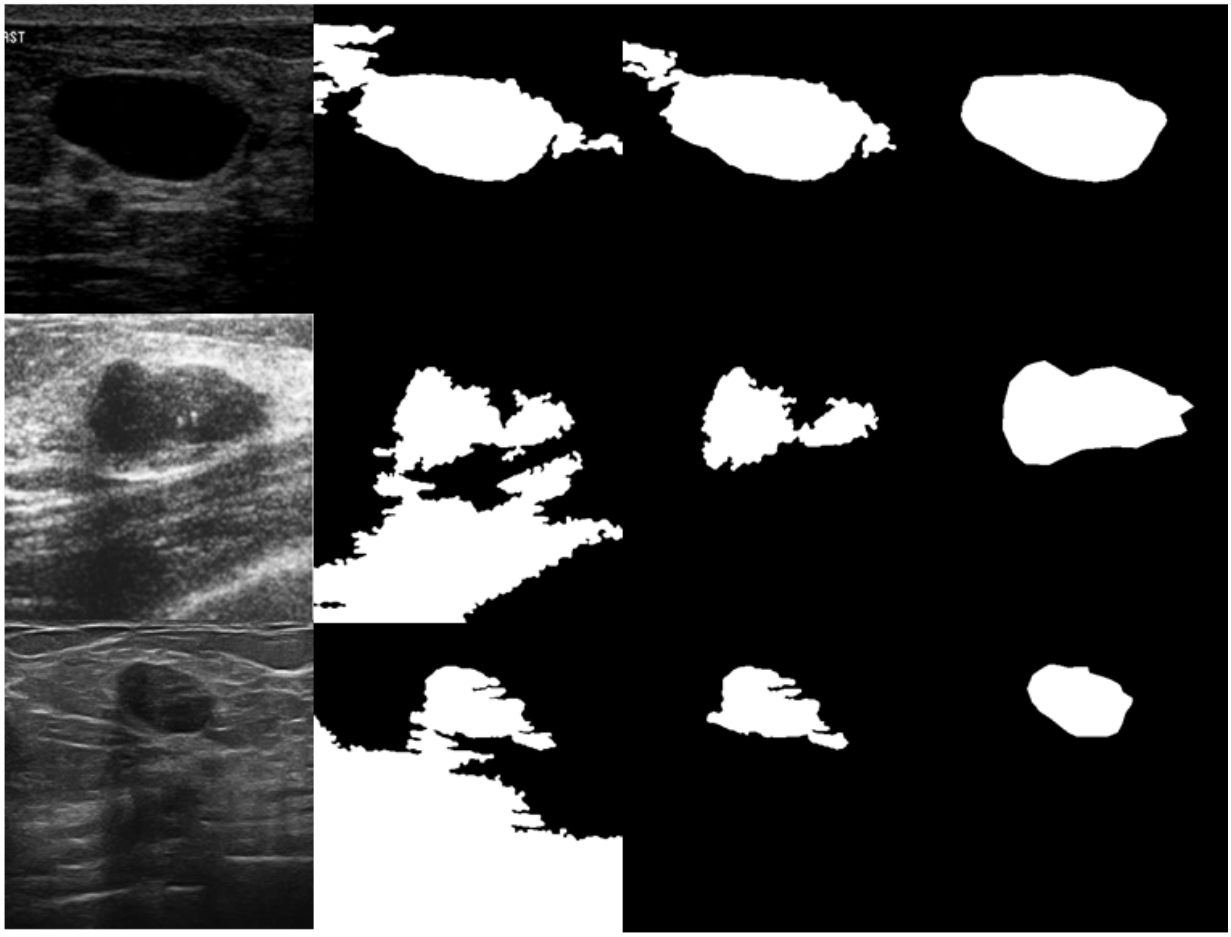

Figure 4.6: Segmentation results: From left to right, the original image, FRG, SC-EFIS-RG, and the gold standard image.

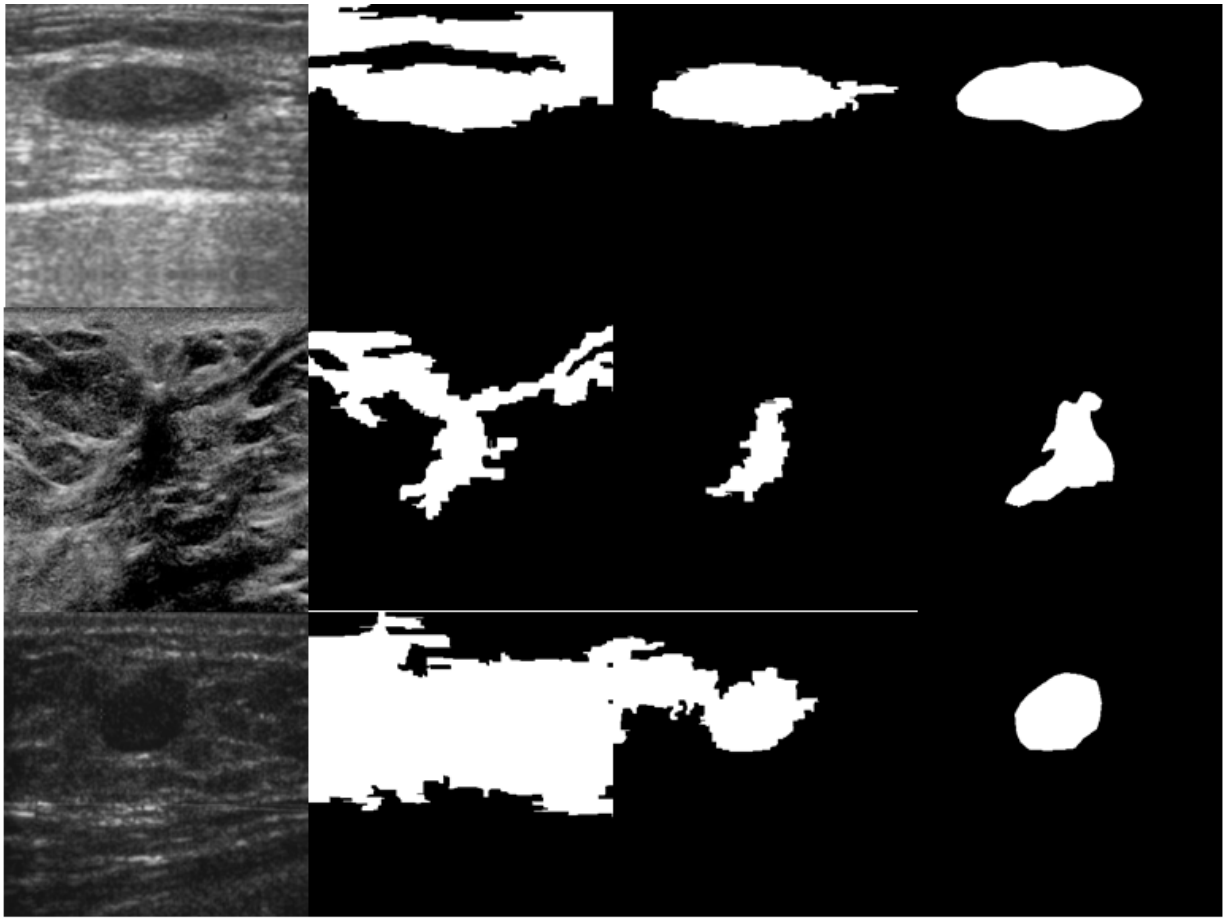

Figure 4.7: segmentation results: From left to right, the original image, SRM, SC-EFIS-SRM, and the gold standard image.

Accuracy Verification - Ten different runs are presented for each method. Each run is an independent experiment involving training and testing with different images.

Table 4.6 presents a comparison of the results for the RG technique: RG results with fuzzy inference, RG results with similarity threshold = 0.17, RG with best similarity threshold (0.12) for the available data (RG-B), the EFIS-RG technique, and the SC-EFIS-RG technique. The best similarity threshold is determined via exhaustive search. However, this selection is a manual process that requires an exhaustive search of all possible values of similarity threshold for a given dataset for the best value to be detected. It can be seen that the results achieved with EFIS are

better than those from SC-EFIS in only two of ten experiments, and that the SC-EFIS results are better than the EFIS results in eight experiments.

Training	Metrics	FRG	RG	RG-B	EFIS-RG	SC-EFIS-RG
1st run	J	63%	54%	69%	68%	67%
	σ_J	26%	30%	21%	21%	23%
	CI_J	53%-73%	43%-65%	62%-77%	60%-76%	58%-75%
2nd run	J	37%	52%	69%	63%	66%
	σ_J	35%	31%	19%	24%	22%
	CI_J	24%-50%	41%-64%	62%-76%	54%-72%	57%-74%
3rd run	J	43%	54%	70%	65%	68%
	σ_J	31%	30%	21%	25%	21%
	CI_J	31%-54%	43%-65%	63%-78%	55%-74%	61%-76%
4th run	J	33%	54%	71%	64%	66%
	σ_J	33%	31%	20%	23%	24%
	CI_J	21%-46%	42%-65%	63%-78%	56%-73%	57%-74%
5th run	J	46%	54%	71%	66%	67%
	σ_J	32%	29%	17%	21%	20%
	CI_J	34%-58%	43%-65%	64%-77%	58%-74%	60%-74%
6th run	J	46%	52%	69%	64%	62%
	σ_J	31%	30%	20%	23%	24%
	CI_J	35%-58%	41%-63%	61%-76%	55%-73%	53%-71%
7th run	J	61%	57%	70%	67%	68%
	σ_J	28%	29%	21%	24%	23%
	CI_J	51%-71%	46%-68%	62%-78%	58%-75%	59%-76%
8th run	J	56%	53%	70%	64%	67%
	σ_J	30%	30%	20%	25%	23%
	CI_J	45%-67%	42%-64%	62%-78%	55%-73%	59%-75%
9th run	J	37%	53%	70%	64%	66%
	σ_J	29%	31%	20%	25%	23%
	CI_J	26%-48%	41%-64%	63%-78%	55%-73%	58%-75%
10th run	J	57%	57%	71%	66%	69%
	σ_J	29%	29%	18%	23%	21%
	CI_J	46%-68%	46%-68%	64%-78%	58%-75%	61%-77%

Table 4.6: Sample results for fuzzy region growing (FRG), RG with a similarity threshold (0.17), RG-B with the best similarity threshold (0.12) (determined via exhaustive search), EFIS-RG, and SC-EFIS-RG. The null hypothesis was rejected in 10/10 runs.

Table 4.7 presents a comparison of the results for the global thresholding technique: THR, the EFIS-THR results, and the SC-EFIS-THR results. It is clear that the SC-EFIS results surpass the EFIS ones in six of ten experiments.

Training	Method	J	σ_J	CI_J
1st run	THR	58%	24%	49%-67%
	EFIS-THR	62%	25%	53%-71%
	SC-EFIS-THR	63%	23%	54%-72%
2nd run	THR	48%	33%	35%-60%
	EFIS-THR	61%	24%	52%-70%
	SC-EFIS-THR	61%	28%	51%-72%
3rd run	THR	43%	32%	31%-55%
	EFIS-THR	63%	25%	54%-73%
	SC-EFIS-THR	63%	26%	53%-72%
4th run	THR	23%	23%	14%-32%
	EFIS-THR	63%	22%	55%-71%
	SC-EFIS-THR	66%	21%	58%-74%
5th run	THR	54%	26%	44%-64%
	EFIS-THR	62%	24%	53%-71%
	SC-EFIS-THR	63%	25%	54%-73%
6th run	THR	55%	30%	44%-66%
	EFIS-THR	63%	23%	55%-72%
	SC-EFIS-THR	64%	23%	55%-72%
7th run	THR	38%	27%	28%-48%
	EFIS-THR	60%	24%	51%-69%
	SC-EFIS-THR	59%	26%	49%-69%
8th run	THR	52%	24%	43%-62%
	EFIS-THR	62%	21%	54%-70%
	SC-EFIS-THR	63%	21%	55%-70%
9th run	THR	39%	31%	28%-51%
	EFIS-THR	63%	23%	54%-73%
	SC-EFIS-THR	65%	21%	57%-73%
10th run	THR	44%	25%	34%-53%
	EFIS-THR	58%	26%	48%-68%
	SC-EFIS-THR	57%	26%	47%-67%

Table 4.7: Sample results for global thresholding: fuzzy thresholding (THR), EFIS-THR, and SC-EFIS-THR. The null hypothesis was rejected in 9/10 runs.

However, EFIS produces better results in two experiments and equal results in another two.

Fig. 4.8 shows the improvement in the Jaccard index of the SC-EFIS-SRM images and the images for SRM with a scale = 32. Table 4.8 presents a comparison of the results for the SRM technique: results for SRM using fuzzy inference FSRM, results for SRM with a scale = 32 (SRM), results for SRM with the best scale (64) for the available images (SRM-B) determined via exhaustive search, EFIS-SRM results, and SC-EFIS-SRM results. It can be seen that the results produced by SC-EFIS are superior to the EFIS results in five experiments, inferior in another four experiments, and the same in the remaining experiments.

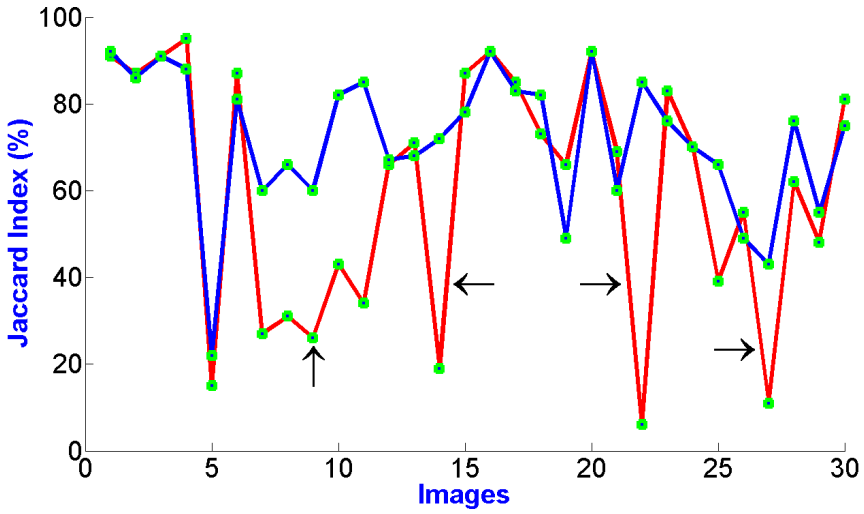

Figure 4.8: Comparison of the Jaccard accuracy obtained with SC-EFIS-SRM (blue) and with SRM (red); arrows point to significant gaps.

Rule Evolution - Fig. 4.9 indicates the change in the number of rules during the evolution of the thresholding (THR) process. The initial number of rules starts to increase with new, incoming images and then begins to decrease as additional images become available. The same behaviour was noted for both SRM and RG.

Training	Metrics	FSRM	SRM	SRM-B	EFIS-SRM	SC-EFIS-SRM
1st run	J	64%	60%	71%	71%	72%
	σ_J	24%	28%	21%	19%	17%
	CI_J	55%-73%	50%-71%	64%-79%	64%-78%	65%-78%
2nd run	J	67%	60%	67%	69%	67%
	σ_J	25%	27%	24%	22%	20%
	CI_J	57%-76%	50%-70%	59%-76%	61%-77%	60%-75%
3rd run	J	63%	61%	70%	67%	69%
	σ_J	25%	28%	22%	24%	18%
	CI_J	53%-72%	50%-71%	62%-78%	58%-76%	62%-76%
4th run	J	57%	59%	69%	71%	71%
	σ_J	28%	29%	24%	21%	19%
	CI_J	46%-67%	48%-70%	60%-78%	63%-79%	64%-78%
5th run	J	42%	59%	68%	67%	68%
	σ_J	33%	29%	24%	23%	22%
	CI_J	30%-54%	49%-70%	59%-77%	59%-76%	60%-77%
6th run	J	63%	60%	69%	69%	68%
	σ_J	26%	28%	22%	21%	20%
	CI_J	53%-73%	49%-70%	61%-77%	61%-76%	61%-76%
7th run	J	55%	61%	71%	71%	70%
	σ_J	30%	29%	23%	22%	20%
	CI_J	44%-67%	50%-71%	62%-79%	62%-79%	63%-78%
8th run	J	67%	59%	70%	68%	69%
	σ_J	19%	28%	22%	22%	20%
	CI_J	60%-74%	48%-69%	62%-78%	60%-76%	62%-76%
9th run	J	47%	58%	69%	71%	67%
	σ_J	31%	30%	24%	22%	24%
	CI_J	36%-59%	47%-70%	60%-78%	63%-79%	58%-76%
10th run	J	64%	61%	69%	68%	71%
	σ_J	28%	29%	24%	23%	19%
	CI_J	54%-74%	51%-72%	60%-78%	60%-77%	64%-78%

Table 4.8: Sample results for fuzzy statistical region merging (FSRM), SRM with the default scale (32), SRM-B with the best scale (64) (determined via exhaustive search), EFIS-SRM, and SC-EFIS-SRM. The null hypothesis was rejected in 8/10 runs.

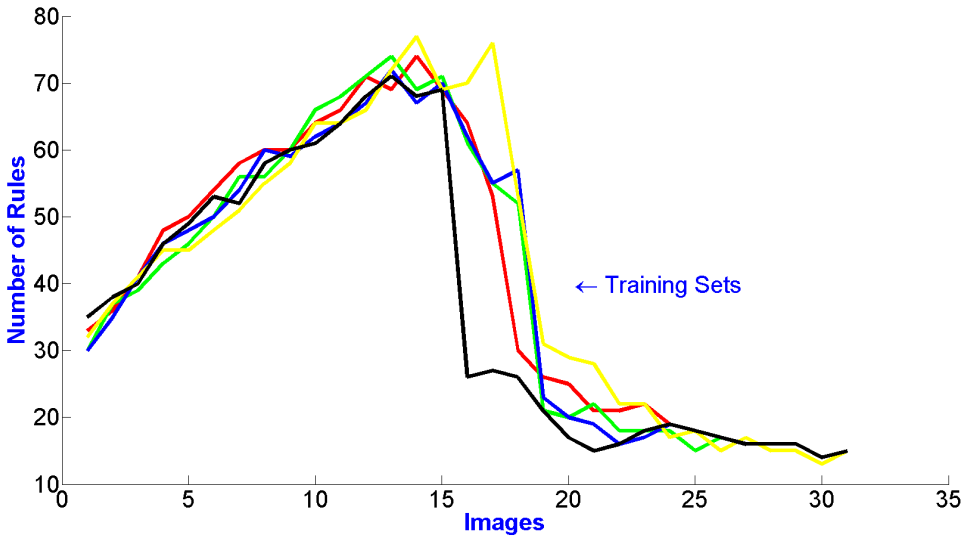

Figure 4.9: Rule evolution for SC-EFIS-THR: The number of rules increases as more images are processed but then converges toward a lower number. Each curve shows the number of rules for a separate run.

In general, SC-EFIS is able to be competitive with and can even surpass EFIS with respect to the three segmentation techniques, while offering a higher level of automation.

On the other hand, the switch/fusion technique [159] was re-examined for use with SC-EFIS. Table 4.9 presents the results of switching and fusion for three methods: Niblack, SRM (scale=32), and RG (similarity = 0.17) using EFIS (EFIS-S and EFIS-F) and using SC-EFIS (SC-EFIS-S and SC-EFIS-F). It is clear that the outcomes with EFIS are superior to the results with SC-EFIS. In addition, the results with EFIS-S and SC-EFIS-S surpass those for SRM, which represents the best method. However, the results of SC-EFIS-F are considered the poorest of all of the EFIS and SC-EFIS results.

Dataset	Niblack	SRM	RG	EFIS-S	EFIS-F	SC-EFIS-S	SC-EFIS-F
1	76%	68%	50%	77%	77%	76%	65%
2	52%	55%	48%	53%	53%	62%	52%
3	77%	74%	72%	80%	72%	80%	81%
4	74%	57%	55%	54%	56%	65%	67%
5	43%	33%	33%	36%	34%	30%	29%
6	59%	59%	62%	62%	62%	61%	56%
7	55%	82%	80%	81%	78%	62%	78%
8	62%	62%	58%	66%	66%	63%	57%
9	68%	64%	63%	76%	70%	73%	65%
10	59%	90%	89%	79%	76%	76%	77%
m	62.3%	64.7%	61.0%	66.5%	64.3%	64.9%	62.7%
σ	11%	16%	16%	15%	14%	14%	15%

Table 4.9: Accuracy of switching and fusion for three methods: Niblack, SRM, and RG using EFIS and SC-EFIS: Each dataset had 30 images for training and 5 images for testing.

Table 4.10 enables a comparison of EFIS and SC-EFIS results for global thresholding with different global and local thresholding techniques. The data listed are taken form three experiments selected from Table 4.7. It is clear that, in the three experiments, EFIS and SC-EFIS provide outcomes that are more accurate than those produced with the non-evolutionary thresholding techniques.

Run	Method	Type	$J \pm \sigma_J$	CI_J
1	MAA	–	79%±12%	[75% 84%]
	EFIS-THR	global	**62%±25%**	**[53% 71%]**
	SC-EFIS-THR	global	**63%±23%**	**[54% 72%]**
	Niblack	local	56%±24%	[47% 65%]
	Huang	global	45%±27%	[35% 55%]
	Kittler	global	39%±32%	[27% 51%]
	Tizhoosh	global	35%±32%	[23% 47%]
	Otsu	global	28%±25%	[18% 37%]
2	MAA	–	79%±11%	[75% 83%]
	EFIS-THR	global	**60%±24%**	**[51% 69%]**
	SC-EFIS-THR	global	**59%±26%**	**[49% 69%]**
	Niblack	local	57%±25%	[48% 66%]
	Huang	global	44%±29%	[34% 55%]
	Kittler	global	41%±31%	[29% 52%]
	Tizhoosh	global	38%±32%	[26% 50%]
	Otsu	global	29%±25%	[19% 38%]
3	MAA	–	79%±12%	[74% 83%]
	EFIS-THR	global	**63%±23%**	**[54% 71%]**
	SC-EFIS-THR	global	**65%±21%**	**[57% 73%]**
	Niblack	local	59%±24%	[49% 68%]
	Huang	global	46%±27%	[35% 56%]
	Kittler	global	41%±33%	[29% 53%]
	Tizhoosh	global	35%±33%	[23% 48%]
	Otsu	global	28%±23%	[20% 37%]

Table 4.10: Comparison of EFIS, SC-EFIS, and four other global thresholding technique as well as one local thresholding method: Average and standard deviation of the Jaccard index $J \pm \sigma_J$ and 95% confidence interval CI_J. The MAA indicates the maximum achievable accuracy determined via exhaustive search and through comparison with gold standard images; no global thresholding method can achieve higher accuracies than the MAA.

4.5 Testing MSC-EFIS

This section presents the experimental results obtained when multiple parameters of the N-cut technique are adjusted. A set of 33 bladder MR and 35 breast ultrasound images were used

to evaluate the performance of the proposed algorithm. The results of applying the algorithms explained in subsections 3.5.1, and 3.5.2 are presented in the following subsections.

4.5.1 Determining the Number of Parameters to Be Adjusted

This subsection presents the results of the training, testing, and the evolving of four different independent evolving fuzzy systems for the purpose of determining the number of parameters that should be adjusted in the N-cut technique. The processes are performed for the four parameters associated with the breast ultrasounds and the bladder MR images. For each parameter, five different experiments were conducted, with five randomly selected images being used for training and the remaining images for testing/evolving. The results for each system were compared with the results using the default values. As is clear from Tables 4.11 and 4.12, no value is to be gained by including the fourth parameter, namely the edge variance (E_V), in the adjustment solution because its Jaccard index drops dramatically when adjusted compared to the default values in most of the training sets either for the bladder or for the breast images. For example, in the first training set, Jaccard index fell from 62% using the default bladder parameters to 42% when adjusted using the evolving fuzzy system. On the other hand, the other three parameters show either improvement or stability when adjusted using the evolving fuzzy system. Based on the use of the proposed MSC-EFIS evolving fuzzy system, three parameters were selected for adjustment: N_S, R_A, and R_D.

Training	Metric	DNS	ENS	DD	ED	DR	ER	DEV	EEV
1st set	J	58%	66%	62%	66%	59%	58%	62%	42%
	σ_J	29%	22%	29%	27%	29%	30%	28%	23%
2nd set	J	58%	62%	59%	62%	61%	61%	60%	45%
	σ_J	29%	25%	29%	28%	29%	29%	29%	27%
3rd set	J	62%	71%	58%	61%	64%	64%	60%	47%
	σ_J	29%	21%	28%	26%	29%	28%	29%	27%
4th set	J	60%	63%	60%	63%	59%	60%	58%	43%
	σ_J	28%	24%	28%	27%	29%	29%	29%	24%
5th set	J	57%	63%	59%	62%	57%	57%	58%	40%
	σ_J	29%	24%	29%	28%	29%	29%	28%	22%

Table 4.11: Results for the four independent evolving fuzzy systems for each parameter for bladder MRI segmentation: default N_S (DNS), evolving N_S (ENS), default R_D, evolving R_D(ED), default R_A (DR), evolving R_A (ER), default E_V (DEV), and evolving E_V (EEV).

Training	Metric	DNS	ENS	DD	ED	DR	ER	DEV	EEV
1st set	J	54%	71%	57%	57%	56%	55%	56%	54%
	σ_J	30%	22%	30%	30%	31%	30%	32%	32%
2nd set	J	57%	71%	56%	55%	54%	52%	54%	48%
	σ_J	30%	22%	30%	30%	30%	29%	31%	31%
3rd set	J	54%	70%	55%	54%	58%	57%	60%	59%
	σ_J	31%	23%	31%	30%	29%	28%	28%	29%
4th set	J	58%	75%	59%	58%	60%	60%	57%	54%
	σ_J	29%	19%	29%	29%	30%	30%	29%	30%
5th set	J	54%	71%	56%	55%	54%	54%	58%	56%
	σ_J	32%	23%	30%	29%	29%	28%	30%	31%

Table 4.12: Results for the four independent evolving fuzzy systems for each parameter for breast ultrasound segmentation: default N_S (DNS), evolving N_S (ENS), default R_D, evolving R_D(ED) default R_A (DR), evolving R_A (ER), default E_V (DEV),and evolving E_V (EEV).

4.5.2 Adjusting the Selected Parameters

In this section, the results of applying the proposed algorithm MSC-EFIS are presented for bladder and breast images. MSC-EFIS was used to adjust three parameters (N_S, R_A, and R_D) in either bladder or breast images. Ten different experiment were performed for both the bladder and breast images with 5 randomly selected images as training images and the remaining images for the testing and evolving the system (each run is an independent experiment trained and tested with different images). The simulated annealing (SA) technique used in the online updating of the parameters may give different estimates of the parameters when repeated, even for the same image (section 3.5.2). Therefore, to verify generality, each experiment was repeated five times, and the resulting best, worst, and average values of the Jaccard index for the five repeated runs were recorded. The proposed approach was compared with the results obtained using the default values. Three different values of N_S were used for comparison: $N_S = 5$, $N_S = 8$, and $N_S = 10$. The results for the bladder and breast images are discussed with respect to visual inspection, accuracy verification, rule evolution, and the analysis of the The Jaccard results.

Visual Inspection - A visual inspection of the results in Fig. 4.10 and Fig. 4.11 shows a significant improvement in the segmentations obtained using the proposed MSC-EFIS approach for the bladder MR and the breast ultrasound images when compared with NC5 (N-cut with $N_S = 5$).

Accuracy Verification - The results for the bladder images were recorded and are presented in Table 4.13. It can be seen that, for the bladder images, the proposed MSC-EFIS approach produce significant improvement compared to the NC5 results. For example, in the first training set, the average Jaccard index of the proposed approach is increased from 55% to 62%, the standard deviation is decreased from 28% to 25%, and the confidence interval is shifted from 45%-66% to 52%-72%. The MSC-EFIS approach also yielded results that surpass those generated with NC10 in all cases and compete favorably with those generated using the NC8 default values which seem to be the optimum selection for this set of data. With respect to the average Jaccard values, those produces with MSC-EFIS surpass those obtained with NC8 in two training sets and are equal in another two training sets. The results for the best value surpass those obtained with NC8 in six training sets and are equal in another training sets.

With respect to the breast ultrasound images (Table 4.14), the best value for N_S that returns the maximum Jaccard value shifted from 8 to 10. It is clear that the best, average, and worst results of MSC-EFIS surpass those generated by NC5 and NC8 in all cases. For example, in

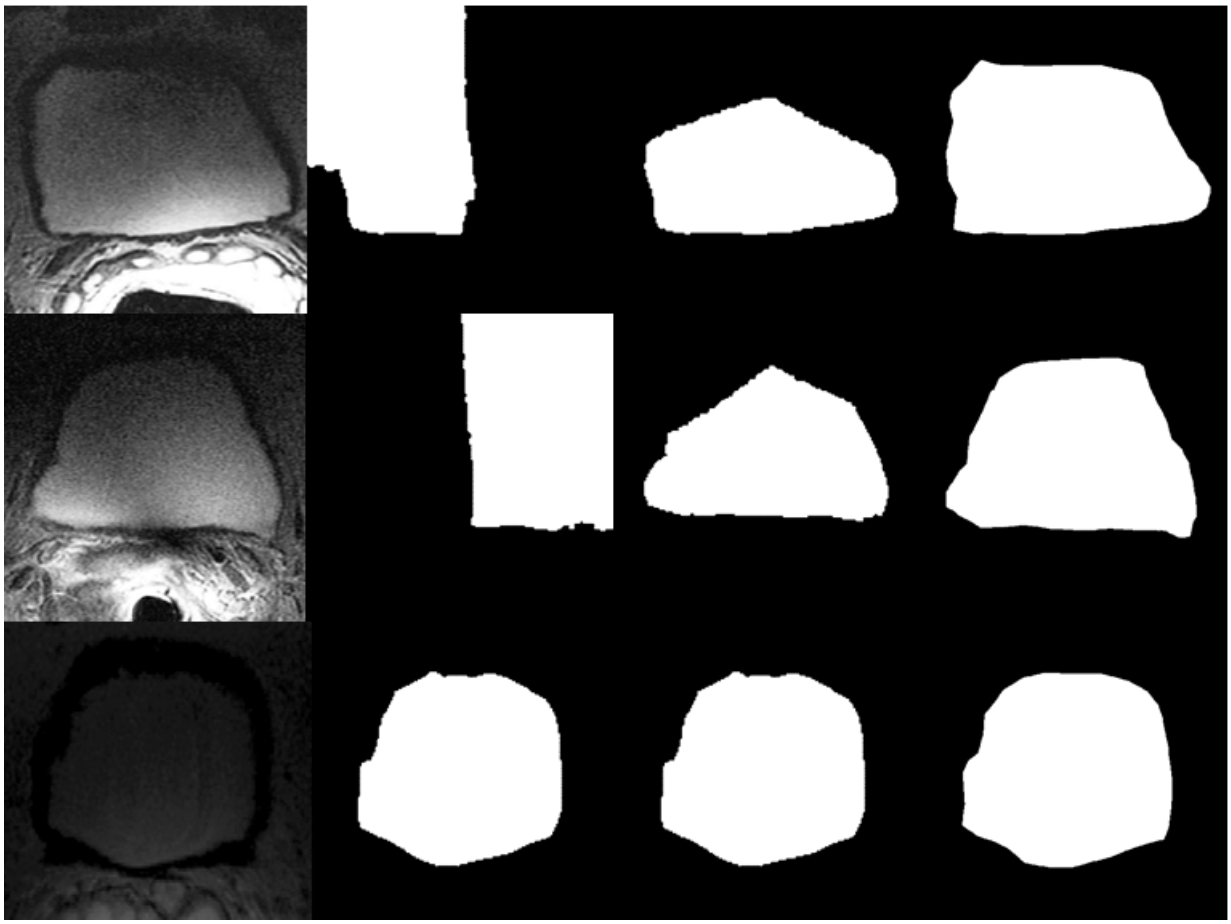

Figure 4.10: Sample results for three segmented bladder images: From left to right, original image, NC5, MSC-EFIS, and the gold standard segment prepared by an expert.

the first training set, with the proposed approach, the average Jaccard index is increased from 57% (NC5) to 72%, the standard deviation is decreased from 30% to 22%, and the confidence interval is shifted from 46%-68% to 63%-80%. With respect to the best results, the results with MSC-EFIS also surpass those generated by NC10 in seven cases and are equal in three cases, while for the average, the MSC-EFIS results are superior to those with NC10 in three training sets and are equal in four sets. Fig. 4.12 shows the improvement in the Jaccard index of the corresponding images when the results obtained with the MSC-EFIS approach applied to breast images are compared with those with NC5.

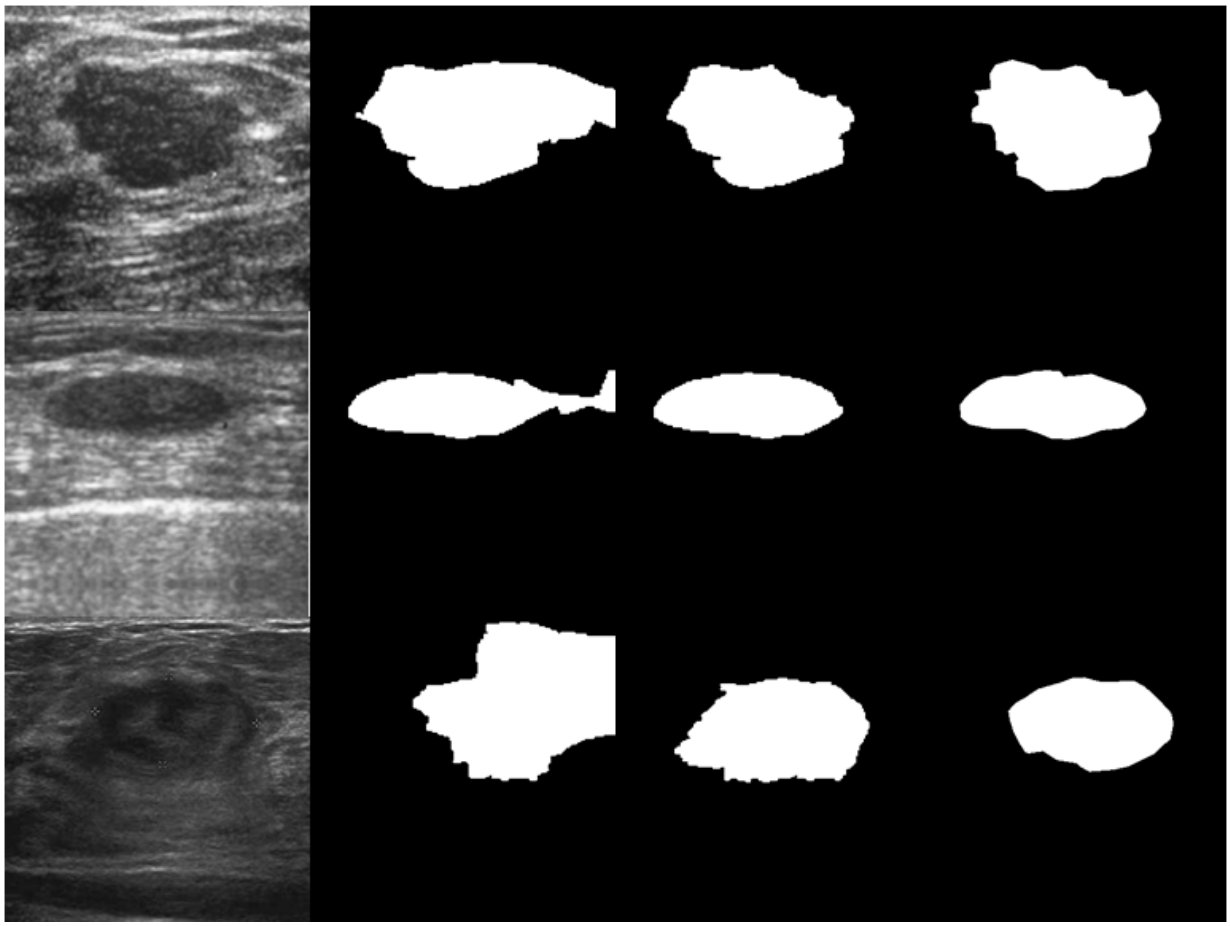

Figure 4.11: Sample results for three segmented breast images: From left to right, original image, NC5, MSC-EFIS, and the gold standard segment prepared by an expert.

Rule Evolution - For the breast ultrasound images, Fig. 4.13 shows the changes in the number of rules during the evolution process for the first five training sets (first run of each). Unlike the behaviour noted in Fig. 4.9, the number of rules starts low, then increases during the evolution process, and continues to increase for as long as new images are introduced. The same behaviour was recorded for the bladder images Fig. 4.14 except in two cases, a dramatic drop is evident at the last image. This effect occurs because the consequent part of the rules now contains more than one parameter (three parameters).

Training	Metrics	NC5	NC8	NC10	MSC-EFIS		
					Best	Average	Worst
1st set	J	55%	67%	58%	65%	62%	58%
	σ_J	28%	20%	20%	23%	25%	26%
	CI_J	45%-66%	60%-75%	50%-66%	57%-74	52%-72%	48%-69%
2nd set	J	62%	70%	58%	71%	69%	68%
	σ_J	28%	21%	21%	24%	25%	26%
	CI_J	51%-73%	62%-78%	50%-66%	62%-80	60%-79%	58%-78%
3rd set	J	59%	69%	61%	66%	66%	65%
	σ_J	29%	21%	20%	24%	25%	25%
	CI_J	48%-70%	61%-77%	53%-68%	57%-76	56%-76%	55%-75%
4th set	J	64%	72%	61%	74%	72%	71%
	σ_J	29%	21%	21%	22%	25%	27%
	CI_J	52%-75%	64%-80%	53%-69%	66%-83	62%-81%	60%-81%
5th set	J	62%	68%	60%	69%	68%	66%
	σ_J	29%%	22%	21%	24%	25%	27%
	CI_J	51%-73%	60%-77%	52%-68%	60%-78	58%-77%	55%-76%
6th set	J	60%	71%	59%	69%	68%	68%
	σ_J	29%	22%	22%	24%	24%	24%
	CI_J	49%-72%	62%-79%	50%-67%	60%-79	59%-78%	58%-77 %
7th set	J	58%	68%	58%	70%	67%	64%
	σ_J	29%	21%	21%	22%	24%	25%
	CI_J	46%-69%	60%-76%	50%-66%	62%-79	58%-76%	54%-73%
8th set	J	62%	71%	59%	73%	72%	70%
	σ_J	30%	22%	22%	21%	22%	24%
	CI_J	51%-74%	62%-79%	50%-68%	65%-82	63%-81%	61%-80%
9th set	J	62%	69%	57%	73%	71%	70%
	σ_J	28%	21%	21%	21%	22%	24%
	CI_J	52%-73%	61%-78%	49%-65%	65%-81	63%-80%	60%-79%
10th set	J	58%	70%	59%	70%	69%	69%
	σ_J	29%	20%	21%	23%	24%	24%
	CI_J	47%-70%	62%-77%	51%-67%	61%-79	60%-78%	60%-78%

Table 4.13: Results for bladder MR segmentation: For each training set (each training set is repeated five times), the results produced with NC5 ($N_S = 5$), NC8 ($N_S = 8$), and NC10 ($N_S = 10$) are compared with the best, average, and worst results obtained with MSC-EFIS for the N-cut technique.

Training	Metrics	NC5	NC8	NC10	MSC-EFIS		
					Best	Average	Worst
1st set	J	57%	65%	71%	72%	72%	71%
	σ_J	30%	27%	24%	20%	22%	25%
	CI_J	46%-68%	54%-75%	62%-79%	65%-79	63%-80%	61%-80%
2nd set	J	52%	63%	69%	70%	69%	69%
	σ_J	31%	28%	24%	20%	20%	21%
	CI_J	40%-63%	53%-74%	60%-78%	62%-77	62%-77%	61%-77%
3rd set	J	56%	68%	72%	72%	71%	70%
	σ_J	31%	26%	22%	21%	21%	21%
	CI_J	44%-67%	58%-77%	64%-81%	65%-80	64%-79%	62%-78%
4th set	J	53%	62%	70%	70%	69%	67%
	σ_J	30%	27%	24%	21%	22%	22%
	CI_J	42%-65%	52%-73%	61%-79%	62%-78	61%-77%	59%-75%
5th set	J	60%	70%	75%	76%	75%	74%
	σ_J	29%%	24%	21%	19%	19%	19%
	CI_J	49%-71%	61%-79%	67%-83%	69%-83	68%-82%	67%-81%
6th set	J	52%	64%	70%	71%	70%	69%
	σ_J	30%	27%	24%	20%	21%	20%
	CI_J	41%-64%	54%-74%	61%-79%	64%-79	62%-78%	62%-77 %
7th set	J	56%	67%	72%	74%	72%	71%
	σ_J	29%	25%	22%	18%	18%	18%
	CI_J	45%-67%	58%-76%	64%-80%	67%-80	65%-79%	64%-78%
8th set	J	54%	64%	70%	74%	72%	70%
	σ_J	31%	28%	24%	18%	21%	23%
	CI_J	42%-65%	53%-74%	61%-79%	67%-81%	65%-80%	62%-79%
9th set	J	56%	65%	71%	71%	70%	69%
	σ_J	31%	28%	24%	20%	21%	22%
	CI_J	45%-68%	54%-75%	62%-80%	63%-78	62%-77%	61%-77%
10th set	J	55%	66%	70%	72%	71%	71%
	σ_J	31%	27%	24%	20%	21%	23%
	CI_J	44%-67%	56%-76%	61%-79%	64%-79	64%-79%	63%-80%

Table 4.14: Results for breast ultrasound segmentation: For each training set (each training set is repeated five times), the results produced with NC5 ($N_S = 5$), NC8 ($N_S = 8$), and NC10 ($N_S = 10$) are compared with the best, average, and worst results obtained with MSC-EFIS for the N-cut technique.

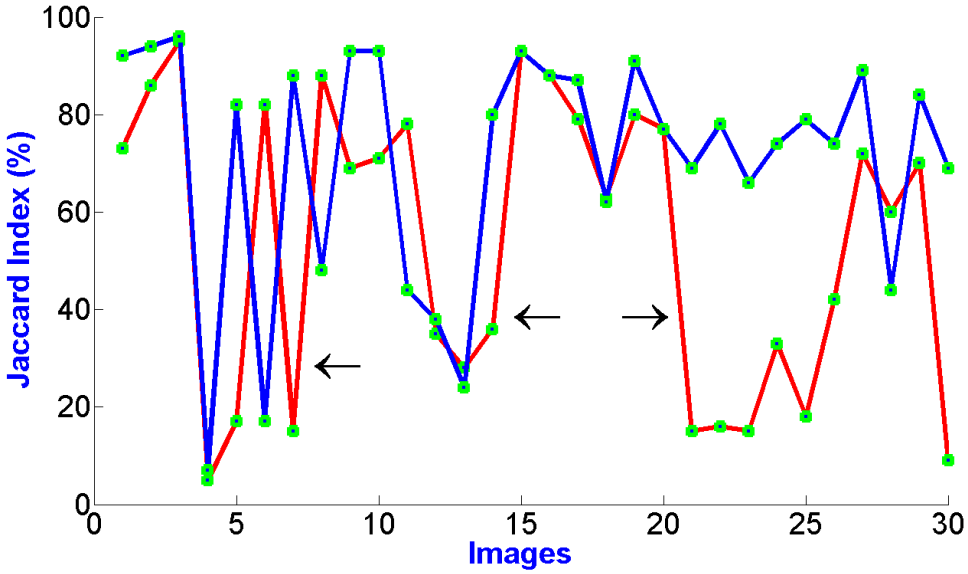

Figure 4.12: Comparison of the Jaccard accuracy obtained with MSC-EFIS (blue) and with NC5 (red); arrows point to significant gaps.

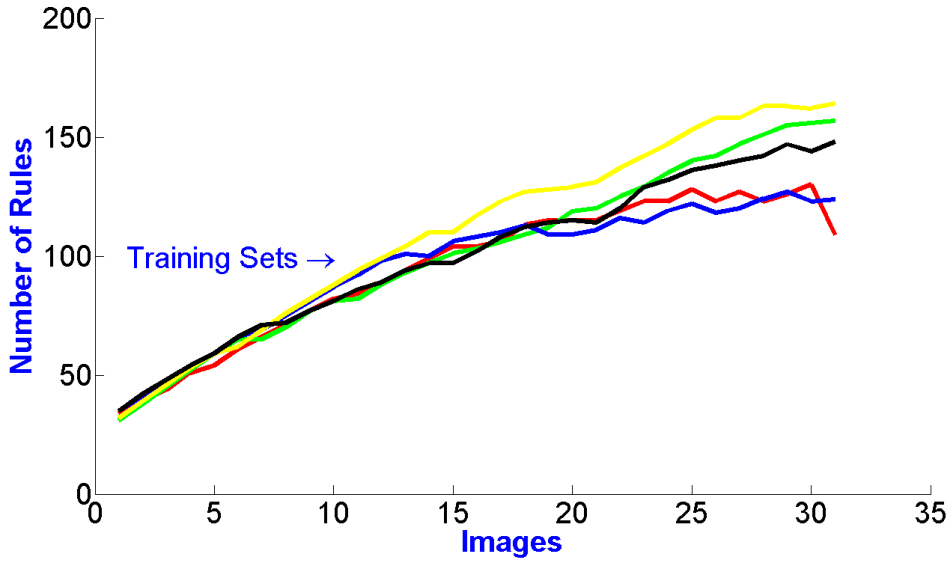

Figure 4.13: Rule evolution for MSC-EFIS for the first five training sets (first run of each) for breast images: A continual increases in the number of rules is exhibited during the evolution process. Each curve shows the number of rules for a separate run.

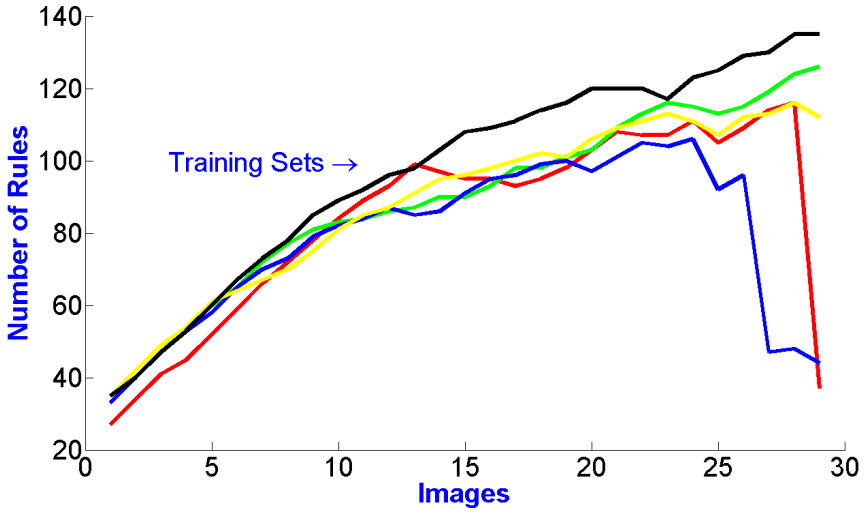

Figure 4.14: Rule evolution for MSC-EFIS for the first five training sets (first run of each) for bladder images: A continual increases in the number of rules is exhibited during the evolution process, except two training set, the number of rules converges at the end of the images. Each curve shows the number of rules for a separate run.

Chapter 5

Conclusions and Future Work

This chapter presents a summary of conclusions that can be drawn based on the research presented in this thesis and includes suggestions for future studies.

5.1 Conclusions

Most image segmentation techniques involve multiple parameters that must be tuned in order to achieve maximum segmentation accuracy. In most cases, the threshold generated by thresholding algorithms, the similarity threshold in region growing techniques, the scale in statistical region merging methods, and a variety of parameters in graph cut must all be adjusted in order to obtain maximum segmentation accuracy. Intelligent segmentation as a means of generating a threshold for unseen images would therefore seem to be a valuable approach. More importantly, the fine-tuning of any segmentation technique for medical image analysis should always be coupled with user feedback because the expert user is the ultimate authority with respect to evaluating the quality of the segmentation.

For this research, an evolving fuzzy image segmentation system has been designed for segmenting medical images through the online updating of the fuzzy rules. This system is a general approach that can be used to adjust the segmentation parameters of any segmentation technique in order to increase its accuracy. Initial fuzzy rules are established using a set of features extracted from training images, and their optimum parameters are determined as the target of the

rules. Based on expert feedback, these rules are then updated online during the processing of each image. The evolving nature of this system makes this approach appealing for applications that can benefit from or that depend on user feedback; medical image analysis is such a field, in which clinical experts usually approve computer-generated results. Processing similar cases leads to continual improvement in the results. This evolving process seems to be efficient for enhancing the accuracy of the segmentation for a variety of images. Three approaches have been proposed: evolving fuzzy image segmentation (EFIS), self configuring EFIS (SC-EFIS), and multi-parametric SC-EFIS (MSC-EFIS).

EFIS is a segmentation scheme that relies on user feedback in order to improve the quality of segmentation. Its evolving nature makes this approach attractive for applications that incorporate high-quality user feedback, such as the analysis of medical images. Two versions of EFIS have been proposed: one for single-parameter output and one for switch and fusion. Both methods showed promising results, with further processing of similar cases leading to continual and measurable improvements. The capability of EFIS was demonstrated for three distinct segmentation algorithms: Global thresholding, statistical region merging, and region growing. EFIS can be leveraged to incorporate observer-driven adjustments and to achieve an overall boost in segmentation accuracy. Further improvements were demonstrated with EFIS, in which is added a component that enables the fusion of the results from variety of methods. The results of EFIS for global thresholding were compared with those from five other methods, including one local method: the evolving EFIS rules consistently generated better results than the non-evolving methods.

EFIS entails some limitations, such as the large number of parameters that must be selected prior to the running of the algorithm and the lack of a feature selection component. These drawbacks restrict the use of EFIS to specific types of images. An improved version of EFIS, called self-configuring EFIS (SC-EFIS) has been proposed. SC-EFIS is a general image segmentation approach that does not require the advance assignment of parameters. SC-EFIS operates with the data available and extracts all the required parameters from those data regardless of the data type. SC-EFIS also has a multi-stage feature selection algorithm that combines many popular unsupervised feature selection approaches. A comparison of the SC-EFIS results with those obtained with EFIS demonstrates the efficiency of the proposed SC-EFIS approach, which also offers a higher level of automation.

An additional new multi-parametric SC-EFIS approach has also been proposed as a means of adjusting the parameters of the N-cut segmentation technique for different images based on image features.

The positive experimental results offer empirical evidence of and indicate confidence in the significant benefits to be derived from the algorithms developed for this thesis, especially in the field of medical image segmentation.

5.2 Future Works

Further research using larger datasets of real-world images is required to verify the benefits of rule evolution and to explore different methods of integrating user feedback, including online web technology. Additional investigation is also necessary with respect to exploring the online estimation of desired (best) output under multi-parametric settings. Examples of areas that could be explored as a means of estimating the best parameters for multi-parametric settings are genetic algorithms, differential evolution, and even independent evolving fuzzy systems. Another possible avenue of inquiry is the generation of separate rule bases for each parameter, which could represent a more practical approach in some cases, such as those in which a correlation between parameters is non-existent or negligible.

Bibliography

[1] R. Gonzalez and R. Woods, Digital Image Processing. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2nd ed., 2001.

[2] L. Shapiro and G. Stockman, Computer Vision. NJ: Prentice-Hall: Upper Saddle River, 2nd ed., 2001.

[3] W. K. Pratt, Digital Image Processing: PIKS Inside. New York, NY, USA: John Wiley & Sons, Inc., 3rd ed., 2001.

[4] D. Pham, C. Xu, and J. Prince, "Current methods in medical image segmentation," Annual Review of Biomedical Engineering, vol. 2, no. 1, pp. 315–337, 2000.

[5] M. Ogiela and R. Tadeusiewicz, Modern Computational Intelligence Methods for the Interpretation of Medical Images. New York, USA: Springer, 2008.

[6] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," Journal of Electronic Imaging, vol. 13, no. 1, pp. 146–165, 2004.

[7] J. Kim and H. Kim, "Multiresolution-based watersheds for efficient image segmentation," Pattern Recognition Letters, vol. 24, no. 1, pp. 473–488, 2003.

[8] C. Jung, "Combining wavelets and watersheds for robust multiscale image segmentation," Image and Vision Computing, vol. 25, no. 1, pp. 24–33, 2007.

[9] A. Cristoforetti, L. Faes, F. Ravelli, M. Centonze, M. Del Greco, R. Antolini, and G. Nollo, "Isolation of the left atrial surface from cardiac multi-detector CT images based on marker

controlled watershed segmentation," Medical Engineering & Physics, vol. 30, no. 1, pp. 48–58, 2008.

[10] B. Liu, H. Cheng, J. Huang, J. Tian, X. Tang, and J. Liu, "Probability density difference-based active contour for ultrasound image segmentation," Pattern Recognition, vol. 43, no. 6, pp. 2028–2042, 2010.

[11] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," International Journal of Computer Vision, vol. 59, no. 2, pp. 167–181, 2004.

[12] Hu, Grossberg, and Mageras, "Survey of recent volumetric medical image segmentation techniques," Biomedical Engineering, vol. 2, no. 1, pp. 315–337, 2009.

[13] G. Karmakar and L. Dooley, "A generic fuzzy rule based image segmentation algorithm," Pattern Recognition Letters, vol. 23, no. 10, pp. 1215–1227, 2002.

[14] W. Tao, J. Tian, and J. Liu, "Image segmentation by three-level thresholding based on maximum fuzzy entropy and genetic algorithm," Pattern Recognition Letters, vol. 24, no. 16, pp. 3069–3078, 2003.

[15] L. Cinque, G. Foresti, and L. Lombardi, "A clustering fuzzy approach for image segmentation," Pattern Recognition, vol. 37, no. 9, pp. 1797–1807, 2004.

[16] C. Chang and P. Chung, "Medical image segmentation using a contextual-constraint-based hopfield neural cube," Image and Vision Computing, vol. 19, no. 9, pp. 669–678, 2001.

[17] V. de Albuquerque, A. de Alexandria, P. Cortez, and J. Tavares, "Evaluation of multilayer perceptron and self-organizing map neural network topologies applied on microstructure segmentation from metallographic images," NDT & E International, vol. 42, no. 7, pp. 644–651, 2009.

[18] M. Kurnaz, Z. Dokur, and T. Ölmez, "An incremental neural network for tissue segmentation in ultrasound images," Computer Methods and Programs in Biomedicine, vol. 85, no. 3, pp. 187–195, 2007.

[19] B. Li, C. Chui, S. Chang, and S. Ong, "Integrating spatial fuzzy clustering with level set methods for automated medical image segmentation," Computers in Biology and Medicine, vol. 41, no. 1, pp. 1–10, 2011.

[20] P. Angelov and R. Buswell, "Evolving rule-based models: A tool for intelligent adaptation," in Proceeding of Joint 9th IFSA World Congress and 20th NAFIPS International Conference, vol. 2, pp. 1062–1067, IEEE, 2001.

[21] P. Angelov and R. Buswell, "Identification of evolving fuzzy rule-based models," IEEE Transactions on Fuzzy Systems, vol. 10, no. 5, pp. 667–677, 2002.

[22] N. Otsu, "A threshold selection method from gray-level histograms," IEEE Transactions on Systems, Man and Cybernetics, vol. 48, no. 9, pp. 62–66, 1979.

[23] A. Rosenfeld and P. De La Torre, "Histogram concavity analysis as an aid in threshold selection," IEEE Transactions on Systems, Man, and Cybernetics, vol. 13, no. 2, pp. 231–235, 1983.

[24] J. Kittler and J. Illingworth, "Minimum error thresholding," Pattern Recognition, vol. 19, no. 1, pp. 41–47, 1986.

[25] J. Kapur, P. Sahoo, and A. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," Computer Vision, Graphics, and Image Processing, vol. 29, no. 3, pp. 273–285, 1985.

[26] A. Shanbhag, "Utilization of information measure as a means of image thresholding," Graphical Models and Image Processing, vol. 56, no. 5, pp. 414–419, 1994.

[27] C. Murthy and S. Pal, "Fuzzy thresholding: Mathematical framework, bound functions and weighted moving average technique," Pattern Recognition Letters, vol. 11, no. 3, pp. 197–206, 1990.

[28] N. Pal and S. Pal, "Entropic thresholding," Signal Processing, vol. 16, no. 2, pp. 97–108, 1989.

[29] W. Niblack, An Introduction to Digital Image Processing. Englewood Cliffs, New Jersey: Prentice-Hall International Inc., 2nd ed., 1986.

[30] H. Tizhoosh, "Image thresholding using type II fuzzy sets," Pattern Recognition, vol. 38, no. 12, pp. 2363–2372, 2005.

[31] D. Liu, Z. Jiang, and H. Feng, "A novel fuzzy classification entropy approach to image thresholding," Pattern Recognition Letters, vol. 27, no. 16, pp. 1968–1975, 2006.

[32] Y. Bazi, L. Bruzzone, and F. Melgani, "Image thresholding based on the EM algorithm and the generalized gaussian distribution," Pattern Recognition, vol. 40, no. 2, pp. 619–634, 2007.

[33] A. Nakib, H. Oulhadj, and P. Siarry, "Image histogram thresholding based on multiobjective optimization," Signal Processing, vol. 87, no. 11, pp. 2516–2534, 2007.

[34] K. Hammouche, M. Diaf, and P. Siarry, "A multilevel automatic thresholding method based on a genetic algorithm for a fast image segmentation," Computer Vision and Image Understanding, vol. 109, no. 2, pp. 163–175, 2008.

[35] S. Wang, F. Chung, and F. Xiong, "A novel image thresholding method based on Parzen window estimate," Pattern Recognition, vol. 41, no. 1, pp. 117–129, 2008.

[36] A. Nakib, H. Oulhadj, and P. Siarry, "Fractional differentiation and non-pareto multiobjective optimization for image thresholding," Engineering Applications of Artificial Intelligence, vol. 22, no. 2, pp. 236–249, 2009.

[37] Z. Li, C. Liu, G. Liu, Y. Cheng, X. Yang, and C. Zhao, "A novel statistical image thresholding method," International Journal of Electronics and Communications, vol. 64, no. 12, pp. 1137–1147, 2010.

[38] J. Xue and D. Titterington, "Median-based image thresholding," Image and Vision Computing, vol. 29, no. 9, pp. 631–637, 2011.

[39] A. Othman and H. Tizhoosh, "Evolving fuzzy image segmentation," in Proceeding of IEEE International Conference on Fuzzy Systems, pp. 1603–1609, IEEE, 2011.

[40] A. Othman and H. Tizhoosh, "Neural image thresholding with SIFT-controlled gabor features," in Proceedings of the International Joint Conference on Neural Networks, pp. 2106 –2112, 2011.

[41] A. Othman and H. Tizhoosh, "Segmentation of breast ultrasound images using neural networks," Engineering Applications of Neural Networks, vol. 363, pp. 260–269, 2011.

[42] Z. Lin, J. Jin, and H. Talbot, "Unseeded region growing for 3D image segmentation," in Selected Papers from the Pan-Sydney Workshop on Visualisation, vol. 2, pp. 31–37, 2000.

[43] R. Stewart, I. Fermin, and M. Opper, "Region growing with pulse-coupled neural networks: an alternative to seeded region growing," IEEE Transactions on Neural Networks, vol. 13, no. 6, pp. 1557–1562, 2002.

[44] Y. Lu, J. Miao, L. Duan, Y. Qiao, and R. Jia, "A new approach to image segmentation based on simplified region growing PCNN," Applied Mathematics and Computation, vol. 205, no. 2, pp. 807–814, 2008.

[45] A. Malek, W. Rahman, A. Ibrahim, R. Mahmud, S. Yasiran, and A. Jumaat, "Region and boundary segmentation of microcalcifications using seed-based region growing and mathematical morphology," Procedia-Social and Behavioral Sciences, vol. 8, pp. 634–639, 2010.

[46] J. Park and C. Lee, "Skull stripping based on region growing for magnetic resonance brain images," NeuroImage, vol. 47, no. 4, pp. 1394–1407, 2009.

[47] R. Nock and F. Nielsen, "Statistical region merging," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 11, pp. 1452–1458, 2004.

[48] F. Kurugollu, B. Sankur, and A. Harmanci, "Image segmentation by relaxation using constraint satisfaction neural network," Image and Vision Computing, vol. 20, no. 7, pp. 483–497, 2002.

[49] J. Glass, W. Reddick, O. Goloubeva, V. Yo, and R. Steen, "Hybrid artificial neural network segmentation of precise and accurate inversion recovery (PAIR) images from normal human brain," Magnetic Resonance Imaging, vol. 18, no. 10, pp. 1245–1253, 2000.

[50] J. Núñez and J. Llacer, "Astronomical image segmentation by self-organizing neural networks and wavelets," Neural Networks, vol. 16, no. 3, pp. 411–417, 2003.

[51] J. Lee, C. Steele, and T. Chau, "Swallow segmentation with artificial neural networks and multi-sensor fusion," Medical Engineering & physics, vol. 31, no. 9, pp. 1049–1055, 2009.

[52] M. Kurnaz, Z. Dokur, and T. Ölmez, "Segmentation of remote-sensing images by incremental neural network," Pattern Recognition Letters, vol. 26, no. 8, pp. 1096–1104, 2005.

[53] Z. Dokur, "A unified framework for image compression and segmentation by using an incremental neural network," Expert Systems with Applications, vol. 34, no. 1, pp. 611–619, 2008.

[54] Z. Iscan, A. Yüksel, Z. Dokur, M. Korürek, and T. Ölmez, "Medical image segmentation with transform and moment based features and incremental supervised neural network," Digital Signal Processing, vol. 19, no. 5, pp. 890–901, 2009.

[55] J. Fu, C. Chen, J. Chai, S. Wong, and I. Li, "Image segmentation by EM-based adaptive pulse coupled neural networks in brain magnetic resonance imaging," Computerized Medical Imaging and Graphics, vol. 34, no. 4, pp. 308–320, 2010.

[56] L. He, Z. Peng, B. Everding, X. Wang, C. Han, K. Weiss, and W. Wee, "A comparative study of deformable contour methods on medical image segmentation," Image and Vision Computing, vol. 26, no. 2, pp. 141–163, 2008.

[57] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," International Journal of Computer Vision, vol. 22, no. 1, pp. 61–79, 1997.

[58] T. Chan and L. Vese, "Active contours without edges," IEEE Transactions on Image Processing, vol. 10, no. 2, pp. 266–277, 2001.

[59] C. Li, C. Xu, C. Gui, and M. Fox, "Level set evolution without re-initialization: A new variational formulation," in Proceeding of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 430–436, IEEE, 2005.

[60] C. Li, C. Xu, C. Gui, and M. Fox, "Distance regularized level set evolution and its application to image segmentation," IEEE Transactions on Image Processing, vol. 19, no. 12, pp. 3243–3254, 2010.

[61] S. Lankton and A. Tannenbaum, "Localizing region-based active contours," IEEE Transactions on Image Processing, vol. 17, no. 11, pp. 2029–2039, 2008.

[62] Y. S. and W. Karl, "A real-time algorithm for the approximation of level-set-based curve evolution," IEEE Transactions on Image Processing, vol. 17, no. 5, pp. 645–656, 2008.

[63] O. Bernard, D. Friboulet, P. Thevenaz, and M. Unser, "Variational B-Spline level-set: A linear filtering approach for fast deformable model evolution," IEEE Transactions on Image Processing, vol. 18, no. 6, pp. 1179 –1191, 2009.

[64] Y. Zhang, B. Matuszewski, L. Shark, and C. Moore, "Medical image segmentation using new hybrid level-set method," in Proceeding of Fifth International Conference on Biomedical Visualization, pp. 71–76, IEEE, 2008.

[65] Y. Chen, "A level set method based on the Bayesian risk for medical image segmentation," Pattern Recognition, vol. 43, no. 11, pp. 3699–3711, 2010.

[66] A. Achuthan, M. Rajeswari, D. Ramachandram, M. Aziz, and I. Shuaib, "Wavelet energy-guided level set-based active contour: A segmentation method to segment highly similar regions," Computers in Biology and Medicine, vol. 40, no. 7, pp. 608–620, 2010.

[67] B. Li, C. Chui, S. Ong, and S. Chang, "Integrating FCM and level sets for liver tumor segmentation," in Proceeding of 13th International Conference on Biomedical Engineering, vol. 23, pp. 202–205, Springer, 2009.

[68] Y. Weiss, "Segmentation using eigenvectors: A unifying view," in Proceeding of the 7th IEEE International Conference on Computer Vision, vol. 2, pp. 975–982, 1999.

[69] J. Shi and J. Malik, "Normalized cuts and image segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 888–905, 2000.

[70] Y. Boykov and M. Jolly, "Interactive organ segmentation using graph cuts," in Proceeding of Medical Image Computing and Computer-Assisted Intervention Conference, vol. 1935, pp. 147–175, Springer, 2000.

[71] Y. Boykov and M. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in ND images," in Proceeding of the Eighth IEEE International Conference on Computer Vision, vol. 1, pp. 105–112, 2001.

[72] 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, vol. 26, no. 9, pp. 1124–1137, 2004.

[73] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient ND image segmentation," International Journal of Computer Vision, vol. 70, no. 2, pp. 109–131, 2006.

[74] Y. Li, J. Sun, C. Tang, and H. Shum, "Lazy snapping," ACM Transactions on Graphics, vol. 23, no. 3, pp. 303–308, 2004.

[75] B. Peng, L. Zhang, D. Zhang, and J. Yang, "Image segmentation by iterated region merging with localized graph cuts," Pattern Recognition, vol. 44, no. 10, pp. 2527–2538, 2011.

[76] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," ACM Transactions on Graphics, vol. 23, no. 3, pp. 309–314, 2004.

[77] D. Freedman and T. Zhang, "Interactive graph cut based segmentation with shape priors," in Proceeding of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 755–762, 2005.

[78] L. Gorelick, A. Delong, O. Veksler, and Y. Boykov, "Recursive MDL via graph cuts: Application to segmentation," Proceeding of the International Conference on Computer Vision, pp. 890–897, 2011.

[79] N. Lermé, L. Létocart, and F. Malgouyres, "Reduced graphs for min-cut/max-flow approaches in image segmentation," Electronic Notes in Discrete Mathematics, vol. 37, no. 10, pp. 63–68, 2011.

[80] W. Tao, F. Chang, L. Liu, H. Jin, and T. Wang, "Interactively multiphase image segmentation based on variational formulation and graph cuts," Pattern Recognition, vol. 43, no. 10, pp. 3208–3218, 2010.

[81] S. Ababneh, J. Prescott, and M. Gurcan, "Automatic graph-cut based segmentation of bones from knee magnetic resonance images for osteoarthritis research," Medical Image Analysis, vol. 15, no. 4, pp. 438–448, 2011.

[82] W. Tao and X. Tai, "Multiple piecewise constant with geodesic active contours (MPC-GAC) framework for interactive image segmentation using graph cuts optimization," Image and Vision Computing, vol. 29, no. 8, pp. 499–508, 2011.

[83] Q. Huang, S. Lee, L. Liu, M. Lu, L. Jin, and A. Li, "A robust graph-based segmentation method for breast tumors in ultrasound images," Ultrasonics, vol. 52, no. 2, pp. 266–275, 2012.

[84] M. Letteboer, O. Olsen, E. Dam, P. Willems, M. Viergever, and W. Niessen, "Segmentation of tumors in magnetic resonance brain images using an interactive multiscale watershed algorithm," Academic Radiology, vol. 11, no. 10, pp. 1125–1138, 2004.

[85] W. Kuo, C. Lin, and Y. Sun, "Brain MR images segmentation using statistical ratio: Mapping between watershed and competitive hopfield clustering network algorithms," Computer Methods and Programs in Biomedicine, vol. 91, no. 3, pp. 191–198, 2008.

[86] G. Hamarneh and X. Li, "Watershed segmentation using prior shape and appearance knowledge," Image and Vision Computing, vol. 27, no. 1, pp. 59–68, 2009.

[87] Wei-Yen and Hsu, "Improved watershed transform for tumor segmentation: Application to mammogram image compression," Expert Systems with Applications, vol. 39, no. 4, pp. 3950–3955, 2012.

[88] D. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110, 2004.

[89] M. Bicego, A. Lagorio, E. Grosso, and M. Tistarelli, "On the use of SIFT features for face authentication," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop, pp. 35–41, 2006.

[90] J. Yun and R. Park, "Self-calibration with two views using the scale-invariant feature transform," Advances in Visual Computing, pp. 589–598, 2006.

[91] Y. Meng and B. Tiddeman, "Implementing the scale invariant feature transform (sift) method," Department of Computer Science University of St. Andrews, 2008.

[92] U. Sinha, "SIFT: Scale invariant feature transform." http://www.aishack.in/2010/05/sift-scale-invariant-feature-transform, Nov. 2012.

[93] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms." http://www.vlfeat.org/, Nov. 2012.

[94] A. Vedaldi, "An implementation of sift detector and descriptor," University of California at Los Angeles, vol. 7, 2006.

[95] W. Cai, S. Chen, and D. Zhang, "Fast and robust fuzzy C-means clustering algorithms incorporating local information for image segmentation," Pattern Recognition, vol. 40, no. 3, pp. 825–838, 2007.

[96] Z. Ji, Q. Sun, Y. Xia, Q. Chen, D. Xia, and D. Feng, "Generalized rough fuzzy c-means algorithm for brain MR image segmentation," Computer Methods and Programs in Biomedicine, vol. 108, no. 2, pp. 644 – 655, 2012.

[97] Z. Ji, Y. Xia, Q. Chen, Q. Sun, D. Xia, and D. Feng, "Fuzzy c-means clustering with weighted image patch for image segmentation," Applied Soft Computing, vol. 12, no. 6, pp. 1659 – 1667, 2012.

[98] J. Feng, L. Jiao, X. Zhang, M. Gong, and T. Sun, "Robust non-local fuzzy c-means algorithm with edge preservation for SAR image segmentation," Signal Processing, vol. 93, no. 2, pp. 487 – 499, 2013.

[99] F. Zhao, L. Jiao, and H. Liu, "Kernel generalized fuzzy c-means clustering with spatial information for image segmentation," Digital Signal Processing, vol. 23, no. 1, pp. 184 – 199, 2013.

[100] L. Coletta, L. Vendramin, E. Hruschka, R. Campello, and W. Pedrycz, "Collaborative fuzzy clustering algorithms: Some refinements and design guidelines," IEEE Transactions on Fuzzy Systems, vol. 20, no. 3, pp. 444–462, 2012.

[101] G. Krell, H. Tizhoosh, T. Lilienblum, C. Moore, and B. Michaelis, "Enhancement and associative restoration of electronic portal images in radiotherapy," International journal of medical informatics, vol. 49, no. 2, pp. 157–171, 1998.

[102] S. Pal and A. Rosenfeld, "Image enhancement and thresholding by optimization of fuzzy compactness," Pattern Recognition Letters, vol. 7, no. 2, pp. 77–86, 1988.

[103] S. Pal and A. Ghosh, "Fuzzy geometry in image analysis," Fuzzy Sets and Systems, vol. 48, no. 1, pp. 23–40, 1992.

[104] N. Pal, D. Bhandarai, and D. Majumder, "Fuzzy divergence, probability measure of fuzzy events and image thresholding," Pattern Recognition, vol. 13, no. 12, pp. 857–867, 1992.

[105] Q. Wang, Z. Chi, and R. Zhao, "Image thresholding by maximizing the index of nonfuzziness of the 2-d grayscale histogram," Computer Vision and Image Understanding, vol. 85, no. 2, pp. 100–116, 2002.

[106] R. Rajesh, N. Senthilkumaran, J. Satheeshkumar, B. Shanmuga, C. Thilagavathy, and K. Priya, "On the type-1 and type-2 fuzziness measures for thresholding MRI brain images," in IEEE International Conference on Fuzzy Systems, pp. 992–995, 2011.

[107] D. Neog, M. Raza, and F. Rhee, "An interval type 2 fuzzy approach to multilevel image segmentation," in IEEE International Conference on Fuzzy Systems, pp. 1164–1170, 2011.

[108] D. Zhang and S. Chen, "A novel kernelized fuzzy C-means algorithm with application in medical image segmentation," Artificial Intelligence in Medicine, vol. 32, no. 1, pp. 37–50, 2004.

[109] K. Chuang, H. Tzeng, S. Chen, J. Wu, and T. Chen, "Fuzzy C-means clustering with spatial information for image segmentation," Computerized Medical Imaging and Graphics, vol. 30, no. 1, pp. 9–15, 2006.

[110] Z. Yang, F. Chung, and W. Shitong, "Robust fuzzy clustering-based image segmentation," Applied Soft Computing, vol. 9, no. 1, pp. 80–84, 2009.

[111] Y. Zhuge, J. Udupa, and P. Saha, "Vectorial scale-based fuzzy-connected image segmentation," Computer Vision and Image Understanding, vol. 101, no. 3, pp. 177–193, 2006.

[112] F. Zhao, L. Jiao, H. Liu, and X. Gao, "A novel fuzzy clustering algorithm with non local adaptive spatial constraint for image segmentation," Signal Processing, vol. 91, no. 4, pp. 988–999, 2010.

[113] W. Dou, S. Ruan, Y. Chen, D. Bloyet, and J. Constans, "A framework of fuzzy information fusion for the segmentation of brain tumor tissues on MR images," Image and Vision Computing, vol. 25, no. 2, pp. 164–171, 2007.

[114] Y. Hata and S. Kobashi, "Fuzzy segmentation of endorrhachis in magnetic resonance images and its fuzzy maximum intensity projection," Applied Soft Computing, vol. 9, no. 3, pp. 1156–1169, 2009.

[115] J. Foo, G. Miyano, T. Lobe, and E. Winer, "Three-dimensional segmentation of tumors from CT image data using an adaptive fuzzy system," Computers in Biology and Medicine, vol. 39, no. 10, pp. 869–878, 2009.

[116] H. Khotanlou, O. Colliot, J. Atif, and I. Bloch, "3D brain tumor segmentation in MRI using fuzzy classification, symmetry analysis and spatially constrained deformable models," Fuzzy Sets and Systems, vol. 160, no. 10, pp. 1457–1473, 2009.

[117] V. Rivas, J. Merelo, I. Rojas, G. Romero, P. Castillo, and J. Carpio, "Evolving two-dimensional fuzzy systems," Fuzzy Sets and Systems, vol. 138, no. 2, pp. 381–398, 2003.

[118] A. Borji and M. Hamidi, "Evolving a fuzzy rule-base for image segmentation," International Journal of Intelligent Systems and Technologies, vol. 28, pp. 178–183, 2007.

[119] S. Yuhui, R. Eberhart, and C. Yaobin, "Implementation of evolutionary fuzzy systems," IEEE Transactions on Fuzzy Systems, vol. 7, no. 2, pp. 109–119, 1999.

[120] R. Lai, C. Fan, W. Huang, and P. Chang, "Evolving and clustering fuzzy decision tree for financial time series data forecasting," Expert Systems with Applications, vol. 36, no. 2, pp. 3761–3773, 2009.

[121] H. Mallinson and P. Bentley, "Evolving fuzzy rules for pattern classification," in Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation - CIMCA99, vol. 55, pp. 184–191, 1999.

[122] A. Pipe and B. Carse, "Michigan and pittsburgh fuzzy classifier systems for learning mobile robot control rules: an experimental comparison," in 14th International Conference of Artificial Intelligence Research Society FLAIRS, pp. 493–497, 2001.

[123] S. Mehta, S. Chaudhury, A. Bhattacharyya, and A. Jena, "Tissue classification in magnetic resonance images through the hybrid approach of michigan and pittsburg genetic algorithm," Applied Soft Computing, vol. 11, no. 4, pp. 3476–3484, 2011.

[124] P. Chang, C. Liu, and Y. Wang, "A hybrid model by clustering and evolving fuzzy rules for sales decision supports in printed circuit board industry," Decision Support Systems, vol. 42, no. 3, pp. 1254–1269, 2006.

[125] R. Linden and A. Bhaya, "Evolving fuzzy rules to model gene expression," Biosystems, vol. 88, no. 1, pp. 76–91, 2007.

[126] N. Kasabov, "On-line learning, reasoning, rule extraction and aggregation in locally optimized evolving fuzzy neural networks," Neurocomputing, vol. 41, no. 1, pp. 25–45, 2001.

[127] N. Kasabov, "Evolving fuzzy neural networks for supervised/unsupervised online knowledge-based learning," IEEE Transactions on Systems, Man, and Cybernetics, vol. 31, no. 6, pp. 902–918, 2001.

[128] C. Lin and Y. Xu, "A self-adaptive neural fuzzy network with group-based symbiotic evolution and its prediction applications," Fuzzy Sets and Systems, vol. 157, no. 8, pp. 1036–1056, 2006.

[129] C. Juang, Y. Lin, and C. Tu, "A recurrent self-evolving fuzzy neural network with local feedbacks and its application to dynamic system processing," Fuzzy Sets and Systems, vol. 161, no. 19, pp. 2552–2568, 2010.

[130] H. Rong, N. Sundararajan, G. Huang, and P. Saratchandran, "Sequential Adaptive Fuzzy Inference System (SAFIS) for nonlinear system identification and prediction," Fuzzy Sets and Systems, vol. 157, no. 9, pp. 1260–1275, 2006.

[131] M. Futschik, A. Reeve, and N. Kasabov, "Evolving connectionist systems for knowledge discovery from gene expression data of cancer tissue," Artificial Intelligence in Medicine, vol. 28, no. 2, pp. 165–189, 2003.

[132] P. Chang, Y. Wang, and C. Liu, "The development of a weighted evolving fuzzy neural network for PCB sales forecasting," Expert Systems with Applications, vol. 32, no. 1, pp. 86–96, 2007.

[133] P. Angelov, "A fuzzy controller with evolving structure," Information Sciences, vol. 161, no. 1, pp. 21–35, 2004.

[134] E. Lughofer, "On-line assurance of interpretability criteria in evolving fuzzy systems achievements, new concepts and open issues," Information Sciences, vol. 251, pp. 22 – 46, 2013.

[135] P. Angelov, E. Lughofer, and X. Zhou, "Evolving fuzzy classifiers using different model architectures," Fuzzy Sets and Systems, vol. 159, no. 23, pp. 3160–3182, 2008.

[136] P. Angelov and X. Zhou, "Evolving fuzzy-rule-based classifiers from data streams," IEEE Transactions on Fuzzy Systems, vol. 16, no. 6, pp. 1462–1475, 2008.

[137] D. Wang, X. Zeng, and J. Keane, "A simplified structure evolving method for mamdani fuzzy system identification and its application to high-dimensional problems," Information Sciences, vol. 220, pp. 110 – 123, 2013.

[138] A. Lemos, W. Caminhas, and F. Gomide, "Adaptive fault detection and diagnosis using an evolving fuzzy classifier," Information Sciences, vol. 220, pp. 64 – 85, 2013.

[139] J. de Barros and A. Dexter, "On-line identification of computationally undemanding evolving fuzzy models," Fuzzy Sets and Systems, vol. 158, no. 18, pp. 1997–2012, 2007.

[140] E. Lughofer, "On-line evolving image classifiers and their application to surface inspection," Image and Vision Computing, vol. 28, no. 7, pp. 1065–1079, 2010.

[141] Wikipedia, "Feature selection." http://en.wikipedia.org/wiki/Feature_selection, 2013.

[142] L. Molina, L. Belanche, and À. Nebot, "Feature selection algorithms: A survey and experimental evaluation," in IEEE International Conference on Data Mining ICDM., pp. 306–313, 2002.

[143] N. Sánchez-Maroño, A. Alonso-Betanzos, and M. Tombilla-Sanromán, "Filter methods for feature selection–a comparative study," in Proceedings of the 8th international conference on Intelligent data engineering and automated learning, pp. 178–187, Springer, 2007.

[144] T. Lal, O. Chapelle, J. Weston, and A. Elisseeff, "Embedded methods," in Feature Extraction, Studies in Fuzziness and Soft Computing Volume, vol. 207, pp. 137–165, Springer, 2006.

[145] J. Cadenas, M. Carmen, and R. Martínez, "Feature subset selection filter-wrapper based on low quality data," Expert Systems with Applications, vol. 40, no. 16, 2013.

[146] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," Bioinformatics, vol. 23, no. 19, pp. 2507–2517, 2007.

[147] J. Martínez and F. Pla, "Supervised feature selection by clustering using conditional mutual information-based distances," Pattern Recognition, vol. 43, no. 6, pp. 2068–2081, 2010.

[148] L. Song, A. Smola, A. Gretton, K. Borgwardt, and J. Bedo, "Supervised feature selection via dependence estimation," in Proceedings of the 24th international conference on Machine learning, pp. 823–830, ACM, 2007.

[149] Z. Zhao and H. Liu, "Semi-supervised feature selection via spectral analysis," in Proceedings of the 7th SIAM International Conference on Data Mining SDM, pp. 1151–1158, 2007.

[150] M. Kalakech, P. Biela, L. Macaire, and D. Hamad, "Constraint scores for semi-supervised feature selection: A comparative study," Pattern Recognition Letters, vol. 32, no. 5, pp. 656–665, 2011.

[151] R. Cai, Z. Zhang, and Z. Hao, "Bassum: A bayesian semi-supervised method for classification feature selection," Pattern Recognition, vol. 44, no. 4, pp. 811–820, 2011.

[152] F. Bellal, H. Elghazel, and A. Aussem, "A semi supervised feature ranking method with ensemble learning," Pattern Recognition Letters, vol. 33, no. 10, pp. 1426–1433, 2012.

[153] G. Doquire and M. Verleysen, "A graph laplacian based approach to semi-supervised feature selection for regression problems," Neurocomputing, vol. 121, pp. 5 – 13, 2013.

[154] P. Mitra, C. Murthy, and S. Pal, "Unsupervised feature selection using feature similarity," IEEE transactions on pattern analysis and machine intelligence, vol. 24, no. 3, pp. 301–312, 2002.

[155] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," Advances in Neural Information Processing Systems, vol. 18, pp. 507–514, 2006.

[156] Z. Zhao and H. Liu, "Spectral feature selection for supervised and unsupervised learning," in Proceedings of the 24th international conference on Machine learning, pp. 1151–1157, ACM, 2007.

[157] D. Cai, C. Zhang, and X. He, "Unsupervised feature selection for multi-cluster data," in Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 333–342, ACM, 2010.

[158] A. Farahat, A. Ghodsi, and M. Kamel, "Efficient greedy feature selection for unsupervised learning," Knowledge and Information Systems, vol. 35, no. 2, pp. 285–310, 2013.

[159] A. Othman, H. Tizhoosh, and F. Khalvati, "EFIS: Evolving fuzzy image segmentation," IEEE Transactions on Fuzzy Systems, no. 99, 2013.

[160] C. Li, J. Zhou, B. Fu, P. Kou, and J. Xiao, "T-s fuzzy model identification with a gravitational search-based hyperplane clustering algorithm," IEEE Transactions on Fuzzy Systems, vol. 20, no. 2, pp. 305–317, 2012.

[161] S. Warfield, "Simultaneous truth and performance level estimation (staple): an algorithm for the validation of image segmentation," IEEE Transactions on Medical Imaging, vol. 23, no. 7, pp. 903–921, 2004.

[162] S. Kirkpatrick, C. Gelatt, and C. Vecchi, "Optimization by simulated annealing," Science, vol. 220, pp. 671–680, 1983.

[163] D. Bertsimas and J. Tsitsiklis, "Simulated annealing," Statistical Science, pp. 10–15, 1993.

[164] P. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005.

[165] L. Huang and M. Wang, "Image thresholding by minimizing the measure of fuzziness," Pattern Recognition, vol. 28, no. 1, pp. 41–51, 1995.

Appendices

Appendix A

Scale-Invariant Feature Transform

The scale-invariant feature transform (SIFT) is an approach developed by Lowe [88] as a method of object recognition. It is based on the detection of a set of key points for an object and the subsequent calculation of a set of descriptors, or features, for these points. These features, invariant to image scale and rotation, can enable the differentiation of one object from a group of different objects. The process of detecting key points and calculating the descriptors consists of the following four stages:

- **Scale-space extrema detection:**
  In this stage, the points of interest called key points are detected. First, Gaussian filters at different scales are generated and then convolved with the image at every scale. The difference of Gaussians (DoGs) blurred images are determined, and the candidate key points are assigned as the maxima and minima of the DoGs $D\left(x, y, \sigma\right)$ at multiple scales. To create the scale space of an image, the convolution of a Gaussian filter $G(x, y, \sigma)$, with an input image $I(x, y)$, is determined and defined as a function $L(x, y, \sigma)$, where

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y), \tag{A.1}$$

  and (DoG) is given by

$$D\left(x, y, \sigma\right) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y). \tag{A.2}$$

The maxima and minima of the DoGs are calculated through a comparison of each point with its eight neighbours in the current image and nine neighbours in the scale above and below. The point is selected to be a key point if it is larger or smaller than all of these neighbours. This process is performed as follows:
Let the input image be $I(x, y)$

1. The scale space $L(x, y, \sigma)$ of the input image $I(x, y)$ is calculated using the following equation:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{A.3}$$

   where $G(x, y, \sigma)$ is the Gaussian filter with the initial value for $\sigma = 1.6$.

2. The σ is changed to $k * \sigma$ where $k = \sqrt{2}$.

3. Steps 1 and 2 are repeated five times in order to generate scales from $L1$ to $L5$ where $\sigma = k * \sigma$.

4. After the five different scales of the image $I(x, y)$ are generated, the different of Gaussian is calculated by subtracting every two adjacent scale spaces of $I(x, y)$ using the following equation:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma). \tag{A.4}$$

5. These calculations produce the first octave, which consists of five different scales $L1 \ldots L5$ of the input image $I(x, y)$ and four different difference of Gaussians images $D1 \ldots D4$ (Fig. A.1 [88]).

6. The input image is downsampled by a factor of 2. If $I(x, y)$ is 512×512, it will then be 256×256 (at the end of the algorithm, the keypoints is re-sampled to be located at the original image).

7. Steps from 1 to 5 are repeated in order to generate the second octave. The second octave is begun with the Gaussian image that was filtered with σ equal to double the initial σ, and this new $\sigma = 2 * 1.6$ is the initial sigma for this octave.

8. Steps 1 to 5 are repeated in order to generate the third octave. The third octave starts with the image filtered with $\sigma = 4 * 1.6$.

9. For every pixel in $D2(x, y, \sigma)$ and $D3(x, y, \sigma)$ in every octave, a pixel with coordinates (x_1, y_1) is selected to be a candidate key point if

   (a) Its intensity $D(x_1, y_1)$ is larger than that of its eight neighbours in a 3×3 window centered in $D(x_1, y_1)$ in its scale and larger than its nine neighbours in the scale above it and the scale below it (Fig A.2). The candidate is then compared with 26 pixels: eight in its scale and nine in the scale above, and nine in the scale below.

   (b) Its intensity $D(x_1, y_1)$ is smaller than that of its eight neighbours in a 3×3 window centered in $D(x_1, y_1)$ in its scale and smaller than its nine neighbours in the scale above it and its nine neighbours in the scale below it Fig (A.2 [88]).

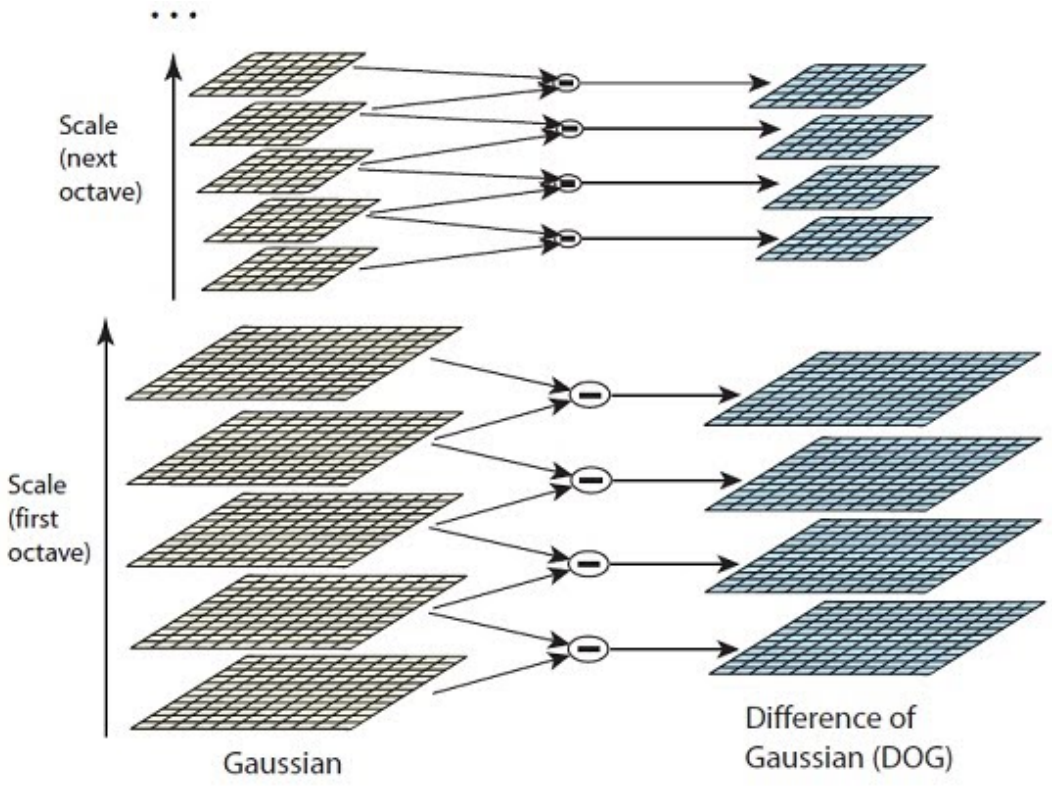

Figure A.1: The process of building the difference of Gaussians and the octaves

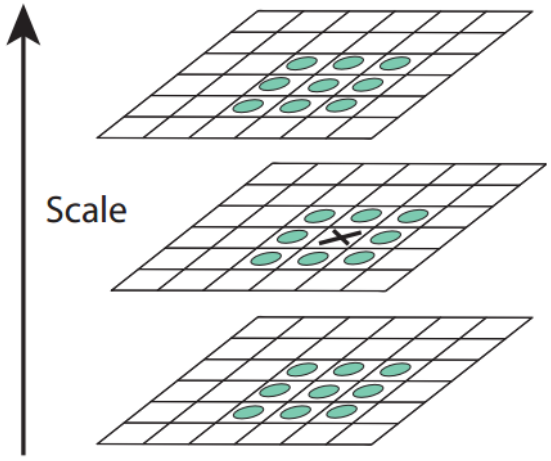

Figure A.2: The process of detecting key points

- **Key point localization:** A substantial number of candidate points are detected in the previous step; however, many of these points are useless due to either low contrast or to poor localization across the edges. Therefore, for each candidate point, a measure of stability is calculated so that points with low contrast or poor localization along the edges can be discarded. This process is performed as follows:

  1. The Taylor expansion up to quadratic terms is used to discard points with low contrast. The Taylor value is calculated from the following equation:

  $$D(x) = D + \frac{\partial(D^T)}{\partial(x)} + \frac{1}{2}x^T\frac{\partial^2(D)}{\partial(x^2)}x, \tag{A.5}$$

  where D and $\partial(D)$ are evaluated at the sample point, and $x = (x, y, \sigma)^T$ is the offset from this point.

  (a) The location of the extremum $\hat{x}$ is calculated using the following equation:

  $$\hat{x} = -\frac{\partial^2(D^{-1})}{\partial(x^2)}\frac{\partial(D)}{\partial(x)}. \tag{A.6}$$

  (b) The value of the Taylor expansion at the etxremum $\hat{x}$ is calculated using the

following equation:

$$D(\hat{x}) = D + \frac{1}{2}\frac{\partial(D^T)}{\partial(x)}\hat{x}. \tag{A.7}$$

(c) Any point with $|D(\hat{x})| < 0.03$ is discarded.

2. The principal curvature across the edges is used to discard points that are poorly localized across the edges.

(a) The principal curvature is calculated from the Hessian matrix from the difference of Gaussian image D using the following equation:

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & Dyy \end{bmatrix}. \tag{A.8}$$

(b) The trace of H and the determinant of H are then calculated:

$$Tr(H) = D_{xx} + D_{yy}, \tag{A.9}$$

$$Det(H) = D_{xx}Dyy - (D_{xy})^2. \tag{A.10}$$

(c) Any point that fails in the following inequality is discarded:

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r}, \tag{A.11}$$

where $r = 10$.

- **Orientation assignment:** Local image gradient directions are used as a means of assigning one or more orientations to each keypoint location in order to ensure that the candidate points are invariant to orientation. The orientation of the keypoint is calculated based on the orientation and magnitude of the pixels in a region around the keypoint. The process is as follows:

1. For every keypoint in the image scale $L(x, y, \sigma)$, a region W (with a maximum size up to 16×16) around the point is generated.
2. For every pixel in the region,

(a) The magnitude $m(x, y)$ of the pixel is calculated:

$$m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + L(x, y + 1) - L(x, y - 1))^2}. \tag{A.12}$$

(b) The orientation $\theta(x, y)$ at each pixel is calculated:

$$\theta(x, y) = \arctan\left(\frac{(L(x, y + 1) - L(x, y - 1))}{(L(x + 1, y) - L(x - 1, y))}\right). \tag{A.13}$$

3. A histogram of 36 bins is formed, with each bin representing 10 of the total 360 degrees.

4. The value at each bin is the sum of the magnitude of the pixels at this orientation weighted according to a Gaussian weighted window of the same size as W and with $\sigma = 1.5\sigma_0$, where σ_0 is the scale used to generate the first Gaussian image in the octave of the detected point. For example, if the point whose orientation is currently being calculated is detected in the second DoG of the first octave, then the σ_0 is equal to the σ used to generate the first Gaussian image $L(x, y, \sigma)$ in the first octave (which, in this case, = 1.6). For example, given a keypoint (x_1, y_1) and $W = 16 \times 16$, for every pixel in W, the magnitude $m(x, y)$ and $\theta(x, y)$ are calculated. If there are 5 pixels with a direction of 95 degrees and with magnitudes equal to $m_1 : m_5 = 50, 80, 20, 100, 40$, respectively. Therefore, the value V in the tenth bin of the histogram that covers the degrees from 90 to 100 is then calculated as follow:
$V = (50 * G) + (80 * G) + (20 * G) + (100 * G) + (40 * G)$,
where G is the Gaussian filter with the same size as W and with a sigma value equal to $\sigma = 1.5\sigma_0$.

5. The peak point in the histogram and the 80% bins are selected to be the orientations of the key point. For example, if the tenth bin that covers the degrees from 90 to 100 has a peak value $V = 200$, and if there is another bin, e.g., the eighth bin, that covers the degrees from 70 to 80 and has a value $V = 180$, which is greater than 80% of the peak, then this point will have two orientations from 70 to 80 and from 90 to 100. On the other hand, if no values at any bin reach 80% of the peak, this point will have only one orientation: the orientation of the peak.

- **Key point descriptor:** After the candidate points have been selected, the local image gradients are measured in the region around each key point. Orientation histograms are created from 4×4 pixel neighbourhoods with eight bins each, and the magnitude and orientation values of the samples in a 16×16 region around the key points are calculated. When $4 \times 4 = 16$ histograms exist, each with eight bins, the feature vector of each key point with 128 elements is calculated as follows:

  1. The image used to calculate the descriptors is selected $L(x, y, \sigma)$. The first Gaussian image $L(x, y, \sigma)$ in the octave where the keypoint is detected is used to calculate the descriptors. For example, if the current point is detected in the second octave, then the image used to calculate the descriptors is the first Gaussian image in the second octave.
  2. A 16×16 region is generated around each keypoint in $L(x, y, \sigma)$.
  3. If the current point has more than one orientation, then, starting from its second orientation, the image $L(x, y, \sigma)$ is rotated using the orientation of the point before starting the process of descriptor calculation. In the first orientation of the point, $L(x, y, \sigma)$ is not rotated, and the descriptors are calculated directly from $L(x, y, \sigma)$.
  4. This 16×16 window is divided into 16 different 4×4 subregions.
  5. At each subregion, the magnitude $m(x, y)$ and the orientation $\theta(x, y)$ are calculated at each pixel.
  6. An eight-bin histogram is initiated, in which each bin covers 45 degree of orientation.
  7. The value at each bin is the sum of the magnitude at each degree (as in the previous step), weighted by a 4×4 Gaussian filter with $\sigma = 1.5*$ the width of the descriptor window (e.g., up to 16×16).
  8. The descriptor vector at each point is formed from eight bins from each subregion (16 subregions), i.e., $16 \times 8 = 128$.

- **Matching process**

  The main challenge associated with SIFT is the matching process, which requires a large number of comparisons. The matching process is performed as follows:

Given two images $I1(x, y)$, and $I2(x, y)$ with descriptor matrices M_1, M_2 of dimensions $M_1 : N_1 \times k$ and $M_2 : N_2 \times k$, where $k = 128$,

1. For every row in M_1, the Euclidean distances between this row and all rows in M_2 are calculated:
$$E(p, q) = \sum_{i=1}^{N_2} (p_i - q_i)^2, \tag{A.14}$$
where $p_1 \cdots k$ are the elements of a row in M_1, and $q_1 \cdots k$ are the elements of a row in M_2.
2. The minimum distance, or nearest point is selected.
3. The second nearest point is selected.
4. The point with the minimum distance is considered the matching point if the ratio between the minimum distance and the second minimum distance is less than 0.8 of the distance to the first nearest point. Otherwise the point is discarded.
5. In summary, point p_1 in M_1 is matched to point p_2 in M_2 if

   (a) The Euclidean distance d_1 between p_1 and p_2 is the minimum distance between p_1 and every point in M_2.
   (b) The ratio of the distance d_1 and the Euclidean distance d_2 between p_1 and p_3 (the second nearest point) is less than 0.8.

   For example, if the distance $d_1 = 0.1$, and $d_2 = 0.15$, then p_1 and p_2 are matched. On the other hand, if the distance $d_1 = 0.1$, and $d_2 = 0.12$, then p_1 and p_2 are not matched.

SIFT has been used in numerous applications, such as object recognition, image stretching, 3D modeling, gesture recognition, video tracking, match moving, and face authentication [89] as well as to provide a self-calibration strategy for estimating intrinsic and extrinsic camera parameters [90]. A number of published reports [91, 92, 93, 94] help provide an understanding of the SIFT technique.