University of Miami
Scholarly Repository

2012-02-17

Segmentation of Multiple Sclerosis Lesions in Brain MRI

Bassem A. Abdullah

University of Miami, bassem.abdullah@gmail.com

Follow this and additional works at: https://scholarlyrepository.miami.edu/oa_dissertations

UNIVERSITY OF MIAMI

SEGMENTATION OF MULTIPLE SCLEROSIS LESIONS IN BRAIN MRI

By

Bassem A. Abdullah

A DISSERTATION

Submitted to the Faculty
of the University of Miami
in partial fulfillment of the requirements for
the degree of Doctor of Philosophy

Coral Gables, Florida

May 2012

www.manaraa.com

UNIVERSITY OF MIAMI

A dissertation submitted in partial fulfillment of

the requirements for the degree of

Doctor of Philosophy

SEGMENTATION OF MULTIPLE SCLEROSIS LESIONS IN BRAIN MRI

Bassem A. Abdullah

Approved:

_____   _____
Mei-Ling Shyu, Ph.D.   Terri A. Scandura, Ph.D.
Associate Professor of   Dean of the Graduate School
Electrical and Computer Engineering

_____   _____
Mohamed Abdel-Mottaleb, Ph.D.   Nigel John, Ph.D.
Professor of   Lecturer of
Electrical and Computer Engineering   Electrical and Computer
                                           Engineering

_____   _____
Pradip Pattany, Ph.D.   Weizhao Zhao, Ph.D.
Associate Professor of Radiology   Associate Professor of
                                           Biomedical Engineering

ABDULLAH, BASSEM A.   (Ph.D., Electrical and Computer Engineering)

<u>Segmentation of Multiple</u>   (May 2012)
<u>Sclerosis Lesions in Brain MRI</u>

Abstract of a dissertation at the University of Miami.

Dissertation supervised by Professor Mei-Ling Shyu.
No. of pages in text. (167)

Multiple Sclerosis (MS) is an autoimmune disease of central nervous system. It may result in a variety of symptoms from blurred vision to severe muscle weakness and degradation, depending on the affected regions in brain. To better understand this disease and to quantify its evolution, magnetic resonance imaging (MRI) is increasingly used nowadays. Manual delineation of MS lesions in MR images by human expert is time-consuming, subjective, and prone to inter-expert variability. Therefore, automatic segmentation is needed as an alternative to manual segmentation. However, the progression of the MS lesions shows considerable variability and MS lesions present temporal changes in shape, location, and area between patients and even for the same patient, which renders the automatic segmentation of MS lesions a challenging problem.

In this dissertation, a set of segmentation pipelines are proposed for automatic segmentation of multiple sclerosis (MS) lesions from brain magnetic resonance imaging (MRI) data. These techniques use a trained support vector machine (SVM) to discriminate between the blocks in regions of MS lesions and the blocks in non-MS lesion regions mainly based on the textural features with aid of the other features. The main contribution of this set of frameworks is the use of textural features to detect MS lesions in a fully automated approach that does not rely on manually delineating the MS

lesions. In addition, the technique introduces the concept of the multi-sectional views segmentation to produce verified segmentation. The multi-sectional views pipeline is customized to provide better segmentation performance and to benefit from the properties and the nature of MS lesion in MRI. These customization and enhancement leads to development of the customized MV-T-SVM.

The MRI datasets that were used in the evaluation of the proposed pipelines are simulated MRI datasets (3 subjects) generated using the McGill University BrainWeb MRI Simulator, real datasets (51 subjects) publicly available at the workshop of MS Lesion Segmentation Challenge 2008 and real MRI datasets (10 subjects) for MS subjects acquired at the University of Miami. The obtained results indicate that the proposed method would be viable for use in clinical practice for the detection of MS lesions in MRI.

Acknowledgment

I would like to thank GOD (ALLAH) in the first place, who created suitable circumstances for me to accomplish this work including putting me in contact with many people who were keen to provide help and support to me to finalize and complete my Ph.D. degree. I would like to thank the prophet Mohamed who is my inspiration for success in life, for the betterment of mankind, and for getting the reward from ALLAH in the heavens on the Day of Judgment.

I extend my sincere appreciation to many people who made this work possible. Many thanks to Dr. Akmal Younis for his valuable directions, for the help and support he provided in research, and for the many hours he spent with me in meetings and the time he devoted to provide deep thought, analysis, and judgment regarding my work. I ask ALLAH to bless him and to make him reach paradise. I ask ALLAH to bless his young kids. I am highly indebted to both Dr. Mei-Ling Shyu and Dr. Nigel John for their support in helping me to complete the thesis after Dr. Akmal Younis passed away. I would like to thank every professor, colleague, and employee at the University of Miami who helped me and made my mission easy. I would like to thank Mohamed Abdel-Latif for his help in reviewing the dissertation write up.

I would also like to acknowledge with much appreciation the crucial role of my family. I would like to thank my father and mother and ask ALLAH to bless them. I would like to thank my sisters, uncles, aunts, brothers-in-law, cousins, and close friends for their encouragement and financial support. A special thanks to my dear wife who

provided a lot of encouragement, patience, and sacrificed a lot of time that I could not spend with her or with our daughter, Mariam.

Contents

vi

CHAPTER 5 CUSTOMIZED MV-T-SVM FOR LESIONS AND TISSUES DETECTION IN MRI... 125

List of Figures

List of Tables

Chapter 1
Introduction

Multiple sclerosis (MS) is a chronic idiopathic disease that results in multiple areas of inflammatory demyelization in the central nervous system (CNS). Progressive MS lesion formation often leads to cognitive decline and physical disability. Due to its sensitivity in detecting MS lesions, Magnetic Resonance Imaging (MRI) has become an effective tool for diagnosing MS and monitoring its progression. Accurate manual assessment of each lesion in MR images would be a demanding and laborious task, and would also be subjective and have poor reproducibility. Automatic Segmentation offers an attractive alternative to manual segmentation which remains a time-consuming task and suffers from intra- and inter-expert variability. However, the progression of the MS lesions shows considerable variability and MS lesions present temporal changes in shape, location, and area between patients and even for the same patient. This makes the automatic segmentation of MS lesions a challenging problem.

The objective of the research work included in this thesis is to provide a robust technique for automatic segmentation of multiple sclerosis lesions from brain MR images. The study focuses on using textural analysis in the segmentation process. Both single channel and multichannel MRIs are included in the research.

Texture analysis in MRI has been used with some success in neuro-imaging to detect lesions and abnormalities. Textural analysis refers to a set of processes applied to characterize special variation patterns of voxels grayscale in an image. Segmentation

1

based on texture properties is promising in cases of lesions that are inhomogeneous, not sharp, and faint, but show an intensity pattern that is different from the adjacent healthy tissue.

The use of textural features is a promising approach for providing accurate segmentation of MS lesions, especially when taking advantage of various MRI sequences to benefit from their relevant and complementary information for MS segmentation. However, to the best of our knowledge, texture based MS segmentation approaches that have been previously reported were applied to ROIs that are manually selected by an expert to indicate potential regions including MS lesions, which makes the segmentation process semi-automated. Therefore, efforts are needed to automate the use of textural features in the detection of MS lesions.

Support Vector Machines (SVM's) are supervised machine learning techniques, recently developed in the framework of statistical learning theory. They have been used with a great success in a variety of applications such as text categorization, face recognition, and bioinformatics. The great advantage of the SVM technique over conventional techniques, besides the well-established theoretical definition, is its capacity of working with high dimensional feature vectors without losing the generalization performance.

Motivated by the ability of textural features to describe the special patterns such as the MS lesions and the ability of the support vector machines (SVM) to work with high dimensional feature vectors without losing the generalization performance, a

technique that combines these two aspects is proposed aiming to get a robust competitive MS lesions automatic segmentation clinical tool.

Most of segmentation techniques in the literature suffer from high false positives due to the similarity between MS lesions and the white matter tissue and also due to basing the learning on voxels while the lesions form regions. Comprehensive study of false positive and negative in MS segmentation is needed to introduce a generic post-processing method that can be merged with our proposed segmentation technique or any other technique to provide more accurate and *clinical* friendly results.

Most of segmentation techniques process the MRI axial slices. Thanks to the 3D nature of the MRI, data can be studied from three sectional views: axial, sagittal, and coronal. Segmentation results obtained from a classifier that is trained with axial slices data can be verified, assessed or enhanced by segmentation performed by classifier on the other brain sectional views. Multi-views segmentation is a concept in our research plan to be incorporated in the proposed technique.

Utilizing textural features without the need for manual labeling of ROIs, utilizing SVM, enabling segmentation using multi-channels MRI data and taking into account the different sectional views of the lesion volume are the pillars of the proposed technique.

In this dissertation, we propose a segmentation framework which is based on an SVM fed by feature vector consists mainly of textural features with other groups of features. We refer to this core engine by textural based SVM. A comprehensive post processing module that addresses all possible segmentation errors is proposed to improve the segmentation quality. The progress of development of the segmentation framework

through this thesis was initialized by providing a technique for segmentation of FLAIR slices in axial view. Improvements were made to this technique to support the multi-channels MRI. Then the proposed multi-sectional views segmentation concept was used to provide a robust technique that provides competitive results.

The multi-sectional views segmentation framework is customized and a more robust framework (Customized MV-T-SVM) is presented to overcome the limitations and issues raised while testing. The customized MV-T-SVM uses 3D nature of the MRI information, template tissue information provided for images which are registered to an anatomical atlas and textural information from the multi-channels images. SVM learning procedure is revised to address imbalanced classes and to synchronize the learning of the engines of the three views.

This technique is tested using both simulated and real MRI datasets. The MRI datasets that were used in the evaluation of the proposed pipelines are simulated MRI datasets (3 subjects) generated using the McGill University BrainWeb MRI Simulator [1], real datasets (51 subjects) publicly available at the Workshop of MS Lesion Segmentation Challenge 2008 [2] and real MRI datasets (10 subjects) for MS subjects acquired at the University of Miami. The obtained results indicate that the proposed method would be viable for use in clinical practice for the detection of MS lesions in MRI.

1.1 Scope and Limitations of the Proposed MS Lesions Segmentation Framework

The proposed MS lesions segmentation framework deals with T1, T2, PD, and FLAIR the MRI channels. The Diffusion Tensor Imaging (DTI) is not included in our research work. The proposed technique is designed mainly for MS lesions detection and the different tissues of the brain are not segmented. The current state of the patient brain is the input dataset for the MS detection technique and there is no follow up for the lesion progression over periods of time.

1.2 Outline of the Dissertation

The dissertation is organized as follows:

In **Chapter 2**, we provide background topics that will be used in the next chapters. These background topics include the Multiple Sclerosis (MS) disease, Magnetic Resonance Imaging (MRI) sequences, segmentation process, textural feature extraction techniques and support vector machines.

In **Chapter 3**, we provide a comprehensive survey that covers the state of the art techniques that address the automatic segmentation of MS lesions in the MRI.

In **Chapter 4**, we present our proposed approach developed that uses the textural features with a robust machine learning technique to achieve multiple sclerosis (MS) lesions segmentation in the brain MR images. The approach is presented in form of single view segmentation pipelines (single channel MRI and multi-channel MRI) and multi-sectional views pipeline.

In **Chapter 5**, we present the customized MV-T-SVM which is proposed to provide optimal segmentation for MS and overcome the limitations and issues raised while testing the multi-sectional views' segmentation pipeline.

In **Chapter 6**, we summarize the work presented in this dissertation and we provide further insight regarding the areas of potential improvement.

Chapter 2
Background

This chapter provides the research objectives along with background topics including the Multiple Sclerosis (MS) disease, Magnetic Resonance Imaging (MRI) sequences, segmentation process, common textural feature extraction techniques, and the definition of the support vector machines.

2.1 Multiple Sclerosis (MS)

Multiple sclerosis (MS) is one of the most common diseases of the central nervous system. Today over 2,500,000 people around the world have MS [**3**]. MS is the result of damage to myelin, a protective sheath surrounding nerve fibers of the central nervous system. When myelin is damaged, this interferes with messages between the brain and other parts of the body. Symptoms vary widely and include blurred vision, weak limbs, tingling sensations, unsteadiness and fatigue. For some people, MS is characterized by periods of relapse and remission while for others it has a progressive pattern. For everyone, it makes life unpredictable.

The cause of multiple sclerosis is not yet known, but thousands of researchers all over the world are meticulously putting the pieces of this complicated puzzle together. The damage to myelin in MS may be due to an abnormal response of the body's immune system, which normally defends the body against invading organisms (bacteria and viruses). Many of the characteristics of MS suggest an 'auto-immune' disease whereby the

7

body attacks its own cells and tissues, which in the case of MS is myelin. Researchers do not know what triggers the immune system to attack myelin, but it is thought to be a combination of several factors. One theory is that a virus, possibly lying dormant in the body, may play a major role in the development of the disease and may disturb the immune system or indirectly instigate the auto-immune process. A great deal of research has taken place in trying to identify an MS virus. It is probable that there is no one MS virus, but that a common virus, such as measles or herpes, may act as a trigger for MS. This trigger activates white blood cells (lymphocytes) in the blood stream, which enter the brain by making vulnerable the brain's defense mechanisms (i.e. the blood/brain barrier). Once inside the brain these cells activate other elements of the immune system in such a way that they attack and destroy myelin. Women are more likely to develop multiple sclerosis than men, with MS occurring 50% more frequently in women than in men (i.e. 3 women for every 2 men). Multiple sclerosis is a disease of young adults; the mean age of onset is 29-33 years, but the range of onset is extremely broad from approximately 10-59 years. Due to its sensitivity in detecting MS lesions, Magnetic Resonance Imaging (MRI) has become an effective tool for diagnosing MS and monitoring its progression [**4**]. In the following section, a brief summary of the MRI is presented.

2.2 Magnetic Resonance Imaging (MRI)

Magnetic resonance imaging (MRI) is a medical imaging technique used in radiology to visualize detailed internal structures. MRI makes use of the property of nuclear magnetic resonance (NMR) to image nuclei of atoms inside the body.

An MRI machine uses a powerful magnetic field to align the magnetization of protons in the body, and radio frequency fields to systematically alter the alignment of this magnetization. This causes the protons to produce a rotating magnetic field of larger frequency detectable by the scanner and this information is recorded to construct an image of the scanned area of the body [5] Strong magnetic field gradients cause nuclei at different locations to rotate at different speeds. 3-D spatial information can be obtained by providing gradients in each direction.

MRI provides good contrast between the different soft tissues of the body, which make it especially useful in imaging the brain, muscles, the heart, and cancers compared with other medical imaging techniques such as computed tomography (CT) or X-rays. Unlike CT scans or traditional X-rays, MRI uses no ionizing radiation.

An MRI sequence is an ordered combination of radiofrequency (RF) and gradient pulses designed to acquire the data to form the image. The data to create an MR image is obtained in a series of steps. First the tissue magnetization is excited using an RF pulse in the presence of a slice select gradient. The other two essential elements of the sequence are phase encoding and frequency encoding (read out), which are required to spatially localize the protons in the other two dimensions. Finally, after the data has been collected, the process is repeated for a series of phase encoding steps. The MRI sequence parameters are chosen to best suit the particular clinical application.

The gradient echo (GE) sequence is the simplest type of MRI sequence. It consists of a series of excitation pulses, each separated by a repetition time TR. Data is acquired at some characteristic time after the application of the excitation pulses and this is defined

as the echo time TE. The contrast in the image will vary with changes to both TR and TE. Advantages of this sequence are fast imaging, low Flip Angle and less RF power where the disadvantages are difficulty to generate good T2 contrast, sensitivity to B0 inhomogeneities and sensitivity to susceptibility effects.

The spin echo (SE) sequence is similar to the GE sequence with the exception that there is an additional 180° refocusing pulse present.

Inversion recovery (IR) sequence is usually a variant of a SE sequence in that it begins with a 180º inverting pulse. This inverts the longitudinal magnetization vector through 180º. When the inverting pulse is removed, the magnetization vector begins to relax back to B0. A 90º excitation pulse is then applied after a time from the 180º inverting pulse known as the TI (time to inversion). The contrast of the resultant image depends primarily on the length of the TI as well as the TR and TE. The contrast in the image primarily depends on the magnitude of the longitudinal magnetization (as in spin echo) following the chosen delay time TI.

Contrast is based on T1 recovery curves following the 180º inversion pulse. Inversion recovery is used to produce heavily T1 weighted images to demonstrate anatomy. The 180º inverting pulse can produce a large contrast difference between fat and water because full saturation of the fat or water vectors can be achieved by utilizing the appropriate TI.

FLAIR is another variation of the inversion recovery sequence. In FLAIR, the signal from fluid (e.g. cerebrospinal fluid (CSF)) is nulled by selecting a TI corresponding to the time of recovery of CSF from 180º inversion to the transverse plane.

The signal from CSF is nullified and FLAIR is used to suppress the high CSF signal in T2 and proton density weighted images so that pathology adjacent to the CSF is seen more clearly. A TI of approximately 2000 ms achieves CSF suppression at 3.0T.

Soft Tissue Contrast in MRI

Contrast is the means by which it is possible to distinguish among soft tissue types owing to differences in observed MRI signal intensities. For example, in musculoskeletal imaging, there are differences among cartilage, bone, and synovial fluid. In neuroimaging, there are differences between white and grey matter. The fundamental parameters that affect tissue contrast are the T1 and T2 values, proton density, tissue susceptibility and dynamics. Tissue pathology will also affect contrast, as will the static field strength, the type of sequences used, contrast media and the sequence parameters (TR, TE, TI, FA, SNR etc…).

T1 Weighting

To demonstrate T1, proton density or T2 contrast, specific values of TR and TE are selected for a given pulse sequence. The selection of appropriate TR and TE weights an image so that one contrast mechanism predominates over the other two. A T1 weighted image is one where the contrast depends predominantly on the differences in the T1 times between tissues e.g. fat and water. Because the TR controls how far each vector can recover before it is excited by the next RF pulse, to achieve T1 weighting the TR must be short enough so that neither fat nor water has sufficient time to fully return to B0. If the TR is too long, both fat and water return to B0 and recover their longitudinal

magnetization fully. When this occurs, T1 relaxation is complete in both tissues and the differences in their T1 times are not demonstrated on the image.

T2 Weighting

A T2 weighted image is one where the contrast predominantly depends on the differences in the T2 times between tissues e.g. fat and water. The TE controls the amount of T2 decay that is allowed to occur before the signal is received. To achieve T2 weighting, the TE must be long enough to give both fat and water time to decay. If the TE is too short, neither fat nor water has had time to decay and therefore the differences in their T2 times are not demonstrated in the image.

Proton Density (PD) Weighting

A proton density image is one where the difference in the numbers of protons per unit volume in the patient is the main determining factor in forming image contrast. Proton density weighting is always present to some extent. In order to achieve proton density weighting, the effects of T1 and T2 contrast must be diminished, so that proton density weighting can dominate. A long TR allows tissues e.g. fat and water to fully recover their longitudinal magnetization and therefore diminishes T1 weighting. A short TE does not give fat or water time to decay and therefore diminishes T2 weighting. Figure 2-1 below shows a comparison of T1, T2, and PD weighting.
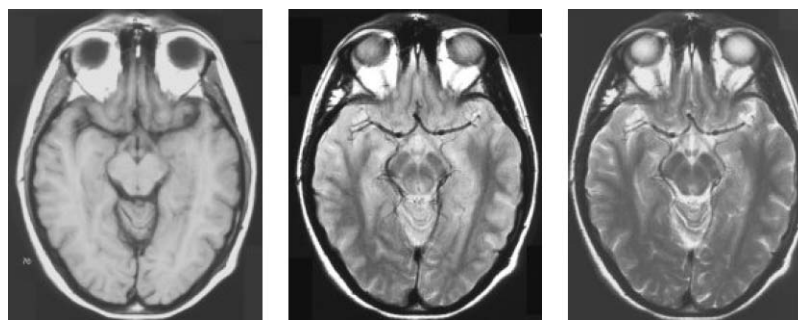
Figure 2-1 : T1, PD and T2 Weighted Axial Brain Images

In Table 2-1, the TR and TE for T1, T2, and PD weighted sequences are compared.

Weighting	TR	TE
T1	Short	Short
T2	Long	Long
PD	Long	Short

Table 2-1 : Summary of T1, T2, and PD Weighting

2.3 MS Segmentation in MRI of the Brain

Accurate manual assessment of each lesion in MR images would be a demanding and laborious task, and would also be subjective and have poor reproducibility [**6**]. Automatic Segmentation offers an attractive alternative to manual segmentation which remains a time-consuming task and suffers from intra- and inter-expert variability [**7**]. However, the progression of the MS lesions shows considerable variability and MS lesions present temporal changes in shape, location, and area between patients and even for the same patient [**8**], [**9**], [**10**], [**11**]. This makes the automatic segmentation of MS lesions a challenging problem.

2.4 Texture Extraction Techniques

Textural features can be categorized according to the matrix or vector used to calculate the feature. In this section, we are interested with histogram, gradient, run-length matrix and co-occurrence based features. These categories include features that are selected after being tested to be identifying for the texture of regions that suffer from the multiple sclerosis lesions. For all feature calculations, the image is represented by a function f(x,y) of two space variables x and y, x=0,1,…N-1 and y=0,1,…, M-1. The function f(x,y) can take any value i=0,1,…,G-1 where G is total number of intensity levels in the image.

2.4.1 Histogram based Features

The intensity level histogram is a function h(i) providing, for each intensity level i, the number of pixels in the whole image having this intensity.

$$h(i) = \sum_{x=0}^{N-1}\sum_{y=0}^{M-1}\delta(f(x,y),i)\,;\delta(i,j) = \begin{cases} 1, & j=i \\ 0, & j \neq i \end{cases} \tag{2-1}$$

The histogram is a concise and simple summary of the statistical information contained in the image. Dividing the histogram h(i) by the total number of pixels in the image provides the approximate probability density of the occurrence of the intensity levels p(i), given by:

$$p(i) = h(i)\,/\,NM \tag{2-2}$$

The following set of textural features is calculated from the normalized histogram:

$$\text{Mean:}\ \mu = \sum_{i=0}^{G-1} ip(i) \tag{2-3}$$

$$\text{Variance: } \sigma^2 = \sum_{i=0}^{G-1} (i-\mu)^2 \, p(i) \tag{2-4}$$

$$\text{Skewness: } \mu_3 = \frac{1}{\sigma^3} \sum_{i=0}^{G-1} (i-\mu)^3 \, p(i) \tag{2-5}$$

$$\text{Kurtosis: } \mu_4 = \frac{1}{\sigma^4} \sum_{i=0}^{G-1} (i-\mu)^4 \, p(i) - 3 \tag{2-6}$$

2.4.2 Gradient based Features

The gradient matrix element g(x,y) is defined for each pixel in the image based on the neighborhood size. For a 3x3 pixels neighborhood, g is defined as follows:

$$\begin{aligned}
\Delta_x &= f(x-1, y) - f(x+1, y) \\
\Delta_y &= f(x, y-1) - f(x, y+1) \\
g(x,y) &= \sqrt{\Delta_x{}^2 + \Delta_y{}^2}
\end{aligned} \tag{2-7}$$

The following set of textural features is calculated from the gradient matrix:

$$\text{Mean of absolute gradient (GrMean)} = \frac{1}{NM} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} g(x,y) \tag{2-8}$$

$$\text{Variance of absolute gradient (GrVariance)} = \frac{1}{NM} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} (g(x,y) - GrMean)^2 \tag{2-9}$$

Skewness and kurtosis of the absolute gradient can be calculated similar to those calculated for histogram.

2.4.3 Run Length Matrix based Features

The run length matrix is defined for a specific direction. Usually a matrix is calculated for the horizontal, vertical, 45° and 135° directions. The matrix element r(i,j) is defined as the

number of times there is a run of length j having gray level i. Let G be the number of gray levels and N_r be the number of runs. The following set of textural features is calculated from the run length matrix:

$$\text{Short run emphasis inverse moments (ShrtREmph)} = (\sum_{i=0}^{G-1}\sum_{j=1}^{N_r}\frac{r(i,j)}{j^2})/C \qquad (2\text{-}10)$$

$$\text{Long run emphasis moments (LngREmph)} = (\sum_{i=0}^{G-1}\sum_{j=1}^{N_r}j^2 r(i,j))/C \qquad (2\text{-}11)$$

$$\text{Gray level non-uniformity (GLevNonUni)} = (\sum_{i=0}^{G-1}(\sum_{j=1}^{N_r}r(i,j))^2)/C \qquad (2\text{-}12)$$

$$\text{Run length non-uniformity (RLNonUni)} = (\sum_{j=1}^{N_r}(\sum_{i=0}^{G-1}r(i,j))^2)/C \qquad (2\text{-}13)$$

where the normalization coefficient C is defined as follows:

$$C = \sum_{i=0}^{G-1}\sum_{j=1}^{N_r}r(i,j) \qquad (2\text{-}14)$$

2.4.4 Co-occurrence Matrix based Features

The co-occurrence matrix is a form of second order histogram that is defined for certain angle θ and certain distance d. The matrix element $h_{d\theta}(i,j)$ is the number of times $f(x1,y1)$ = i and $f(x2,y2)$ = j where $(x2,y2)=(x1,y1) + (d\cos\theta, d\sin\theta)$. Usually the co-occurrence matrix is calculated for d = 1 and 2 with angles θ = 0°, 45°, 90° and 135°. When the matrix element $h_{d\theta}(i,j)$ is divided by the total number of neighboring pixels, the matrix becomes the estimate of the joint probability $co_{d\theta}(i,j)$ of two pixels, a distance d apart along a given direction θ having co-occurring values i and j. Let μ_x, μ_y, σ_x and σ_y denote the

mean and standard deviation of the row and column sums of the matrix co, respectively. The following set of textural features is calculated from the co-occurrence matrix:

$$\text{Angular second moment (AngScMom)} = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1}(co(i,j))^2 \tag{2-15}$$

$$\text{Contrast} = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1}(i-j)^2\,co(i,j) \tag{2-16}$$

$$\text{Correlation} = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1}\frac{ijco(i,j)-\mu x\mu y}{\sigma_x\sigma_y} \tag{2-17}$$

$$\text{Inverse Difference} = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1}\frac{co(i,j)}{1+(i-j)^2} \tag{2-18}$$

$$\text{Entropy} = -\sum_{i=0}^{G-1}\sum_{j=1}^{N_r}co(i,j)\log_2(co(i,j)) \tag{2-19}$$

2.5 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised learning algorithm, which has at its core a method for creating a predictor function from a set of training data where the function itself can be a binary, a multi-category, or even a general regression predictor. To accomplish this mathematical feat, SVMs find a hypersurface which attempts to split the positive and negative examples with the largest possible margin on all sides of the hyperplane. It uses a kernel function to transform data from input space into a high dimensional feature space in which it searches for a separating hyperplane. The most common types of kernel functions used are: polynomial for polynomial classifiers, Gaussian for radial-basis function (RBF) classifiers, and tangent hyperbolic for two-layer perceptron classifiers.

For binary classification problems, the main idea of the SVM is to find a decision boundary between classes in the original feature space by mapping the feature vectors to a high dimensional space, where the features are more likely to be linearly separable. The optimal separating hyperplane (OSH) is choose in the high-dimensional space as the one that maximizes the margin of separability between the two sets of data points. This choice counteracts the increasing generalization error when constructing a decision boundary in higher-dimensional space. This approach also satisfies the Empirical Risk Minimization principle, which seeks to minimize the upper bound on the generalization error [12].

The SVM formulation proposed initially by Vapnik [12] does not include any criteria to select the regularization parameter C or a kernel function that gives good generalization (or results in a classifier with low expected error bound). In practical problems, where the dataset does not present a large number of vectors, the VC-dimension and the training errors can be used to select the best kernel function. However, in problems involving several thousands of examples, such as segmentation of MS-lesions from magnetic resonance images, this process becomes a very time consuming task.

2.5.1 Kernels: From Linear to Non-Linear Classifiers

The data for a two class learning problem consists of objects labeled with one of two labels corresponding to the two classes; for convenience we assume the labels are +1 (positive examples) or -1 (negative examples). In what follows x denotes a vector with components x_i. The notation x_i will denote the ith vector in a dataset $\{(x_i, y_i)\}_{i=1}^{n}$ where y_i

is the label associated with x_i. The objects x_i are called patterns or examples. The examples are assumed to belong to some set X. Initially, the examples are assumed to be vectors, but once kernels are introduced this assumption will be relaxed, at which point they could be any continuous/discrete object. A key concept required for defining a linear classifier is the dot product between two vectors, defined as $w^T x = \sum_i w_i x_i$. A linear classifier is based on a linear discriminant function of the form

$$f(x) = w^T x + b \tag{2-20}$$

The vector w is known as the weight vector and b is called the bias. Consider the case b=0 first. The set of points x such that $w^T x = 0$ are all points that are perpendicular to w and go through the origin, a line in two dimensions, a plane in three dimensions, and more generally, a hyperplane. The bias b translates the hyperplane away from the origin. The hyperplane f(x) given by:

$$\{x: f(x) = w^T x + b = 0\} \tag{2-21}$$

divides the space into two: the sign of the discriminant function f(x) denotes the side of the hyperplane a point is on as shown in Figure 2-2.
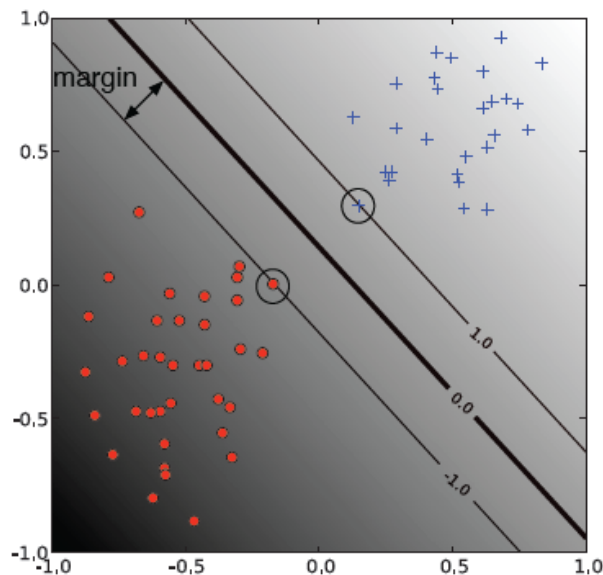
Figure 2-2 : A linear SVM. The circled data points are the support vectors - the examples that are closest to the decision boundary. They determine the margin with which the two classes are separated.

The boundary between regions classified as positive and negative is called the decision boundary of the classifier. The decision boundary defined by a hyperplane is said to be linear because it is linear in the input examples (Equation 2-21). A classifier with a linear decision boundary is called a linear classifier. Conversely, when the decision boundary of a classifier depends on the data in a non-linear way the classifier is said to be non-linear.

In many applications a non-linear classifier provides better accuracy. And yet, linear classifiers have advantages, one of them being that they often have simple training algorithms that scale well with the number of examples.

The naive way of making a non-linear classifier out of a linear classifier is to map our data from the input space X to a feature space F using a non-linear function $\emptyset: X \rightarrow F$. In the space F the discriminant function is:

$$f(x) = w^T \emptyset(x) + b \tag{2-22}$$

The approach of explicitly computing non-linear features does not scale well with the number of input features: when applying the mapping from the above example the dimensionality of the feature space F is quadratic in the dimensionality of the original space. These results in a quadratic increase in memory usage for storing the features and a quadratic increase in the time required to compute the discriminant function of the classifier. This quadratic complexity is feasible for low dimensional data; but when handling gene expression data that can have thousands of dimensions, quadratic complexity in the number of dimensions is not acceptable. Kernel methods solve this issue by avoiding the step of explicitly mapping the data to a high dimensional feature-space. Suppose the weight vector can be expressed as a linear combination of the training examples, i.e. $w = \sum_{i=1}^{n} \alpha_i x_i$. Then:

$$f(x) = \sum_{i=1}^{n} \alpha_i x_i^T x + b \tag{2-23}$$

In the feature space, F this expression takes the form:

$$f(x) = \sum_{i=1}^{n} \alpha_i \emptyset(x_i)^T \emptyset(x) + b \tag{2-24}$$

The representation in terms of the variables α_i is known as the dual representation of the decision boundary. As indicated above, the feature space F may be high dimensional, making this trick impractical unless the kernel function $K(x, x')$ defined as

$$K(x, x') = \emptyset(x)^T \emptyset(x') \tag{2-25}$$

can be computed efficiently. In terms of the kernel function the discriminant function is:

$$f(x) = \sum_{i=1}^{n} \alpha_i k(x, x_i) + b \qquad\qquad (2\text{-}26)$$

A linear decision boundary can be "kernelized", i.e. its dependence on the data is only through dot products. In order for this to be useful, the training algorithms need to be kernelizable as well. It turns out that a large number of machine learning algorithms can be expressed using kernels including ridge regression, the perceptron algorithm, and SVMs.

2.5.2 Support Vector Machines: Large Margin Classifier

In what follows the term linearly separable is used to denote data for which there exists a linear decision boundary that separates positive from negative examples as shown in Figure 2-2. Initially, linearly separable data is assumed, and later how to handle data that is not linearly separable is indicated.

In this section, the notion of a margin is defined. For a given hyperplane we denote by $x_+(x_-)$ the closest point to the hyperpalne among the positive (negative) examples. The norm of a vector w denoted by $\|w\|$ is its length. A unit vector $\widehat{W}$ in the direction of w is given by w / $\|w\|$ and has $\|\widehat{W}\| = 1$. From simple geometric considerations the margin of a hyperplane f with respect to a dataset D can be seen to be:

$$m_D(f) = \frac{1}{2}\widehat{W}^T(X_+ - X_-) \qquad\qquad (2\text{-}27)$$

where $\widehat{W}$ is a unit vector in the direction of w, and x_+ and x_- are assumed to be equidistant from the decision boundary i.e.

$$f(X_+) = W^T X_+ + b = a$$

$$f(X_-) = W^T X_- + b = -a$$

<div align="right">(2-28)</div>

for some constant a > 0. Note that multiplying our data points by a fixed number will increase the margin by the same amount, whereas in reality, the margin hasn't really changed - we just changed the "units" with which we measure it. To make the geometric margin meaningful we fix the value of the decision function at the points closest to the hyperplane, and set a = 1 in equation (2-28). Adding the two equations and dividing by ||w|| we obtain:

$$m_D(f) = \frac{1}{2}\widehat{W}^T(X_+ - X_-) = \frac{1}{\|\widehat{W}\|}$$

<div align="right">(2-29)</div>

Now that we have the concept of a margin we can formulate the maximum margin classifier. We will first define the hard margin SVM, applicable to a linearly separable dataset, and then modify it to handle non-separable data. The maximum margin classifier is the discriminant function that maximizes the geometric margin 1/ ||w|| which is equivalent to minimizing ||w||². This leads to the following constrained optimization problem:

$$minimize_{w,b} \quad \frac{1}{2}\|w\|^2$$

<div align="right">(2-30)</div>

$$subject\ to: y_i(W^T X_i + b) \geq 1 \quad i = 1, \ldots., n$$

The constraints in this formulation ensure that the maximum margin classifier classifies each example correctly, which is possible since we assumed that the data is linearly separable. In practice, data is often not linearly separable; and even if it is, a greater margin can be achieved by allowing the classifier to misclassify some points. To allow

errors, the inequality constraints are replaced in equation (2-30) with $y_i(W^T X_i + b) \geq 1 - \varepsilon_i \ \ i = 1, ..., n$.

where $\varepsilon_i \geq 0$ are slack variables that allow an example to be in the margin ($0 \leq \varepsilon_i \leq 1$, also called a margin error) or to be misclassified ($\varepsilon_i > 1$). Since an example is misclassified if the value of its slack variable is greater than 1, $\sum_i \varepsilon_i$ is a bound on the number of misclassified examples. Our objective of maximizing the margin, i.e. minimizing $\frac{1}{2}\|w\|^2$ will be augmented with a term $C \sum_i \varepsilon_i$ to penalize misclassification and margin errors. The optimization problem becomes:

$$minimize_{w,b} \quad \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{n} \varepsilon_i$$

$$subject\ to: y_i(W^T X_i + b) \geq 1 - \varepsilon_i, \ \ \varepsilon_i \geq 0, i = 1, ..., n$$

(2-31)

The constant $C > 0$ sets the relative importance of maximizing the margin and minimizing the amount of slack. This formulation is called the soft-margin SVM, and was introduced by Cortes and Vapnik [**12**]. In the following, the common basic kernel functions:

$$\text{Linear kernel:} \quad K(x_i, x_j) = x_i^T x_j \quad (2\text{-}32)$$

$$\text{Polynomial:} \quad K(x_i, x_j) = (\gamma x_i^T x_j + r)^d \quad (2\text{-}33)$$

$$\text{Radial basis function (RBF):} \quad K(x_i, x_j) = \exp\left(-\gamma\|x_i - x_j\|^2\right); \gamma > 0 \quad (2\text{-}34)$$

$$\text{Sigmoid function:} \quad K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r) \quad (2\text{-}35)$$

where γ, r and d are kernel parameters.

Chapter 3
Automatic Segmentation of Multiple Sclerosis Lesions in Brain MR Images

This chapter provides a survey on the area of segmentation of multiple sclerosis in the Brain MR images. The segmentation techniques, while their approaches vary widely in the literature, their frameworks composed of similar components which are addressed differently in each framework.

The segmentation framework of the multiple sclerosis lesions can be decomposed into set of components as listed in table 1. Each component is labeled (with labels $C_\#$) to help place certain topics in the context of a segmentation framework and to refer easily. MR images are available in different sequences including T1, T2, PD, and FLAIR...etc. One or combination of MRI sequences is selected by any of the segmentation technique to be an input. MRI Sequence selection (C_1) is the entry component. Preprocessing (C_2) is usually applied on the MR images either to enhance the quality of the images or to provide features to be used in the segmentation processing. In Feature Extraction (C_3), each technique extracts image features from the feature space to represent the voxels of the image. Pattern recognition algorithm (C_4) is the core of the segmentation processing. Post-Processing steps (C_5) usually are applied on the results to improve the performance of the segmentation. Performance Evaluation (C_6) is performed to evaluate the accuracy of each of segmentation techniques using different metrics. The segmentation framework (C_7) is the main component that specifies the design of the segmentation process, breaks

25

down the process into smaller components, manages the flow of their outputs and integrates them to get segmentation results.

Label	Component	Description
C_1	MRI Sequence Selection	Choice of MRI modality of the subject brain: Either Single Channel or Multi-Channels and Selection of which channels to use.
C_2	Pre-Processing	Image Processing Steps applied on the subject MRI to enhance the image quality, remove the effect of MRI acquisition noise and aid in feature extraction.
C_3	Feature Extraction	Image Features are extracted to represent the Image Voxels.
C_4	Pattern Recognition Algorithm	Classification of the Voxels into Healthy or MS Lesions Voxels.
C_5	Post-Processing	Image Processing and additional Pattern Recognition Steps applied on the segmentation result to Improve the performance.
C_6	Performance Evaluation	Metrics used to evaluate performance of the segmentation.
C_7	Segmentation Pipeline	Integrating the components of the segmentation framework to achieve the segmentation task.

Table 1 – Components of Multiple Sclerosis Lesions Segmentation Framework

In the following sections, a categorization for the segmentation frameworks is presented followed by discussion for each of the frameworks' components with the related work in literature is reviewed.

3.1 Segmentation Framework

The segmentation framework refers to the concept behind the approach used to perform the MS segmentation. According to the segmentation framework, the problem is defined and the needed components (C_1 to C_6) are implemented. These components are integrated in a pipeline (C_7) that would be the software design of the segmentation framework. The segmentation frameworks in literature can be categorized into two main categories. In the first category, the problem is defined as pattern classification problem. The second category, the problem is modeled by imitating the expert knowledge in labeling the MS

lesion either implicitly or explicitly. Such a categorization is not always straightforward since some techniques resort to more than one strategy to achieve segmentation and thus cannot be sharply categorized.

3.1.1 Pattern Classification based Framework

It is also referred to by feature space clustering. In this framework, the MS lesions segmentation problem is defined as a classification problem. A pattern classification technique from literature is selected or modified. Some frameworks propose a novel one. The pattern classification technique is used to learn the features of the MS lesion voxels and normal voxels in binary classifications or to learn the different brain tissues and including the MS lesion as a tissue in multi-classes classifications.

The work proposed by Geremia et al. [13] and Akselrod-Ballin et al. [14], the decision forests are used to perform the classification. In [15], [16], and [17], a model is proposed and upgraded sequentially that involves using spectral gradient and graph cuts in binary classification of the brain voxels into MS lesion or background normal brain tissues. Expectation maximization (EM) is used by [18] and [19]. K-Nearest Neighbors (KNN) is used by Anbeek et al. [20], Wu et al. [21] and Cardenes et al. [22]. Support Vector Machines (SVM) is used by Ferrari et al. [23]. Principle component analysis (PCA) is used by Kroon et al. [24].

Model outlier detection can be included in this category of segmentation framework where the brain tissues are modeled and the MS lesions are detected as outliers to the model. It is used by Leemput et al. [25] and [26]. In [27] and [28], Brqiue

et al. uses Hidden Markov Chains (HMC) to perform tissues classification and detect the MS lesions as outlier outliers to the model.

In spite of providing segmentation with acceptable performance, these frameworks lack tackling and understanding the MS lesion as an object.

3.1.2 Contextual model based Framework

This framework includes the techniques that model the expert knowledge in MS delineation and tries to get maximum benefit from the MS lesion context information. In this framework, the MS lesions segmentation problem is modeled based on object and scene understanding. In object and scene understanding, it has been increasingly realized that context information plays a vital role [29]. Medical images contain complex patterns including features such as textures (homogeneous, inhomogeneous, and structured) which are also influenced by acquisition protocols. The concept of context covers intra-object consistency (different parts of the same structure) and inter-object configurations (e.g., expected symmetry of left and right hemisphere structures).

In [30], Moora et al. integrated appearance and context information in a seamless way by automatically incorporating a large number of features through iterative procedures.

Scully et al. [31] constructed an explicit model of the feature vector intensities that identify lesion tissue. This model is the joint histogram calculated over the vector image constructed by co-registered T1, T2, and FLAIR slices, then creating an image where each voxel contains a vector representing the T1 intensity, the T2 intensity, the FLAIR intensity, and the tissue class for that voxel's location. The joint histogram then

represents the number of times a given feature vector was labeled lesion, along with the number of times that vector was not labeled lesion and the 4D joint histogram was calculated. An initial predicted lesion label map was constructed for each test subject by setting the voxel value equal to the frequency of that voxel location's feature vector in the 4D joint histogram model. These values were then thresholded in order to reduce the amount of false positives while keeping the number of false negatives low. The threshold value was selected empirically and could likely be improved through the application of a more rigorous experimental process. In order to obtain rough tissue segmentation, KMeans, with initial class means, was performed on the T1image for each subject. This label map was combined with the thresholded lesion mask to create a label map with different integer values for Grey, White, CSF, and Lesion tissue. The generated label map was then filtered so that lesion tissue that was within 2 voxels of CSF tissue was discarded. The label map was then used to pull 200 exemplar points to train a naive Bayesian classifier which then proceeded to classify the vector image composed of the T1, T2, and FLAIR images. Each independent connected component in the lesion map produced by the Bayesian classifier was then filtered based on a minimum lesion size provided by a local MS expert. Each lesion component had to have a least one dimension with three voxels with the other two dimensions being at least two voxels. While lesions may occur below this size they are generally not labeled by human experts due to high error rates.

Zhuge et al. [**32**] utilized absolute fuzzy connectedness and certain morphological operations to generate the brain intracranial mask. The optimum thresholding method was applied to the product image (the image in which voxel values represent T2 value×PD

value) to automatically recognize potential MS lesion sites. Then, the recently developed technique, vectorial scale-based relative fuzzy connectedness, was utilized to segment all voxels within the brain intracranial mask into WM, GM, CSF, and MS lesion regions.

Zhu et al. [**33**] applied knowledge guided information fusion. In this work, T1, T2, and PD images of, providing information on the properties of tissues from different aspects, were treated as three independent information sources for the detection and segmentation of MS lesions. Based on information fusion theory, a knowledge guided information fusion framework is proposed to accomplish 3-D segmentation of MS lesions. This framework consists of three parts: information extraction, information fusion, and decision. Information provided by different spectral images was extracted and modeled separately in each spectrum using fuzzy sets, aiming at managing the uncertainty and ambiguity in the images due to noise and partial volume effect. In the second part, the possible fuzzy map of MS lesions in each spectral image was constructed from the extracted information under the guidance of experts' knowledge, and then the final fuzzy map of MS lesions was constructed through the fusion of the fuzzy maps obtained from different spectrum. Finally, 3-D segmentation of MS lesions was derived from the final fuzzy map.

Ghazel et al. [**34**] proposed a semi-automated MS lesion detection and segmentation method based on optimal filter design for maximal feature selection and separation utilizing expert knowledge. The method is based on designing an optimized filter that aims for separating the texture features and energies of the two main classes assumed to be within the prescribed region of interests (ROI's), namely the MS lesions and the normal white matter. This has resulted in a system output where the MS lesions

have been transformed to brighter areas with increased intensity, while the background (healthy white matter tissues) has been transformed to a darker area. Consequently, the MS lesions and the healthy tissues become more distinguishable at the output of the system than they were originally. They may even be separated from one another by applying a thresholding operation.

Yamamoto et al. [6] proposed a false positive reduction scheme, which consisted of a rule-based method, a level set method, and a support vector machine. Identification of initial MS candidate regions is done by using an automated thresholding technique based on a linear discriminate analysis on the pixel-value histogram in the T1-weighted image. Next, a brain region was extracted by thresholding the pixel value for removing fat regions with high pixel values around the brain. In addition, a morphological opening operation with a circular structuring element of one pixel radius was also applied after thresholding. MS lesions were enhanced by subtraction of a background image approximated by the first order polynomial in a brain region from the FLAIR image. Then, an unsharp masking filter was applied to the subtraction image for enhancement of boundaries of MS lesions. The MS lesions were well enhanced in the subtraction image compared with the original FLAIR image. The initial candidates were identified by using a multiple-gray level thresholding technique on the subtraction image. MS initial candidate points were picked up according to the following three criteria: (1) Longest and shortest Euclidean distances between the centroids of a brain and each candidate region, because MS lesions would be developed within a certain distance from the centroid of the brain, (2) Minimum pixel value of MS candidates, because MS lesions have relatively higher pixel values compared with normal tissues in a brain parenchyma in the

subtraction image and (3) Minimum distance between candidate points so that one point with a higher pixel value can be selected as a final candidate point. MS candidate regions were segmented by using a region growing technique. Image features of MS candidate regions are determined. Next, false positive outliers were reduced using a rule-based method. In the rules of the contrast and the maximum pixel value, false positive outliers were removed by the minimum and maximum threshold of the feature values of all true positive regions. However, a different rule was employed for the distance feature value. Further reduction of false positives and determination of MS candidate regions is done based on a level set method. Finally, classification of MS candidate regions is done using a support vector machine (SVM). All candidate regions were classified into true positive and false positive candidate regions by using the SVM.

Freifeld et al. [35] proposed a probabilistic model termed Constrained Gaussian Mixture Model (CGMM) to capture the complex tissue spatial layout based on a mixture of multiple spatially oriented Gaussians per tissue. The intensity of a tissue is considered a global parameter and is constrained, by a parameter-tying scheme, to be the same value for the entire set of Gaussians that are related to the same tissue. MS lesions are identified as outlier Gaussian components and are grouped to form a new class in addition to the healthy tissue classes. A probability-based curve evolution technique is used to refine the delineation of lesion boundaries.

This category of segmentation frameworks is promising but lacks the incorporation of enough knowledge about the MS lesions to create robust model. More efforts are still needed to present mature techniques.

3.2 MRI Sequence Selection

In the acquisition of the MR images for the patient, many MRI sequences can be generated. Each of the MRI sequence provides information about the MS lesion at specific stage. Acute or enhanced lesions can be visualized on Gad-E-T1-w images, chronic lesions in T1-w images, and other lesions specified on FLAIR and T2-w images [36], [21]. Because of more inflammation, edema, and little demyelination, acute plaques appear with less signal changes in T1-w images. Consequently, these plaques, in T1-w images, compared with white matter, will be signal isointense or hypointense (less darkness). In addition, their borders are vague and cannot be marginated well. With progress in demyelination process and also chronicity of the disease, the plaques gradually become darker in T1-w images. More demyelination and gliosis (replacement of fibrous tissue instead of myelin and neuron) lead to sharpness of the lesions borders. This process results in the formation of some chronic plaques known as black holes. These plaques will appear as hyperintense areas in T2-w images and their signal intensity will not change in enhanced T1-w images [37].

According to the channels used, the techniques can be classified into single channel and multi-channels MS lesion segmentation techniques.

3.2.1 Single Channel

FLAIR image contains the most distinctive information for segmentation of white matter lesions. Since MS lesion tissue has comparable characteristics as white matter lesion, with respect to location and signal intensity, wide category of techniques chose to use only the FLAIR image in the MS segmentation process.

According to [20], Anbeek et al. use only the FLAIR image. The addition of other image types (T1, T2, FLAIR, proton density and T1-weighted inversion recovery) wasn't significantly beneficial for the segmentation, and sometimes had negative effect on the results. Although they use T1 image for creation of the brain mask, only the FLAIR image was involved in the classification stage. Also, Khayati et al. [36] used the FLAIR channel in his proposed work.

3.2.2 Multi Channels

Taking advantage of the various protocols that acquire images using multiple modalities is a current issue (typically T1, T2, PD, DTI or Flair sequences in MR neuroimaging). The data are becoming more and more multi-channel data and their unique and complementary information should be merged together before segmentation in order to get rid of the inconsistencies one can encounter when segmenting each modality separately. Today, reliable registration methods, using different resolution and time, are available; nevertheless, a simple, robust, fast and reliable segmentation approach still does not exist for such kind of problem especially when dealing with pathologies. Multichannel segmentation usually relies on clustering or classification.

In [17], three gray-level MRIs are merged into a single MRI to form a colored image to be processed by a colored edge detector. In [13] and [6], T1, T2, and FLAIR intensities are used in the feature vector separately. Zhu et al. [33], T1, T2, and PD images were treated as three independent information sources for the detection and segmentation of MS lesions and data fusion was applied.

3.3 Preprocessing steps

Image Processing Steps applied on the subject MRI to enhance the image quality, remove the effect of MRI acquisition noise and aid in feature extraction.

Studies have shown the usefulness of applying inhomogeneity correction exclusively to the IC portion of MR images [38], [39]. In [21], Wu et al include preprocessing unit for intra- and inter-scan intensity inhomogeneities and normalizes the observed scan intensities.

In [13], the inter-subject intensity variations were normalized. RF acquisition field inhomogeneities are corrected. Spatial prior is added by registering the MNI atlas to the anatomical images, each voxel of the atlas providing the probability of belonging to the white matter (WM), the grey matter (GM) and the cerebro-spinal fluid (CSF). These priors can be used as part of the feature vector.

In [20], a brain mask was created, indication the region of interest for the segmentation. This reduces the amount of voxels being processed, thus saving computer time and memory. The mask was created by applying the brain extraction tool [40] on the T1 image with a relatively high value for the fractional threshold. This procedure resulted in a narrow brain mask, consisting of brain tissue only. This narrow mask was observed empirically to perform well with the segmentation method.

In [21], Wu et al. proposed a procedure for intracranial cavity (IC) extraction. This IC extraction procedure combined non-parametric intensity-based statistical (Parzen window) segmentation and automated morphological operations. After the human-supervised Parzen window tissue sampling was calibrated one time on two test scans, the

Parzen window classifier was saved for automated segmentation of ICs on the remaining scans. Occasionally, minimal operator manual editing was conducted on the resulting IC masks. The IC masks were superimposed onto images to exclude extracranial tissues, skull and large vessels on the brain surface.

3.4 Features Extraction

Independent on the segmentation technique, the brain MRI voxels are represented by set of features. Good features are those which are most discriminative between the normal brain tissues and multiple sclerosis lesions. The features commonly used in the literature techniques may be categorized into intensity, spatial, neighborhood, shape, texture, atlas based priors and context defined features. Most of the techniques are using combination of features from the different categories.

Segmentation frameworks do feature selection either by using analysis or by using software tools. Software tools are used for features selection by defining pool of potentially informative features to draw from and analysis is done to select the most discriminative features [30].

3.4.1 Features Generated without Image Processing

These are the features generated by picking first order information from the input dataset without using any algorithms for feature extraction.

Intensity Features

This category of features is the basic features used by almost all of the techniques. The grayscale intensity for each voxel in the used MRI sequence can be used as a feature representing the voxel [20]. In case of using multi-channels, the grayscale intensity of the

voxel in each sequence can be used a part of the feature vector assuming that all channels are registered to a reference channel [13].

One of the problems of using intensity as a feature is the noise in MR images. To overcome the problem, other than the intensity correction techniques discussed in the previous section, the intensity can be averaged over a neighborhood (i.e. a cube) around the each voxel [13].

Spatial Features

Spatial features are commonly used to benefit from the common locations of multiple sclerosis lesions in the brain. The simplest features used in the feature vector are the three Cartesian coordinates [20], [24]. To use such features, registration should be done among the different dataset used in training and segmentation. Also, distance to the approximate center of the brain is used as a spatial feature [24].

Using spatial features adds inaccuracy to the segmentation due to the differences in size and location between the patients. This problem was avoided in [20] by normalizing the spatial features using variance scaling. In variance scaling, each spatial feature, feature, the mean of the feature values was subtracted from the voxel value, and the outcome was divided by the standard deviation. This approach resulted in a mean of 0 and variance of 1 for all features. This implicitly corrects for differences in size and location between the patients.

Neighborhood Features

This category of features is used to represent the local intensity pattern of the voxel of interest. Voxel intensity along with the surrounding voxels intensities is used. Neighborhood size varies according to the number of surrounding voxels involved in the neighborhood and whether the current slice or the neighboring slices are considered.

Shape Features

Curvature is used with different neighborhood size in [30]. In [6], initial candidate regions are initially selected. Processing is done later to classify these regions. A feature vector is calculated for each regios that includes five shape features. The five shape features were: the effective diameter, the circularity, the slenderness, periphery length and the Euclidean distance between a candidate and the centroid of a brain.

3.4.2 Features Generated with Image Processing

Several features derived directly or indirectly from the analysis of the visual characteristics of multiple sclerosis lesions in the MRI were introduced to provide better discrimination.

Atlas based Priors Features

When a standard atlas is registered to the anatomical images, spatial priors can be extracted and added to the feature vector. T1 sequence can be registered to the MNI atlas to get tissues probability priors [13] and [30]. These tissue probability priors are the probability given for each voxel of the atlas of belonging to the white matter, the grey matter and the cerebo-spinal fluid (CSF).

Spectral Gradient Features

In [**17**], based on the psycho-visual color theory, a scale-space approach was proposed to build a color-edge detector. This color edge detector is applied to MRI by merging three gray-level MRIs into a single MRI.

Context-rich features

This group of features was proposed by Geremia et al. [**13**] to use the similarity nature of the brain which is obvious in the axial view of the MRI. Context-rich features compare the voxel of interest with distance regions. As shown in Figure 3-1, the first context rich feature looks for relevant 3D boxes R1 and R2 to compare within an extended neighborhood. The regions R1 and R2 are sampled randomly in a large neighborhood of the voxel v. The sum over these regions is efficiently computed using integral volume processing. The second context rich feature compares the voxel of interest at x with its symmetric counterpart with respect to the mid-sagittal plane, noted S(x) Instead of comparing with the exact symmetric S(x) of the voxel, its 6, 26, and 32 neighbors in a sphere, centered on S(x) are considered respectively.

Texture Features

Textural analysis refers to a set of processes applied to characterize special variation patterns of voxels grayscale in an image. Segmentation based on texture properties is promising in cases of lesions that are inhomogeneous, not sharp, and faint, but show an intensity pattern that is different from the adjacent healthy tissue. In [**41**], Subanna et al. presented an automatic tissue classification scheme based on Markov Random Fields (MRF) that probabilistically models the local spatial relationships between voxels and

their neighbors. Their approach explicitly built distributions for lesions as separate tissue classes, as opposed to considering them as outliers. They adapted simulated annealing techniques to obtain the required MRF parameters. Standard texture features (provided in Chapter 2) were used by Zhang et al. [**42**]. In (Ghazel-2006), a feature extraction method based on optimal filter design is proposed that aims for producing output texture features corresponding to the MS lesions and healthy tissues background which are maximally separable. The good results for MS texture classification obtained by Zhang et al. motivated us to use the same features in the proposed work in our dissertation.
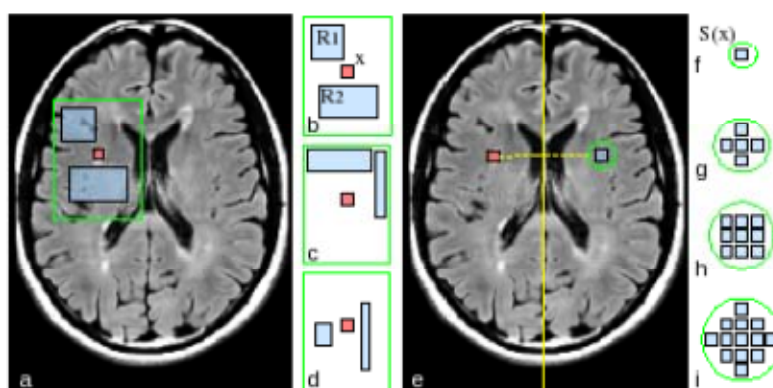
Figure 3-1: Context-rich features proposed by Geremia et al., 2010.

(a) A context-rich feature depicting two regions R1and R2 with constant offset relatively to x. (b-d) Three examples of randomly sampled features in an extended neighborhood. (e) The symmetric feature with respect to the mid-sagittal plane. (f) The hard symmetric constraint. (g-i) The soft symmetry feature considering neighboring voxels in a sphere of increasing radius. See text for details.

3.5 Pattern Recognition Algorithm

The core of the segmentation framework is the pattern recognition algorithm. For the MS Segmentation framework, any of the pattern recognition algorithms is adapted to fit the binary classification of the voxels into two classes: voxel which is part of MS lesion and

non-MS voxel. The pattern recognition technique is used to perform training with the training data sets' voxels labeled by the experts to get a discriminative model. The pattern recognition algorithm is again used by the model to perform the binary classification of the testing data sets' voxels. Based on the pattern recognition algorithm, the techniques in the literature can be classified into categories discussed in the following subsections.

3.5.1 K-Nearest Neighbor (KNN)

K-Nearest Neighbor classification [43], [44] is a statistical pattern recognition method, assigning samples (e.g. image voxels) to a class (e.g. MS-lesion) by searching for samples in a learning set with similar values in a predefined feature space. In this space each axis represents one of the voxel features. The learning set consists of pre-classified samples, which are added to the feature space according to their feature values. A new image voxel is classified by comparison with the K learning samples that are closest in terms of Euclidian distance (or any other defined feature distance) to it in the feature space. Commonly, the most frequent class among the K learning samples is assigned to this voxel. The choice of variable K in is dependent on the relation between the number of features and the number of cases. A small K will cause the result being influenced by individual cases, while a large value of K makes the classification outcome smoother. In general, for this type of problems a large K is favorable [45], [46]. The KNN is effective for multichannel MR data and particularly suitable for this three-channel segmentation. Previous studies also show that post-probability k-NN segmentation is more accurate and stable than segmentation approaches based on a priori statistical assumptions [47], [48].

In [20], Anbeek et al. used this technique for classification without assigning one class to the voxel, but with assigning probability per voxel being part of a lesion by taking the K learning samples into account. To get binary classification, threshold is applied on the probability of each voxel. They chose K = 40 by performing experiments on the training set with different K-values. The decision for this choice was made by visual inspection of the images in the training set. A larger K did not improve the results appreciably, but had a negative effect on the computational efficiency.

In [21], Wu et al. used KNN segmentation for the initial image intensity-based statistical classification of pre-processed three-channel image data into tissue classes. Seven classes are involved in the classification: T1 hyperintense enhancing lesions, T1 hypointense, cerebrospinal fluid (CSF)-like "black holes", T1 isointense but T2WI and PDWI hyperintense lesions ''other T2 lesions'', normal appearing white matter (NAWM), gray matter; CSF and background. K value was interactively selected by an expert neuroradiologist to maximize the classification accuracy on the basis of classification results. After the expert was satisfied with the classification on the two calibration scans, the classifier was saved as a master classifier. A k value of 3 was estimated.

3.5.2 Decision Forests

Discriminative random decision forest is an ensemble learner using decision trees as base learners. Decision trees are discriminative classifiers which are known to suffer from over-fitting. A random decision forest [49] achieves better generalization by growing an ensemble of many independent decision trees on a random subset of the training data and by randomizing the features made available to each node during training [7].

In [**13**], Geremia et al. used this technique within their segmentation framework. Training according to their implementation assumes that the forest has T components with t indexing each tree. The training data consists of a set of labeled voxels. When asked to classify a new image, the classifier aims to assign every voxel in the volume a label: 1 for MS lesion and 0 for non-MS. During training, all observations are pushed through each of the trees. Each internal node applies a binary test according based on visual features function. To perform the prediction, when applied to a new test data, each voxel is propagated through all the trees by successive application of the relevant binary tests. When reaching the leaf node in all trees, posteriors are gathered in order to compute the final posterior probability. This probability may be thresholded at a fixed value if a binary segmentation is required.

In [**14**], Akselrod-Ballin uses segmentation to obtain a hierarchical decomposition of a multichannel, anisotropic MR scans. It then produces a rich set of features describing the segments in terms of intensity, shape, location, neighborhood relations, and anatomical context. These features are then fed into a decision forest classifier, trained with data labeled by experts, enabling the detection of lesions at all scales. Unlike common approaches that use voxel-by-voxel analysis, this system can utilize regional properties that are often important for characterizing abnormal brain structures.

3.5.3 Principal Components Analysis

Principal component analysis (PCA) is a mathematical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of uncorrelated variables called principal components. This transformation is defined in such a way that the first principal component has as high a variance as possible

and each succeeding component in turn has the highest variance possible under the constraint that it be orthogonal to the preceding components. Principal components are guaranteed to be independent only if the data set is jointly normally distributed. It is mostly used as a tool in exploratory data analysis and for making predictive models.

In [24], Kroon et al. used PCS with log-liklihood to classify each voxel based on feature vector method that contains neighbourhood voxel intensities, histogram and MS probability atlas information.

3.5.4 Graph Cut

In the Graph Cuts framework, the image is represented by a weighted graph where each image voxel is represented by a node and each edge links the voxel to each of its neighboring voxels. The binary graph cut associates each node to one of two special nodes, called the "source" node and the "sink" node. These two nodes (the terminal nodes) represent the labels (i.e. "object" or "background").

In [17], Leceour et al. build a color MRI from three grey-level MRI sequences by assigning each red, green or blue channel to a sequence. The spectral gradient is computed and used it in a graph cut framework which requires seeds as input to obtain the segmented structures (e.g. brain, MS lesions). The object (O) and background (B) seeds are placed by the user at the beginning of the process are used to compute spectral intensity distribution models.

They improved this in [15] and [50] where seed source and sinks were automated using expectation minimization.

3.5.5 Support Vector Machine (SVM)

Support Vector Machines are supervised machine learning techniques, recently developed in the framework of statistical learning theory [51]. They have been used with a great success in a variety of applications such as text categorization, face recognition, and bioinformatics. In many of these areas SVM have outperformed well-established methods such as artificial neural networks, radial basis functions, and non-parametric cluster classification [52]. The great advantage of the SVM technique over conventional techniques, besides the well-established theoretical definition, is its capacity of working with high dimensional feature vectors without losing the generalization performance [12]. This property makes SVMs very suitable for segmentation of multispectral MR images.

In [23], Ferrari et al. used SVM with radial basis function (RBF) kernel. The kernel parameter (γ) and the penalty value for the errors (C) were determined by using a very loose stopping criterion for the SVM decomposition.

Yamamoto et al. [6] used a support vector machine (SVM) to classify MS candidate regions. All candidate regions were classified into true positive and false positive candidate regions by using the SVM.

3.5.6 Expectation Maximization

The expectation maximization (EM) algorithm is an iterative method for finding maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables. The EM iteration alternates between performing an expectation (E) step, which computes the expectation of the log-likelihood evaluated using the current estimate for the parameters,

and maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the E step. These parameter-estimates are then used to determine the distribution of the latent variables in the next E step.

According to the work of Pernod et al. [**18**], the expectation-maximization algorithm consists of two steps: labelization of the image (Expectation step) and estimation of the Gaussian class parameters (Maximization step). In this last step, the class parameters are computed from the intensities of the different voxels. In order to improve the algorithm speed, only a part of the image voxel can be taken into consideration thanks to a ratio parameter.

3.5.7 Outlier Detection

An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism [**53**].

In [**25**], Leemput et al. performs intensity-based tissue classification using a stochastic model for normal brain images and simultaneously detects MS lesions as outliers that are not well explained by the model. This approach circumvents explicit lesion modeling, which is difficult because of their widely varying appearance in MR images, and because not every individual scan contains a sufficient number of lesions for estimating the model parameters. The core of this method is a clustering algorithm that is made robust against model outliers. From an algorithmic point of view, this method bears close resemblance to an adaptation of the EM classifier described by Schroeter et al.[**54**], who iteratively classified normal brainMR images into a small number of Gaussian distributions, each time rejecting voxels that exceed a predefined Mahalanobis distance to

each of the Gaussians, and updating the model parameters only based on non rejected voxels. In contrast this method that either accepts or rejects voxels, the method proposed by Leemput et al. [25] uses a soft rejection scheme and also takes the classification of the voxels and their neighbors into account

In [26], Prastawa et al. combines outlier detection and region partitioning. This method is based on an atlas of healthy subjects and detects lesions as outliers, without requiring the use of training data with segmented lesions. In order to segment lesions as spatially coherent objects and avoid spurious lesion detection, they perform classification on regions (connected groups of voxels) instead of individual voxels. Each voxel location is assigned to a region that would maximize overall relative entropy divergence between neighboring regions.

In [27], [28] Bricq et al. proposes a method that performs tissue classification using a Hidden Markov Chain (HMC) model and detects MS lesions as outliers to the model. For this aim, they used the Trimmed Likelihood Estimator (TLE) to extract outliers. Furthermore, neighborhood information is included using the HMC model a priori information brought by a probabilistic atlas is incorporated.

3.6 Post processing steps

Image Processing and additional Pattern Recognition Steps applied on the segmentation result to improve the performance.

False Positive reduction is a common post processing step. Due to similarity between MS lesions and the white matter tissue, it is common to have huge number of

false positives. To improve the segmentation results, many of the techniques add a rule based stage to reduce the false positives.

In [**19**], Gracia-Lorenzo et al. applied lesion rules to discriminate between the white matter lesions and false positives. The rules were defined with neurologists and neuroradiologists based on image intensities from the respective MR sequences and voxel connectivity. Different intensity rules can be implemented for the different types of MS lesions: black holes, Gadolinium-enhanced lesions and T2-w lesions. Their focus was on the T2-w lesions that are, compared to the normal appearing WM, hyperintense in T2-w and FLAIR, and isointense or hypointense (e.g.black holes) in T1-w. Hyperintense and hypointense voxels are defined by $3.0 \times \sigma_{WM} \pm \mu_{WM}$, where σ_{WM} and μ_{WM} are the standard deviation and the mean of the white matter respectively. Voxel connectivity allows the use of neighboring rules instead of classifying each voxel independently. In this case, a minimal size of MS lesion is defined, so detected lesions that have a size smaller than 3 mm^3 were discarded. They also removed detected lesions that are contiguous to brain border or not contiguous to WM tissue.

In [**55**], a post-processing step is performed since many false positives occur due to artifacts in the external CSF, all lesions detected neighboring the brain border are removed from the segmentation.

In [**27**], to remove voxels which are not MS lesions that are detected outside MS lesions, especially in the CSF class, voxels for which the prior probability of CSF given by the atlas is higher than 0.5 were removed and lesions with a small volume (3mm^3) were excluded.

In [**35**], the segmentation approach usually results in an overestimation in the size of the lesions. This is most likely due to the fact that the area in the proximity of a lesion is somewhat different from the other tissues. In addition, the segmentation step model is done voxel wise and does not take into account the smoothness of the lesion boundaries. Hence, a post-processing step for refining the lesion boundary is needed. Active contour technique is proposed and applied as a post processing step for lesion delineation.

Lesion boundary refinement is also addressed in the post processing approaches. In [**21**], refining ''black holes'' segmentation ''Black holes'' identified by the main segmentation technique do not include areas of the white matter that are hypointense with respect to healthy white matter but isointense with respect to grey matter. Therefore, in the post processing step the classification of ''black holes'' is refined to include subtly hypointense signal. A more sensitive k-NN classifier is obtained by adding training points from mildly T1-hypointense WM regions and is selectively applied to lesion classes. One master classifier for this second segmentation step was generated and stored for k-NN segmentation of all the subjects.

3.7 Performance evaluation

To evaluate the segmentation techniques, comparison should be performed between the automatic segmentation (Seg) generated by the proposed approached and the manual segmentation provided by the expert neuroradiologist (Ref) as shown in Figure 3-2. Performance evaluation formulas are incorporated in the segmentation frameworks.
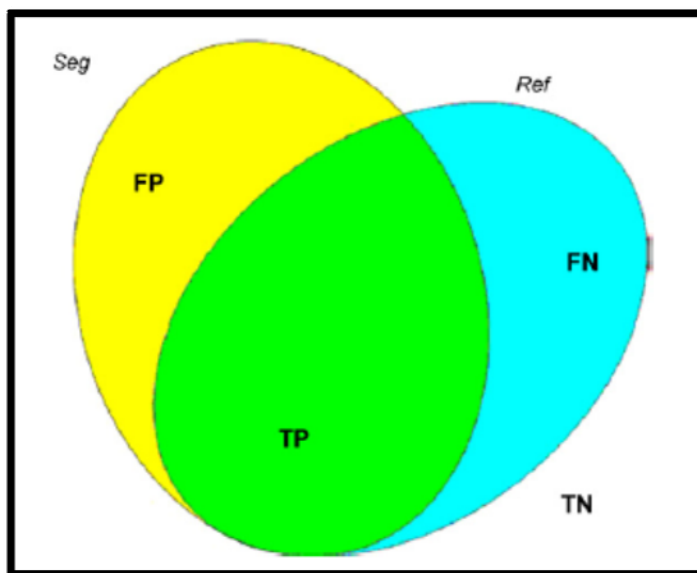
Figure 3-2: Comparison between the automatic segmentation (Seg) and the manual segmentation (Ref)

The voxels that are marked as MS in both Seg and Ref are the true positives (TP), voxels appears only in Seg are false positives (FP), voxels appears only in Ref are false negatives (FN) and voxels that are marked as non-MS in both sets are true negatives (TN). Based on these parameters, number of metrics is used in the literature to evaluate the segmentation performance. Dice similarity (Similarity Index), Tanimoto Overlap and Sensitivity are common metrics.

3.8 MS Lesions Segmentation Pipeline

The segmentation pipeline is method of integrating the components of the segmentation framework to achieve the segmentation task. Pipelines used in MS lesion segmentation techniques can be classified to single processing core, preprocessing-processing, processing-post processing, pre processing-processing-post processing and special pipelines.

3.8.1 One stage: Processing Core Pipelines

In this category of pipelines, the segmentation framework is composed mainly of a segmentation core that performs the segmentation without either improvement to the input quality or improvement to the output segmentation. Examples to segmentation pipelines in this category are those proposed by [28],[30] and [36]

3.8.2 Two stages: Pre processing and Processing Pipelines

In this category of pipelines, the segmentation framework is composed of a segmentation core that performs the segmentation proceeded by a preprocessing module that mainly provides improvement to the input quality. An example to segmentation pipelines in this category are those proposed by [13].

3.8.3 Two stages: Processing and Post processing Pipelines

In this category of pipelines, the segmentation framework is composed mainly of a segmentation core that performs the segmentation followed by a post processing module that mainly improves the quality of the output segmentation. The segmentation pipeline proposed by [35] is an example to a pipeline classified in this category.

3.8.4 Three stages: Pre processing, Processing, and Post processing Pipelines

In this category of pipelines, the segmentation framework is composed mainly of a segmentation core that performs the segmentation proceeded by a preprocessing module that mainly provides improvement to the input quality and followed by a post processing module that mainly improves the quality of the output segmentation. The segmentation pipelines proposed by [56] and [20] are classified in this category.

3.8.5 Special Pipelines

In this category of pipelines, the components and their integration are custom to the concept of the technique. Samples of these pipelines are discussed in this subsection.

In [**21**], the pipeline shown in Figure 3-3 is proposed for the automated three-channel segmentation. Dashed boxes indicate procedures performed only once and then used for all the subjects. Abbreviations used in the pipeline are: 3ch-MRI = three-channel MRI consisted of proton density-weighted image, T2-weighted image and post-contrast T1-weighted image; IC = intracranial cavity mask extraction; TDS$_+$ = template-driven segmentation plus partial volume artifact correction.
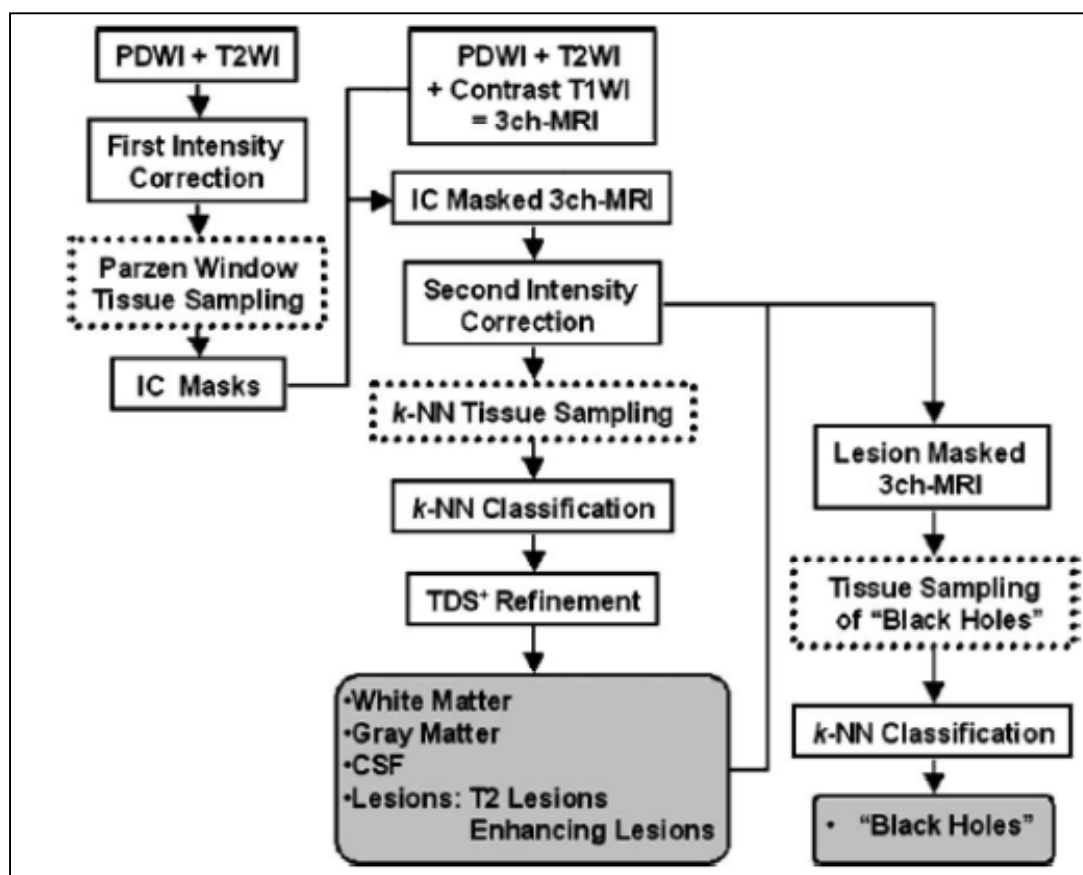
Figure 3-3: Customized pipeline proposed by Wu et al., 2006.

In [**17**], the following framework have been designed as follow: from three grey-level MRI sequences, a color MRI was built by assigning each red, green or blue channel to a sequence. Then the spectral gradient was computed and used in a graph cut framework which requires seeds as input. In the end of this framework, the segmented structures (e.g. brain, MS lesions) are generated. Figure 3-4 summarizes this pipeline.
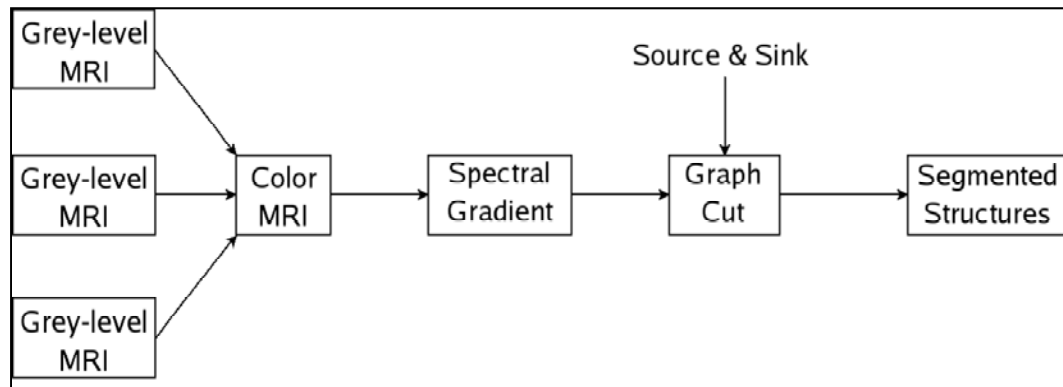
Figure 3-4: Customized pipeline proposed by Leceour et al., 2008.

3.9 Limitations and Open Areas

Each of the segmentation frameworks discussed in this chapter contributes to the problem of automatic segmentation of multiple sclerosis lesions in MRI in one of the following situations only:

1. In Preprocessing step: improving the properties of the MRI.
2. In Feature Extraction: extracting features that try to separate the MS voxels from non-MS voxels.
3. In Pattern Recognition: customizing a well known machine learning technique.
4. In Post-processing steps: rejection of false positives.

Most of the segmentation framework in literature deals with the MR images as stream of data, and the aim is to classify these data without deep study of the nature of the appearance of MS lesions in the different modalities of MRI and giving attention to common location and intensity behavior in the neighborhood of brain structures. More efforts are needed to provide a contextual based framework that tackles the MS lesion segmentation explicitly.

The following study is used as a base for our technique. In [**42**], Zhang et al. presented a comparative study that targets texture analysis of multiple sclerosis. Texture analysis was performed on MR images of MS patients and normal controls and a combined set of texture features were explored in order to better discriminate tissues between MS lesions, normal appearing white matter (NAWM), and normal white matter (NWM) in region of interests (ROIs). Features were extracted from gradient matrix, run-length (RL) matrix, gray level co-occurrence matrix (GLCM), autoregressive (AR) model, and wavelet analysis, and were selected based on greatest difference between different tissue types. The results of the combined set of texture features were compared with our previous results of GLCM-based features alone. The results of this study demonstrated that:

1. With the combined set of texture features, classification was perfect (100%) between MS lesions and NAWM (or NWM), less successful (88.89%) among the three tissue types and worst (58.33%) between NAWM and NWM.

2. Compared with GLCM-based features, the combined set of texture features were better at discriminating MS lesions and NWM, equally good at discriminating MS

lesions and NAWM and at all three tissue types, but less effective in classification between NAWM and NWM.

This study suggested that texture analysis with the combined set of texture features may be equally good or more advantageous than the commonly used GLCM-based features alone in discriminating MS lesions and NWM/NAWM and in supporting early diagnosis of MS.

They reported in future improvement the need to apply texture analysis with automatic selection of the ROIs rather than drawing ROIs manually.

To the best of our knowledge, texture based MS segmentation approaches that have been previously reported were applied to ROIs that are manually selected by an expert to indicate potential regions including MS lesions, which makes the segmentation process semi-automated. Therefore, efforts are needed to automate the use of textural features in the detection of MS lesions.

Also, MR images are processed in most of the algorithm in the axial view, where the other views sagittal and coronal can be used in assessment of the MS lesion obtained from automatic segmentation in axial view.

Utilizing textural features without the need for manual labeling of ROIs, enabling segmentation using multi-channels MRI data and taking into account the different sectional views of the lesion volume are the pillars of the proposed technique that is explained in detail in Chapter 4.

Chapter 4
Textural based SVM for MS Lesion Segmentation in MRIs

This chapter presents the approach developed in this dissertation to use the textural features (C_3) with a robust machine learning technique (C_4) to achieve multiple sclerosis (MS) lesions segmentation in the brain MR images. The approach is based on emulating the expert in labeling the MS lesions. The core modules of the segmentation framework are presented. Further progression in pipelines (C_7) based on these modules is subsequently presented with the motivation of enhancing the performance as well as supporting different problems in datasets.

4.1 Texture Analysis Open Area

Textural analysis refers to a set of processes applied to characterize special variation patterns of voxels grayscale in an image. Segmentation based on texture properties is promising in cases of lesions that are inhomogeneous, not sharp, and faint but show an intensity pattern that is different from the adjacent healthy tissue [**57**].

Texture based MS segmentation approaches that have been previously reported were applied to ROIs that are manually selected by an expert to indicate potential regions including MS lesions, which makes the segmentation process semi-automated. Therefore, efforts are needed to automate the use of textural features in the detection of MS lesions [**42**].

4.2 Performance Evaluation

To evaluate the performance of the automatic segmentation method, comparisons are performed with state of the art methods. Different metrics (C_6) are used to evaluate the performance for these literature techniques, therefore, according to the comparison needed; difference metrics will be calculated in the later sections that shows the results of the segmentation. Some of the performance evaluation metrics used in this dissertation are calculated as defined in the literature. Others are introduced to evaluate clinical valuable parameters. Besides, MS Lesion Challenge workshop [2] provides scores to compare the competitive techniques.

4.2.1 Dice Similarity (DS):
The dice similarity (DS) is a measure of the similarity between the manual segmentation (X) and the automatic segmentation (Y). The equation for the calculation can be written as:

$$DS = 2\,|X \cap Y| / (|X| + |Y|) \qquad (4\text{-}1)$$

As stated in [58] and [59], a DS score above 0.7 is generally considered as very good, especially when the segmented structures are small. In our evaluation, dice similarity is calculated based on the similarity of lesion regions which means that the number of common MS lesion regions between manual and automatic segmentation is used for $|X \cap Y|$ and the number of MS lesion regions of manual and automatic segmentation are used for $|X|$ and $|Y|$ respectively. In this context, the automatically segmented lesion region that shares at least one pixel with a manually segmented lesion region is

considered as a common MS lesion region since the number of MS lesion regions is more clinically relevant than the number of voxels [**60**], [**61**].

4.2.2 **Sensitivity**

Sensitivity is a measure of how many lesions are detected. It can be calculated as the percentage of true positive voxels to the total number of MS voxels in the ground truth. For the two sets manual segmentation (X) and automatic segmentation (Y), the sensitivity is calculated as:

$$\text{Sensitivity} = |X \cap Y| / |X| \tag{4-2}$$

4.2.3 **Detected Lesion Load (DLL):**

Detected lesion load is a measure of how much lesion volume is detected compared to the original lesion volume. The detected lesion volume takes into account all the positive lesions whether true or false. Having a percentage of detected lesion load close to 1.0 is clinically satisfactory since it provides a relatively accurate measure of the MS lesions volume. For the two sets manual segmentation (X) and automatic segmentation (Y), the Detected Lesion Load is calculated as:

$$\text{DLL} = |Y| / |X| \tag{4-3}$$

4.2.4 **True positive rate (TPR) and Positive Predictive Value (PPV)**

For the two sets manual segmentation (X) and automatic segmentation (Y), true positives (TP), false positives (FP) and false negatives (FN) can be calculated as:

$$TP = |X \cap Y|$$

$$FP = |Y - X|$$ (4-4)

$$FN = |X - Y|$$

The True positive rate (TPR) and Positive Predictive Value (PPV) are defined as:

$$TPR = TP / (TP + FN)$$

(4-4)

$$PPV = TP / (TP + FP)$$

4.2.5 MS Lesion Challenge Metrics and Score

The automated evaluation system used by the MS Lesion Challenge uses the volume difference (Volume diff.), average distance (Avg. Dist.), true positive rate (True Pos.) and false positive rate (False Pos.) to evaluate the segmentation. These metrics are defined in [2] as follows:

Volume Difference, in percent: The total absolute volume difference of the segmentation to the reference is divided by the total volume of the reference, in percent.

Average Distance, in millimeters: The border voxels of segmentation and reference are determined. These are defined as those voxels in the object that have at least one neighbor (of their 18 nearest neighbors) that does not belong to the object. For each voxel along one border, the closest voxel along the other border is determined (using unsigned Euclidean distance in real world distances, thus taking into account the different resolutions in the different scan directions). All these distances are stored, for border voxels from both reference and segmentation. The average of all these distances gives the averages symmetric absolute surface distance.

True Positive Rate, in percent: This is measured by dividing the number of lesions in the segmentation that overlap with a lesion in the reference segmentation with the number of overall lesions in the reference segmentation. This evaluates whether all lesions have been detected that are also in the reference segmentation.

False Positive Rate, in percent: This is measured by dividing the number of lesions in the segmentation that do not overlap with any lesion in the reference segmentation with the number of overall lesions in the segmentation. This rate represents whether any lesions are detected that are not in the reference.

All measures have been scored in relation to how the expert raters compare against each other. A score of 90 for any of the metric indicate a comparable performance with an expert rater. The overall score for each test case is an average of the score of the above four metrics calculated for two different raters. An overall score is an average for the scores of each individual test case.

4.3 Dataset

4.3.1 Synthetic Data
The simulated MRI datasets generated using the McGill University BrainWeb MRI Simulator ([**1**], [**62**], [**58**], [**63**], [**59**]) include three brain MRI datasets with mild, moderate and severe levels of multiple sclerosis lesions. We will refer to these templates in this dissertation, including the results, as MSLES 1, MSLES 2, and MSLES 3 for the mild, moderate, and severe levels, respectively. The MRI data was generated using T1, T2, and Inversion Recovery (IR) channels. Isotropic voxel size of 1mm x 1mm x 1mm

and spatial in-homogeneity of 0% are used in this dissertation. For each channel, the images are available at six different noise levels (0%, 1%, 3%, 5%, 7%, and 9%).

4.3.2 Real Data

Datasets of 61 cases are used to verify the segmentation technique proposed by this dissertation. The sources of these datasets are the workshop of MS Lesion Segmentation Challenge 2008 ([2], [64]) and real MRI studies for MS subjects acquired at the University of Miami.

MS Lesion Segmentation Challenge 2008

Datasets used for evaluation in this dissertation include 51 cases which are publicly available from the MS Lesion Segmentation Challenge 2008 website [64]. For each case, three MR channels are made available (T1-,T2-weighted, and Flair). The datasets are divided into labeled cases used for training (20 cases) and non-labeled cases used for testing (31 cases). The MRI datasets are from two separate sources: 28 datasets (10 for training and 18 for testing) from Children's Hospital Boston (CHB) and 23 datasets (10 for training and 13 for testing) from University of North Carolina (UNC). The UNC cases were acquired on a Siemens 3T Allegra MRI scanner with slice thickness of 1 mm and in-plane resolution of 0.5 mm. No scanner information was provided about the CHB cases. All subjects MRI are re-sliced to be 512x512x512 with resolution 0.5mm x 0.5 mm x 0.5mm. We will refer to training datasets of CHB and UNC as CHB_train_CaseXX and UNC_train_CaseXX respectively (with XX refers to study number in two digits). Smilarly the test datasets of CHB and UNC are referred to by CHB_test1_CaseXX and UNC_test1_CaseXX respectively.

MS Subjects MRI Data acquired at University of Miami

The acquired MRI datasets for MS subjects are composed of multi-channel MRI, including T1, T2, PD, and FLAIR, for 10 subjects (4 males, age range: 50-72 and 6 females, age range: 30-59). The corresponding volumes in the different sequences are co-centered and have the same field of view of 175x220 mm. The slice thickness and spacing between slices for T2, PD, and FLAIR sequences are 3mm and 3.9 mm, respectively, while both of the slice thickness and spacing between slices for the T1 sequence is 1mm, respectively, for the same field of view. On average, each T2 and FLAIR MRI sequence consists of thirty seven slices while the T1 MRI sequence consists of one hundred and sixty slices that cover the whole brain. The axial FLAIR sequences used in this dissertation were acquired using the following imaging parameters: $9000/103/2500/256 \times 204/17/123$ (repetition time ms/echo time ms/inversion time/matrix size/echo train length/ imaging frequency). The parameters for axial T1 sequences: $2150/3.4/256 \times 208/1/123$ (repetition time ms/echo time ms /matrix size/echo train length/ imaging frequency), while the parameters for axial T2 sequences: $6530/120/256 \times 204/11/123$ (repetition time ms/echo time ms /matrix size/echo train length/ imaging frequency). The parameters for axial PD sequences: $6530/10/256 \times 204/11/123$ (repetition time ms/echo time ms /matrix size/echo train length/ imaging frequency). All the subjects were referred for brain MRI studies based on an earlier diagnosis of MS. The MS lesions were manually labeled on the FLAIR sequences by a neuro-radiologist. The ten MRI studies were acquired using a 3.0T MR scanner under a human subject's protocol approved by our institutional review board. We will refer to these subjects as MSX (with X is an integer number from 2 to 11).

4.4 Segmentation Framework

The segmentation framework for the proposed technique is based on an SVM fed by feature vector consists mainly of textural feature. We refer to this core engine by textural based SVM. The initial pipeline was designed to deal with single channel MRI mainly the FLAIR sequence (pipeline 1). Then, we aimed to benefit from the complementary information in the multi-channels MRI and the tissues probabilities obtained when registered to spatial atlas. Hence, the pipeline 1 was developed to incorporate these functions and we obtained pipeline 2. Subsequently, we introduced the concept of multi-sectional views segmentation, where each brain sectional views' slices are segmented individually and aggregate function is applied to get more accurate segmentation. We developed pipeline 3 to incorporate the multi-view segmentation. In the following sections, the three pipelines are discussed and followed by an experimental results section.

4.5 Pipeline 1 - Single View and Single-Channel MRI (FLAIR) Pipeline

In this pipeline, the MRI FLAIR slices of the brain are preprocessed for intensity correction to remove the effect of noise and differences in brightness and contrast between different scans of different subjects. The next step is the main processing module which is used for the detection of initial MS lesions regions based on textural features. This stage generates scores for each voxel in the slices that represents the probability of being MS voxel or not. The connected voxels having non-zero scores form regions of potential MS lesions. The post processing step involves addressing false positive MS regions based on location attributes and detecting false negative MS regions through inter-slice comparisons using the 3D nature of the MRI. After that, the post processing

step improves the MS lesions for each slice by removing voxels based on distance and grayscale features, and including neighboring voxels using region growing based on grayscale features and including voxels by removing holes in lesion regions based on the concept of lesion continuity. The pipeline is shown in Figure 4-1.
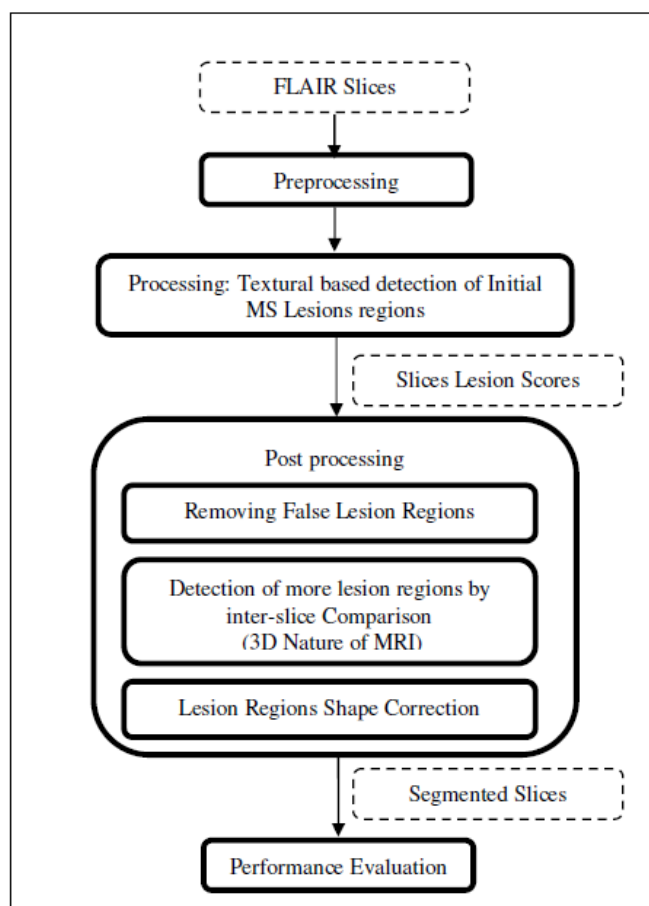
Figure 4-1: Single View Pipeline – Single Channel MS Segmentation Pipeline (Pipeline 1).

The modules of this pipeline are explained in details in the following subsections.

4.5.1 Preprocessing

This Module involves intensity correction of the subject's MRI data to improve the quality of the input FLAIR slices prior to texture extraction.

Intensity Correction:

Due to different operating conditions, brightness and contrast of the imaging slices may vary among subjects. This affects performance of segmentation that is based on textural features which are calculated based on grayscale intensities. If a dataset is used for training, better histogram matching of the dataset to be segmented and the training data will lead to more accurate definition of MS lesions. We used our preprocessing technique used before in [**65**] that starts with applying contrast-brightness correction to maximize the intersection between the histogram of the training and segmentation datasets followed by using 3D anisotropic filter to eliminate empty histogram bins. Figure 4-2 shows the effect of the preprocessing technique on improving a FLAIR slice from MS6 (subject dataset to be segmented) with reference to MS3 (subject dataset used in training).
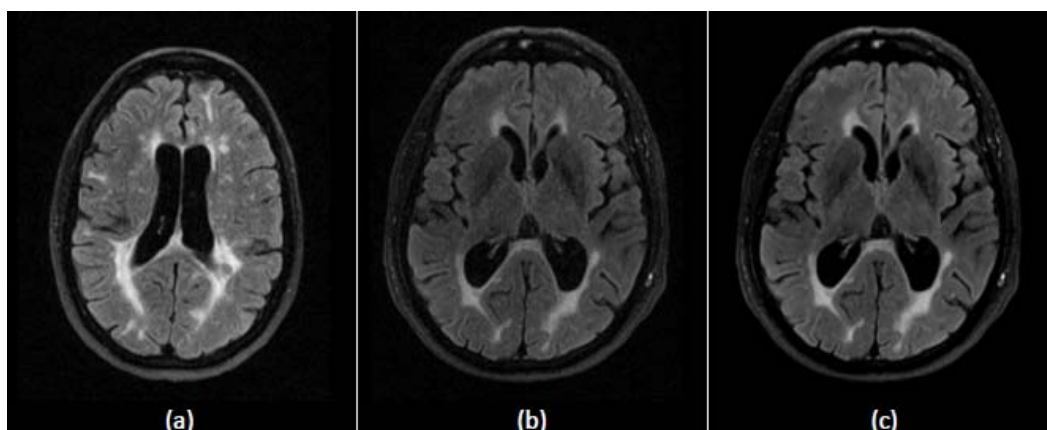
Figure 4-2: Intensity correction for FLAIR Sequence.

(a) A slice from the reference subject MS3 (used in training), (b) a slice from subject MS6 before preprocessing and (c) the same slice of MS6 after preprocessing.

4.5.2 Processing Core Module: Single Sectional View Textural Based SVM

Each preprocessed MRI axial FLAIR slice is processed through a trained detector engine to obtain potential MS lesions regions. The detector engine in our method is implemented

using support vector machine. Training the detector engine is done by processing the training dataset and dividing its slices into square blocks and assigning a binary class for each block. If the block contains at least one pixel manually labeled as MS, it is classified as MS block (class 1). Otherwise, the block is classified as non-MS block (class 0) if all of its pixels are labeled as non-MS pixels. Each block is described by a feature vector which mainly represents textural features of the block. During segmentation, the slice to be segmented is divided into square overlapping blocks and each block is classified by the trained engine as MS block or non-MS block.

Block Size

Statistics were previously made to measure the size of the multiple sclerosis lesions. The common values for the diameter are between 3.5 mm and 13.5mm [**66**]. For input MRI studies, the size of the square blocks w$\times$w pixel^2 is selected automatically to be within the range of (4$\times$4 mm^2) which tightly covers the smallest possible MS lesion diameter based on the input dataset pixel size which is determined from the input resolution and field of view.

Feature Vector

In order to describe each square block of the MRI slice, a features vector of is calculated. The block features are classified into three categories: textural features, position features, and neighboring blocks features. The features vector is listed in Table 4-1.

Textural Features

The textural features are calculated for the FLAIR sequence (C_1). Textural features include histogram-based features (mean and Variance), gradient-based features (gradient mean and gradient Variance), run length-based features (gray level non-uniformity, run length non-uniformity) and co-occurrence matrix-based features (contrast, entropy and absolute value). Run length-based features are calculated 4 times for horizontal, vertical, 45 degrees and 135 degrees directions. Co-occurrence matrix-based features are calculated using a pixel distance d=1 and for the same angles as the run length-based features. The details for calculating the textural-based features are provided in Chapter 2. The motives for using these features specifically for statistical texture are the recommendation for features provided in the study of Zhang et al. [42] to describe the MS texture as well as our feature selection study done for a pool of hundreds of textural features extracted from healthy and lesion blocks.

Position Features

The position features are the slice relative location with reference to the bottom slice and the radial Euclidean distance between the block's top left pixel and the center of the slice normalized by dividing it by the longest diameter of the slice as shown in Figure 4-3. The center and the longest diameter of the slice are parameters that are geometrically calculated in the preprocessing step.

The neighboring blocks features are the difference between the mean intensity of the current block and the mean intensity of each of the eight neighboring blocks as shown in Figure 4-4.
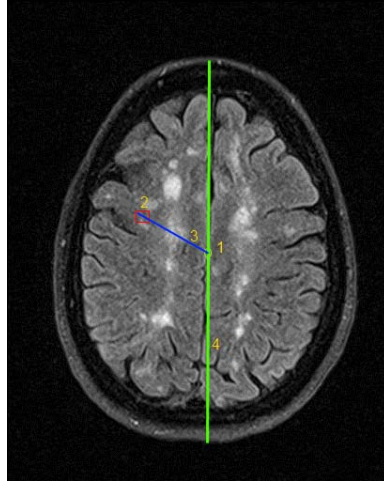
Figure 4-3: Position Features Extraction on a Sample Slice.

Calculation of the normalized radial distance between block (2) and slice center (1) (length of line 3 / length of line 4).

Neighboring Features
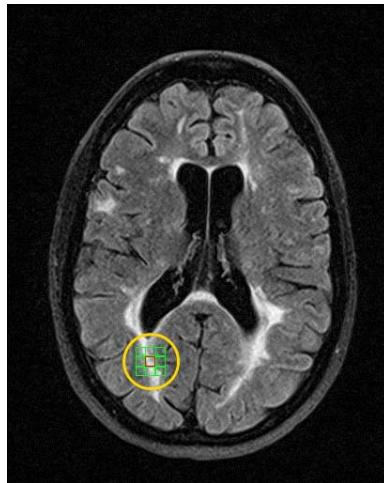
Figure 4-4: Neighboring Blocks Features on a Sample Slice

Calculation of the eight neighboring blocks features (difference between mean grayscale of the centered red block and mean grayscale of each of the eight neighboring green blocks in the grid circled by the yellow circle).

SVM training and segmentation

Support Vector Machine (SVM) is a supervised learning algorithm, which has at its core a method for creating a predictor function from a set of training data where the function itself can be a binary, a multi-category, or even a general regression predictor. To accomplish this mathematical feat, SVMs find a hypersurface which attempts to split the positive and negative examples with the largest possible margin on all sides of the hyperplane. It uses a kernel function to transform data from input space into a high dimensional feature space in which it searches for a separating hyperplane.

Features Category	Features
Textural features	1-Mean (Histogram based feature) 2-variance (Histogram based feature) 3-Gradient mean (Gradient based feature) 4- Gradient variance (Gradient based feature) 5-8 Grey level non-uniformality in the 4 directions (Run length based feature) 9-12 Run length non-uniformality in the 4 directions (Run length based feature) 13-16 Contrast in the 4 directions (Co-occurrence matrix based feature) 17-20 Absolute value in the 4 directions (Co-occurrence matrix based feature) 21-24 Entropy in the 4 directions (Co-occurrence matrix based feature)
Position features	25-Slice relative location 26- Normalized radial distance between block and slice center
Neighboring blocks features	27-34 Differences between grayscale of the block and each of the 8 neighboring blocks

Table 4-1 : A square Block Features Vector in Single-View Single Channel Pipeline.

The radial basis function (RBF) kernel is selected to be the kernel of the SVM. This kernel nonlinearly maps samples into a higher dimensional space so it can handle the case when the relation between class labels and attributes is nonlinear. The library libsvm 2.9

[**67**] includes all the methods needed to do the implementation, training and prediction tasks of the SVM. It is incorporated in our method to handle all the SVM operations.

Training

The dataset of one subject (or more) is used to generate the SVM training set. In the selected brain view, the slices of this training dataset are divided into n square blocks of size w𝐱w pixels. SVM training set (T) is composed of training entries t_i (x_i, y_i) where x_i is the features vector of the block b_i, y_i is the class label of this block for i =1 to n (n is the number of blocks included in the training set). The segmentation of MS lesions amounts to a binary classification problem, i.e., y_i is either 0 or 1. The training entry is said to be positive entry if y_i is 1 and negative in the other case.

For each slice of the training dataset, each group of connected pixels labeled manually as MS pixels forms a lesion region. Blocks involved in the set of positive training entries (T_P) are generated by localizing all the lesion regions and for each of them, the smallest rectangle that encloses the lesion region is divided into non-overlapping square blocks of size w𝐱w pixels. Each block b_i of these blocks is labeled by y_i=1 if any of the w^2 pixels inside this block is manually labeled as MS pixel. Any block that contains at least 1 MS pixel is referred to in our method as MS block. Similarly, the blocks involved in the set of negative training entries (T_N) are generated by localizing the non-background lesion regions whose pixels are not manually labeled as MS pixels and dividing them into non-overlapping square blocks of size w𝐱w pixels. These blocks are referred to in our method as non-MS blocks. Each block b_i of these blocks is labeled by y_i=0. Feature vector x_i is calculated for each block of both positive and negative training

entries. The positive training entries T_p contain blocks that contain 1 to w^2 MS pixels. This helps the SVM engine to learn the features of the blocks that either partially or completely contain MS pixels. The training set T is composed of the positive training entries and the negative training entries: $T = T_P \cup T_N$. The training set entries were fed to the SVM engine to generate a MS classifier which is able to classify any square w**x**w block of a brain MRI slice as MS block (y=1) or non-MS block (y=0) based on its features vector (x).

Training the classifier with more than one subject dataset allow the classifier to learn broad ranges for the features of real MS-blocks. However, due to computation time, training using many datasets is supported with each contributing only with a specified percentage to the overall training set. Usually the amount of MS blocks is much lower than the non-MS blocks. Hence, the share of blocks from each single dataset in the training set includes all MS blocks and the share is completed by selecting randomly non-MS blocks that cover all brain areas. One of the datasets will be the reference for intensity correction, while the other datasets are intensity corrected according to the reference dataset before adding their shares to the training set to maintain one intensity reference that can be used for the datasets to be segmented.

SVM Parameters selection directly affects the classification performance. The SVM penalty parameter C and the RBF kernel parameter γ are chosen via a grid search using cross validation as proposed in [**67**]. In cross validation, the training dataset are divided into subsets. Sequentially one subset is tested using the classifier trained on the remaining subsets. Cross validation accuracy is the percentage of data which are correctly classified. Various pairs of (C; γ) values are tried in the cross validation tests and the one

with the best cross validation accuracy is picked. We applied the cross validation on the training subject (MS3) dataset. Using this offline exhaustive search, C and γ that provided best accuracy in the cross validation for our technique are 1 and 0.029 respectively.

On the other hand, the training set used in training the SVM is highly imbalanced. The size of the negative training entries (TN) is much higher than the positive training entries (TP) due to the relative size of lesion with respect to the normal brain tissues. This affects the performance of SVM. However traditional approaches to overcome imbalanced data involve either over-sample the minority class (MS blocks) or under-sample the majority class (non-MS blocks). The first results in a distribution that no longer approximates the target distribution and the later results in discarding instances that may contain valuable information. Our decision was to leave the data with neither over-sampling nor under-sampling to avoid biasing the classifier and to keep the real distribution. Future improvement should address balancing the dataset with no added inaccuracy.

Segmentation

Each of the axial FLAIR slices of the subject dataset to be segmented is divided into overlapping square blocks of size w**x**w pixels depending in the computation time constraints. The features vector for each block is calculated. The trained SVM for the corresponding brain view is used to predict the class labels for all the blocks. For any block classified as MS block, assuming true positive classification, this does not mean

that all pixels of the block should be classified as MS pixels because the SVM engine is trained to detect the blocks that contains MS pixels completely or partially.

For each slice, all pixels are assigned an integer score. This score is initialized with a zero value. During segmentation, if any block is classified as MS block (y=1), the scores of all pixels inside the block are incremented. As the blocks are overlapped, each pixel is part of w^2 blocks as demonstrated in Figure 4-5. Thus, the score will be any value from 0 to w^2.
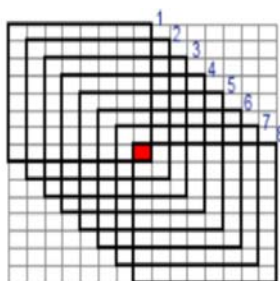
Figure 4-5: All possible overlapping blocks that contain a pixel.

for 8x8 blocks (w=8), the red pixel is part of w^2=64 blocks. The eight bold blocks are samples where the red pixel lies in the coordinates (8,8) of block 1, (7,7) of block 2 … and (1,1) of block 8.

After classification, these scores act as initial lesion probability maps where a large score indicates high probability for the pixel to be an MS pixel. Initial MS lesions can be generated by assigning any pixel of non-zero score as MS pixel.

Figure 4-6 shows segmentation of sample slice from subject (MS6) in axial views. Figure 4-6(a) shows the preprocessed FLAIR slice. Figure 4-6(b) is the ground truth for the lesions generated through manual segmentation. Figure 4-6(c) is the initial segmentation by considering any pixel of score higher than zero as MS pixel. Figure 4-6(d) provides a colored evaluation of the segmentation where the true positive

pixels are marked by blue, false positive are marked by red, false negatives are marked by green and true negatives are the background pixels.
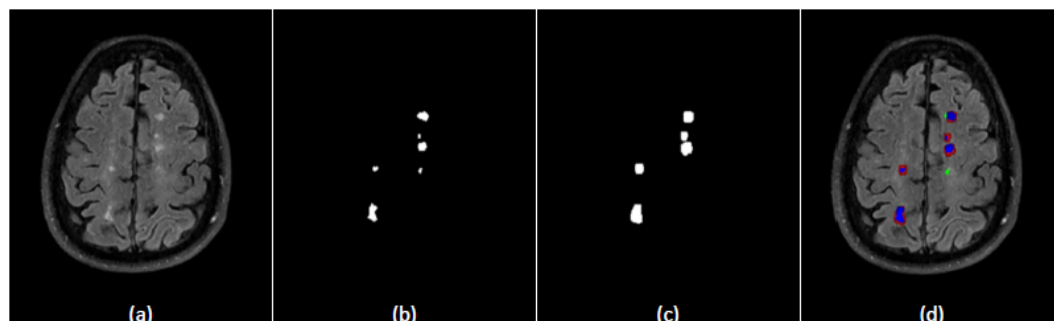
Figure 4-6: Using Textural SVM in Single Channel and Single View.
(a) preprocessed FLAIR axial slice from MS6, (b) ground truth, (c) initial segmentation and (d) colored evaluation of segmentation.

4.5.3 Post-Processing Module

The purpose of the post processing step is to improve and refine the performance of initial segmentation through dealing with different types of errors (false positives and false negatives). Figure 4-7 shows the initial segmentation of a sample slice from subject MS5 and the colored evaluation of the segmentation in which the false negatives and positives are marked in green and red colors, respectively. Errors in the initial segmentation of MS lesions can be classified as:

- Type 1: False negatives resulting from not detecting MS lesion regions (labeled by 1 in Figure 4-7 (b)).

- Type 2: False negatives resulting from incomplete MS lesion regions (labeled by 2 in Figure 4-7 (b)).

- Type 3: False positives resulting from false MS lesion regions (labeled by 3 in Figure 4-7 (b)).

- Type 4: False positives resulting from false portions of true MS lesion regions (labeled by 4 in Figure 4-7 (b)).

The following steps, which constitute the post processing of the MS lesion segmentation framework, are a set of logical operations that aim to address the different types of errors in the initial segmentation of MS lesions without adding new errors. The subject dataset used in training (MS3) was segmented by the textural SVM to get the initial segmentation which was analyzed for the different errors in MS lesions segmentation to formulate the criteria and thresholds used in post processing. This is summarized in the block diagram shown in Figure 4-8
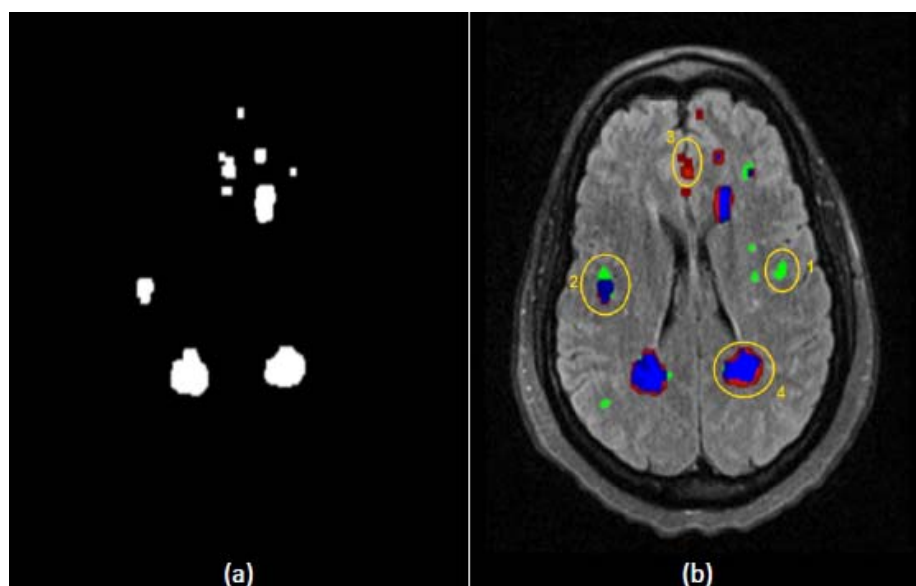
Figure 4-7: False negatives and positives in the textural segmentation.

(a) initial segmentation of a slice from MS5 and (b) colored evaluation of the initial segmentation where the different types of errors are labeled by a number matching the corresponding error type.
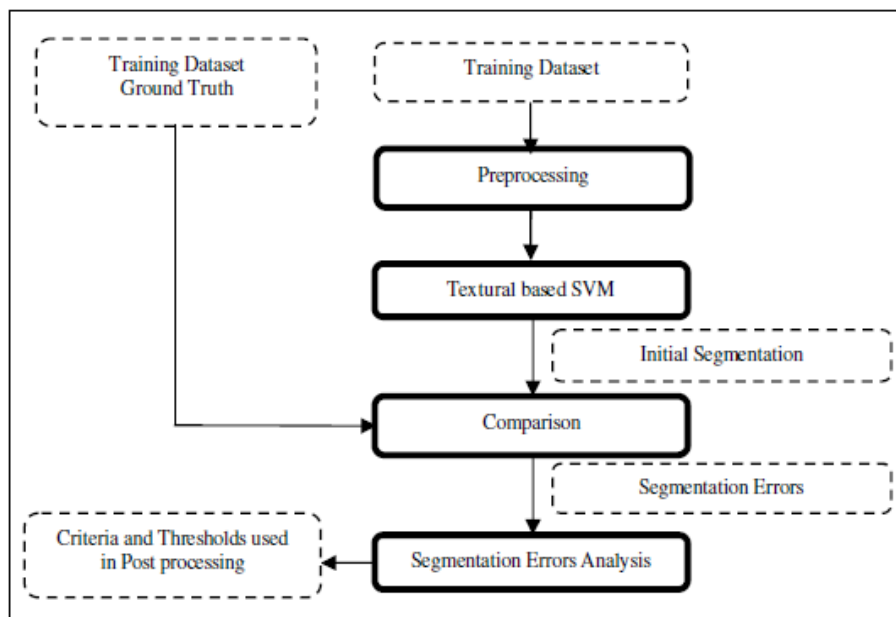
Figure 4-8: Formulation of Criteria and Thresholds used in Post processing.

Elimination of false positives resulting from lesion regions in uncommon locations

In this step, errors of type 3 in MS lesions are addressed. This type of errors in MS lesions results from detected MS lesion regions which are completely false. Some of these MS lesion regions that are located in uncommon locations can be eliminated. Odd locations include MS lesions outside the brain area, close to the brain boundary, or close to the sagittal plan [27].

In Figure 4-9, step 1 of post processing is applied to the initial segmentation of a slice from subject MS6, shown in Figure 4-9(a) and color evaluated in Figure 4-9(b), to eliminate the erroneous MS lesion regions circled in yellow circle as they are located so close to the boundary of the slice. The same slice after applying step 1 of post processing is shown in (Figure 4-9(c), Figure 4-9(d)). The same results were obtained for this slice by deletion based on location or by using the following SVM based Deleter.
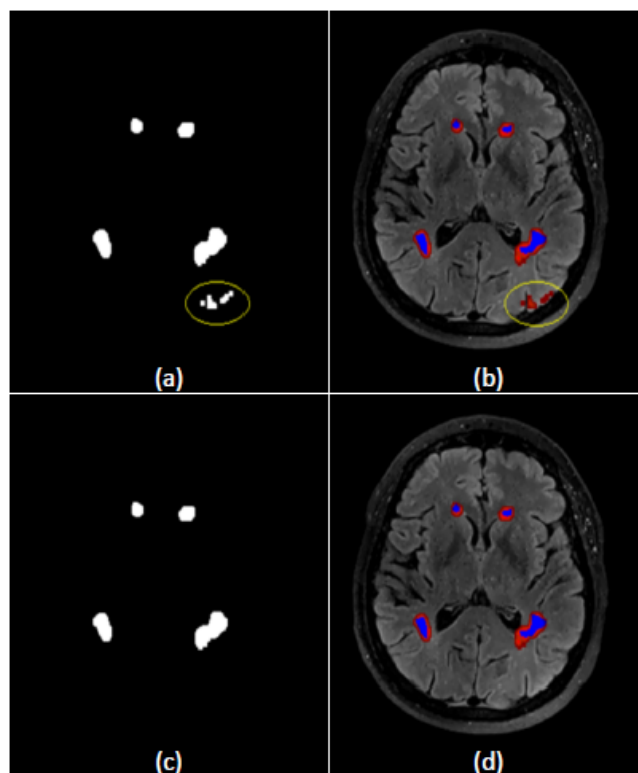
Figure 4-9: Post processing Step1: Elimination of segmented lesion regions in odd locations.

(a) and (b) are the slice segmentation and colored evaluation of the segmentation before applying step 1 (MS lesion regions in odd locations are circled in yellow). (c) and (d) are the slice segmentation and colored evaluation of the segmentation after applying step1.

Detection of non-detected MS lesion regions (false negatives)

In this step, errors of type 1 in MS lesions are addressed. This type of errors in MS lesions results from not detecting the lesion region, i.e., completely missing it. According to [68], in most cases the MS lesions extend only into one to three consecutive slices when the thickness of the slices is 3 mm. Therefore, In order to recover the missing MS lesion regions the initial segmentations of the previous and the next slices (or neighboring slices) are considered. The detected MS lesion regions in the previous and next slices are intersected based on the common coordinates on both slices generating a new slice of MS

lesion regions. Any pixel in the MS lesion regions resulting from the intersection is assigned the average of the segmentation scores of the two pixels located at the same slice local coordinates in the intersecting slices. Each intersection lesion region is assigned a score which is the average of the scores of the pixels in the lesion region. Because of the 3D nature of the MRI slices and the fact that the lesion occupies a volume, the lesion regions generated from the intersection should be highly correlated to the lesion regions in the original slice (the slice between the intersecting slices) especially if the generated lesions are of high scores. If this intersection leads to new regions in the current slice that have high scores, there will be a high probability that these new lesion regions are part of non-detected lesions in the current slice, and they should be added as initial segmentations. In Figure 4-10, step 2 of post processing is applied to the initial segmentation of a slice from subject MS6. Figure 4-10(a) shows slice 12 and the circled green lesion is a sample for a completely non-detected MS lesion region. Figure 4-10 (b) and Figure 4-10 (c) show slices 11 and 13, respectively. The circled lesion regions in both of them are detected lesion regions in the textural segmentation step. When these lesion regions are intersected as shown in Figure 4-10(d), they recover part of the non-detected lesion region in slice 12 as shown in Figure 4-10(e). The recovered part of lesion region can act as a seed that can be expanded in the region growing used as part of post processing step 3.
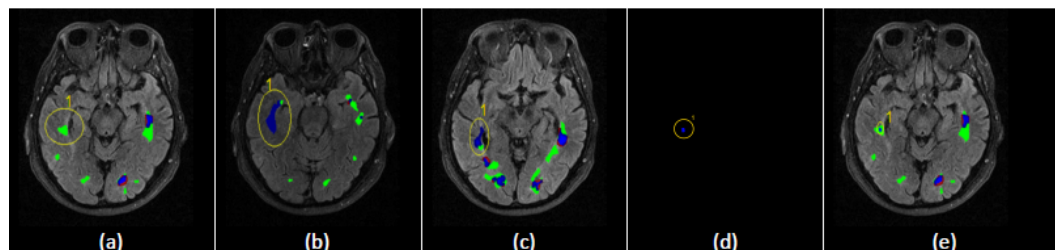
Figure 4-10: Post processing Step2: Detection of the non-detected lesion regions using neighboring slices.

(a) non detected lesion in slice 12 of MS6. (b) and (c) detected lesions in the slices 11 and 13 respectively are intersected to recover part of the non detected lesion region. (d) Intersection provides part a new lesion region (e) Portion of the lesion region in the slice 12 is recovered by adding (a) and (d).

Step 3: Lesion Regions Shape Correction

In this step, errors of type 2 and 4 in MS lesions are addressed. These two types of errors represent false parts in the segmented MS lesion regions in the form of either additional parts that need to be removed or incomplete parts that need to be detected. Both types of errors are addressed through shape correction of each segmented MS lesion region without adding or deleting MS lesion regions. Each detected MS lesion region in each slice is processed to correct its shape through the elimination of false positive pixels (type 4) and adding non-detected or false negative pixels (type 2). The shape correction of detected MS lesions is performed through the following three operations. Figure 4-12 will be used to illustrate the application of the different operations in post processing step 3 to a sample segmented slice from MS6. Figure 4-12 (i1) and Figure 4-12(i2) show the initial colored evaluation of the segmentation and the initial segmentation, respectively. The other parts of Figure 4-12 will be used to illustrate the corresponding operations in the following discussion of step 3.

Step A - Elimination of the false positives on the boundary of detected MS lesion regions

The colored evaluation of the segmentation depicted in Figure 4-12(i1) shows that each lesion regions colored in blue area (true positive) is surrounded by a red boundary (false positive). These false positive pixels on the boundary of the lesion regions may arise from the similarity between the textural properties (features) of non-MS regions and MS lesion regions. These false positive pixels may also arise from blocks classified in the initial segmentation as MS blocks which will cause all the block pixels' scores to be incremented increasing their probability to be MS-pixel even when some of the pixels in the block are not MS pixels.

To eliminate these false positive pixels, especially on the boundary, the following parameters are considered for each pixel:

1. Euclidean distance between the pixel and the lesion region boundary.
2. Difference between the grayscale of the pixel and the mean grayscale of the lesion region.
3. Segmentation score of the pixel at the conclusion of the initial segmentation step.

Some of the pixels of the non-MS pixels on the boundary of lesion regions can be eliminated based on these parameters using a fuzzy engine designed for this purpose where the membership functions and the threshold values are calculated based on the analysis of the initial segmentation of the training subject dataset (MS3). A Summary of the fuzzy engine including variable fuzzification, fuzzy rules, and defuzzification is provided in Figure 4-11. The fuzzy rules output is the decision which is binary variable

have two values; either keep the pixel in the lesion region (KEEP) or remove it (REMOVE). The defuzzification is performed using the centroid rule which is used in case of classification [**69**].
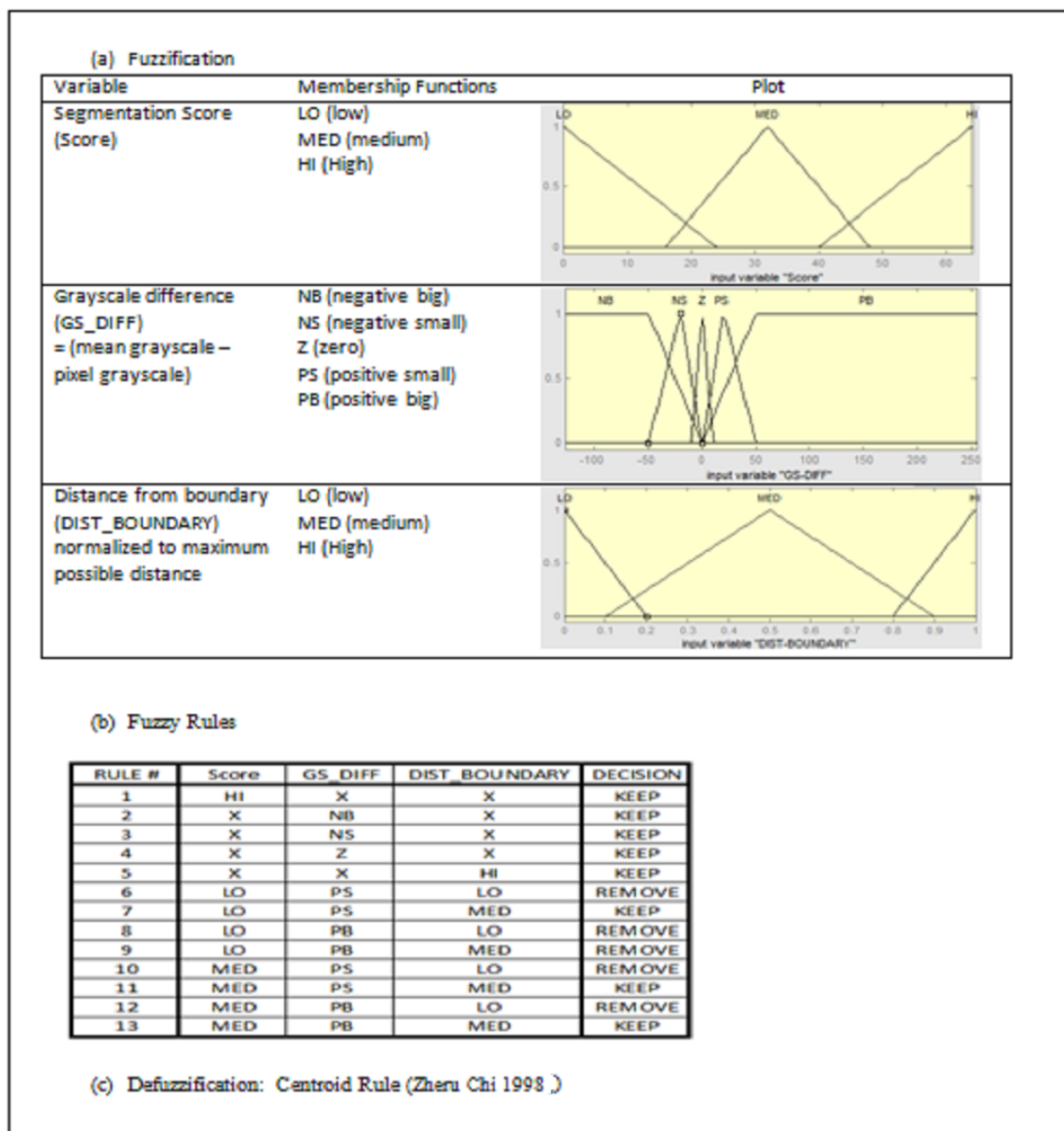
Figure 4-11: Fuzzy Engine used in Lesion Regions Shape Correction (Step 3-a).

Variables fuzzification, fuzzy rules, and defuzzification. (In Fuzzy rules, X indicates don't care condition and DECISION=KEEP means keep the pixel in the MS lesion region and DECISION=REMOVE means remove the pixel from the MS lesion region).

Figure 4-12 (a1) and Figure 4-12 (a2) show the effect of applying this step (A) on a slice from MS6 in case of using the fuzzy based engine. The lesion regions became smoother after trimming the false positive pixels on the boundary, but some pixels inside the lesion regions were eliminated by mistake leaving some holes (green voxels in Figure 4-12 (a1) and black voxels in Figure 4-12(a2). These holes will be addressed in operation (c) of step 3 of post processing, to be discussed later. Any excessive pixels trimmed from the boundary can be recovered in operation (b) of step 3 of post processing where false negatives are addressed.

Step B - Elimination of the false negatives on the boundary of the lesion regions

The colored evaluation of the segmentation depicted in Figure 4-12(i1) shows green pixels (false negatives) connected to some of the lesion regions blue areas (true positives). These false negative pixels on the boundary of the lesion regions can arise from the dissimilarity between the textural properties (features) of the non-detected lesion areas and the textural properties of the detected lesion region itself. To recover these pixels, region growing is applied for each lesion region. Any of the pixels neighboring to each lesion region are included in the closely adjacent MS lesion region if the absolute difference between the grayscale of the pixel and the mean grayscale of the lesion region does not exceed the standard deviation of the grayscale of the lesion region. The region growing operation is repeated recursively until no new pixels are added. Figure 4-12(b1) and Figure 4-12(b2) show the effect of applying this recursive region growing operation, where most of the false negative pixels were recovered.

Step C - Forcing lesion continuity to eliminate false negatives (holes) inside the lesion regions

For each lesion region, there may be some pixels inside the region which are not detected in the initial segmentation (due to the block's textural features being different from features of MS blocks) or detected but removed as part of the elimination of false positives on the boundary of MS lesions during operation (a) of this step while trimming the lesion region. By applying the logical concept of lesion continuity, which means that the lesion cannot have inside holes, all the pixels inside the boundary of the lesion regions are assigned to be MS pixels. Figure 4-12(c1) and Figure 4-12(c2) show the effect of applying this operation where all the holes were filled.

All the operations visualized in Figure 4-12 shows that step 3 did not add or remove any lesion regions from the initial segmentation but only the lesion regions became more completed and smoother.

4.6 Pipeline 2 - Single View and Multi-Channels MRI Pipeline

This pipeline is similar to the previous pipeline but supports inputs of multi-channels MRIs. The supported channels are FLAIR, T1, T2 and PD MRI sequences. The following subsections describes the added functions to the modules of pre-processing, core textural SVM and post-processing to benefit from the multi-channels data and also benefits from the registration of the T1 channel to the MNI atlas template to provide spatial priors.
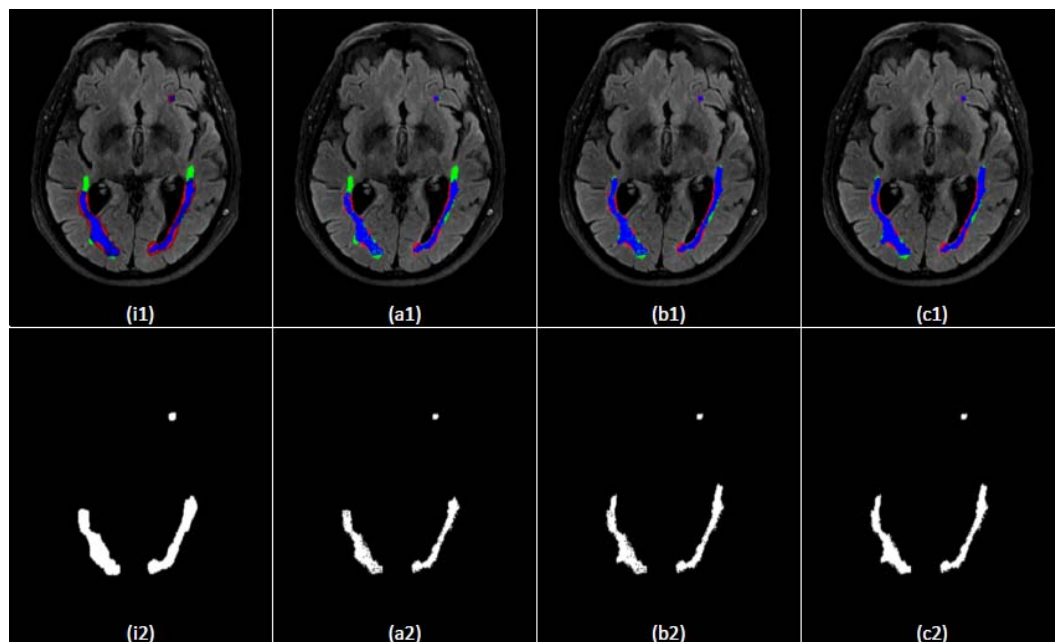
Figure 4-12: Post processing Step3: Lesion Regions Shape Correction.

(i1) and (i2): the initial colored evaluation of the segmentation and the initial segmentation. (a1) and (a2): Effect of applying step 3a. (b1) and (b2): Effect of applying step 3b. (c1) and (c2): Effect of applying step 3c.

4.6.1 Changes in Preprocessing Module

In pipeline 1, the preprocessing module provides intensity correction to the FLAIR slices. In pipeline 2, offline registration to the probabilistic MNI atlas [**70**], co-registration among the different channels and Intensity correction to all channels are added.

Registration with MNI Probabilistic Atlas

The probabilistic MNI atlas [**70**] provides for each voxel the probability of belonging to the white matter (WM), the grey matter (GM) and the cerebro-spinal fluid (CSF). The template of the MNI atlas is a T1 sequence, thus the T1 sequence of the subject MRI is used for registration with the atlas. The registration is done offline before the segmentation using AIR package [**71**]. Based on the registration step, three values are provided for each voxel representing the prior probabilities of the voxel to belong to

white matter, gray matter or CSF tissues. A sample slice from the MNI atlas template and T1 slice along with the registered priors of the probabilistic MNI atlas are shown in Figure 4-13. Figure 4-13 (c-e) are the corresponding slice from the probabilistic tissue for the white matter (WM), gray matter (GM) and CSF tissue prior probabilities, respectively.
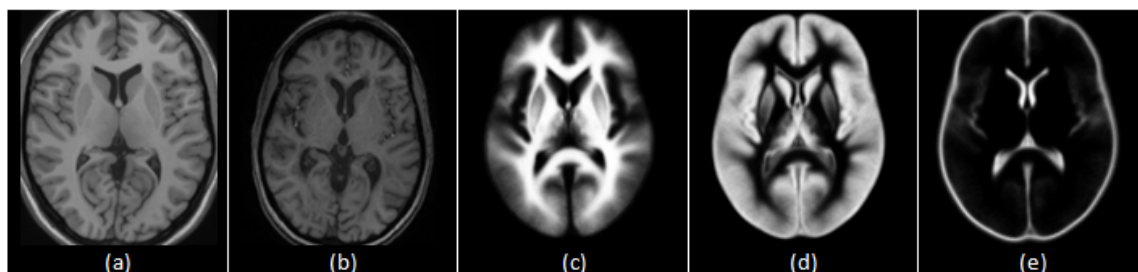
Figure 4-13: Registration To MNI Atlas with T1 sequence.

(a) a slice from the MNI atlas template (b) a T1 slice from the subject MS3 registered to slice shown in (a)
(c)Registered White Matter Probability (d) Registered Gray Matter Probability (e) Registered CSF Probability

Co-registration

As described in the datasets section 2.1, the datasets acquired at the University of Miami Miller School of Medicine, have different resolutions across the channels. These datasets are used in our research in three settings: single-channel, multi-channels, and MS Challenge settings. In multi-channels settings (this pipeline), the other sequences (T2, PD, and FLAIR) are co-registered to the T1 sequence that was already registered to the MNI atlas. The expert MS labels are manually delineated on the FLAIR sequence. In order to keep the manual MS labels unaffected through the registration, the FLAIR sequence should not go through minimum changes in the registration process. Thus, T2 and PD are co-registered to T1. Then, all of (T1, T2, and PD) are re-sampled to the same resolution and the same inter-slices distances of the FLAIR. Thus, each slice of FLAIR sequence corresponds to a slice from the other sequences. Then, 2D registration is

applied to each slice of FLAIR to match corresponding slice of T1 sequence and the transformation matrix parameters are extracted and applied to the MS label overlay. A sample T1 slice along with the other channels and the ground truth lesion mask registered to the T1 sequence are shown in Figure 4-14. Figure 4-14 (a) shows a T1 slice (from subject MS6) registered to the MNI atlas template and down sampled to the resolution of FLAIR. Figure 4-14 (b) shows the corresponding FLAIR slice corresponding to the T1slice after registration. Figure 4-14 (c) shows the ground truth lesion mask superimposed over the FLAIR slice after registration. In Figure 4-14 (d), lesion mask is extracted. Figure 4-14 (e and f) show the corresponding T2 and PD slices corresponding to the T1slice after registration, respectively.
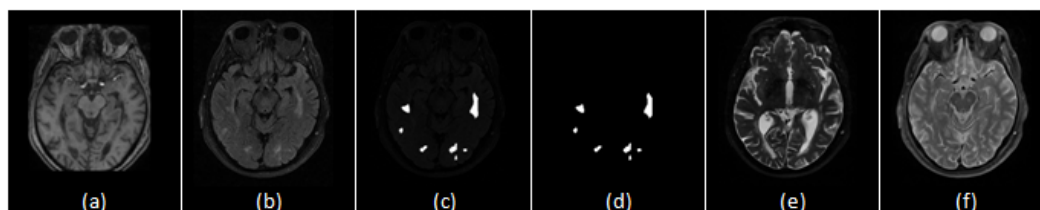
(a) a T1 slice from the subject MS6 (b) FLAIR slice registered to T1 (c) Ground Truth Lesion mask superimposed over the FLAIR slice is registered (d) the lesion mask is extracted after registration (e) T2 slice registered to T1 (f) PD slice registered to the T1.

Intensity Correction:

The same intensity correction module used in pipeline 1 is applied in pipeline 2 for the all channels.

4.6.2 Changes in the Processing Core Module "Textural Based SVM"

The same engine used in pipeline 1 is used in pipeline 2 with only two added categories of blocks features are added to the feature vector: other channels intensities and the tissues priors obtained from the MNI atlas.

Changes in the Feature Vector

The block features are classified into five categories: textural features, position features, co-registered intensities, tissues priors and neighboring blocks features. The updated features vector is listed in Table 4-2.

Multi-Channels Intensities

The other channels features ($C2$ and $C2$) include intensity means for the corresponding block in the T1, T2, and PD channels.

Spatial probabilistic tissues Priors

The atlas spatial prior probabilities features include the means of the priors extracted from the probabilistic atlas (White Matter, Gray Matter, and CSF probabilities) for the block.

The textural features, position features, and neighboring blocks features. categories were explained in pipeline 1.

4.6.3 Post-Processing Module

To benefit from the tissues probabilities in the post-processing module, step 1 and step 3A of the post-processing module used in pipeline 1 are replaces with SVM based Deleter and SVM based Trimmer respectively.

Features Category	Features
Textural features	1-Mean (Histogram based feature) 2-variance (Histogram based feature) 3-Gradient mean (Gradient based feature) 4- Gradient variance (Gradient based feature) 5-8 Grey level non-uniformality in the 4 directions (Run length based feature) 9-12 Run length non-uniformality in the 4 directions (Run length based feature) 13-16 Contrast in the 4 directions (Co-occurrence matrix based feature) 17-20 Absolute value in the 4 directions (Co-occurrence matrix based feature) 21-24 Entropy in the 4 directions (Co-occurrence matrix based feature)
Position features	25-Slice relative location 26- Normalized radial distance between block and slice center
Other Channels	27-T1 Mean 28- T2 Mean 29- PD Mean
Atlas Spacial Priors	30-WM Probability Mean 31- GM Probability Mean 32-CSF Probability Mean
Neighboring blocks features	33-40 Differences between grayscale of the block and each of the 8 neighboring blocks

Table 4-2 : A square Block Features Vector in Single-View Multi-Channels Pipeline.

SVM based Deleter

To improve elimination of false positive regions (step 1), an SVM lesion region eraser engine is introduced. This is an engine is based on a features vector for the segmented lesion region. The features vector is summarized in Table 4-3.

Features Category	Features
Lesion Region Area and Position	Lesion Region Area Normalized Cartesian coordinates of the top left point of the lesion region.
Segmentation Mean Score of the Lesion Region	Mean of Segmentation scores of the lesion regions pixels at the conclusion of the initial segmentation step.
Region Intensity Mean	Mean of Grayscale intensities of the lesion region pixels (for the supported channels).
Difference between Lesion Region Intensity Mean and Background Intensities mean	Difference between the grayscale intensity of the pixel and the mean grayscale intensity of the lesion region neighboring background (for the supported channels). Difference between the grayscale intensity of the pixel and the mean grayscale intensity of the slice background (for the supported channels).

Table 4-3 : Segmented Lesion Region Features Vector

The lesion region features include the lesion region area which is calculated by counting the pixels of the region. The Cartesian coordinates of the top left pixel of the lesion region are normalized; for each feature, the mean of the feature values was subtracted from the voxel value, and the outcome was divided by the standard deviation. This approach resulted in a mean of 0 and variance of 1 for all features. The position features are added because some of the false positive MS lesion regions are located in uncommon locations. Odd locations include MS lesions outside the brain area, close to the brain boundary, or close to the sagittal plan. Since the spatial features were also normalized by this method, this implicitly corrects for differences in size and location between the patients. The segmentation score of the pixels of the lesion region are averaged to get a feature for the region. The lesion region mean intensities (for the supported channels) are included as a feature. The difference between the mean grayscale intensity and the background mean intensity is used to formulate features representing the lesion region. There are two types of region backgrounds; the neighboring background and the slice

background. The neighboring background for the lesion region is defined as the rectangle whose width and length are three times the width and length of the smallest rectangle enclosing the lesion region respectively and the lesion region are centered in this rectangle. Figure 4-15 shows the lesion region neighboring background marked with yellow lines. The slice background is the brain area surrounding the lesion region. The means of intensities of neighboring and slice backgrounds are calculated excluding any other lesions in these backgrounds. The difference between grayscale intensity of the lesion region and both backgrounds intensities mean for all of the four channels are used as features in the features vector.
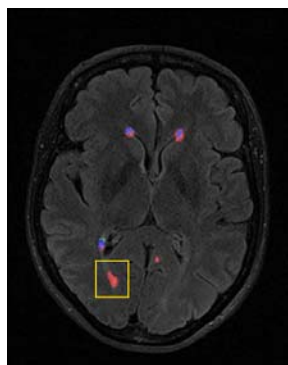
Figure 4-15: Neighboring Background (Marked by Yellow rectangle) of a Lesion Region (Marked by Red indicating False Positive).

The deleter engine is a SVM engine added to provide a decision to keep (or discard) each segmented lesion region. It is trained with the evaluation of the result of the main segmentation process. The same subject dataset used in training the main segmentation engine is segmented by the main segmentation engine and the segmented lesion regions are compared with the ground truth. For each lesion region of the segmented lesion region, the above features are calculated and Keep/Discard decision of the lesion region

is assigned to 1 (Keep) in the training set if the lesion region has at least one pixel marked as MS lesion in the ground truth and 0 (Discard) otherwise. The trained SVM engine is used to delete the false positive lesion regions in the texturally segmented lesion regions.

 SVM based Trimmer

To improve elimination of false positive pixels in the true regions (step 3A), especially on the boundary, a SVM lesion region trimmer engine is introduced. This is engine is based on a features vector for the pixel of the segmented lesion region. This feature vector is summarized in Table 4-4.

The trimmer engine is a SVM engine added to provide Keep/Discard decisions for each pixel of the lesion region. It is trained with the evaluation of the result of the main segmentation process. The same subject dataset used in training the main segmentation engine is segmented by the main segmentation engine and the segmented lesion regions are compared with the ground truth. For each pixel of the segmented lesion region, the above features are calculated and a Keep/Discard decision for the pixel is assigned to 1 (Keep) in the training set if the pixel is part of an MS lesion in the ground truth and 0 (Discard) otherwise. The trained SVM engine is used to trim the false positive pixels in the segmented lesion regions.

Features Category	Features
Relative Position of Pixel to region	Euclidean distance between the pixel and the lesion region calculated center. This distance is normalized by dividing by the lesion region perimeter. Euclidean distance between the pixel and the lesion region boundary. This distance is normalized by dividing by the lesion region perimeter.
Segmentation Score of the pixel	Segmentation score of the pixel at the conclusion of the initial segmentation step.
Pixel Intensity	Grayscale intensity of the pixel (for the supported channels).
Difference between Pixel Intensity and Region and Background Intensities mean	Difference between the grayscale intensity of the pixel and the mean grayscale intensity of the lesion region (for the four channels). Difference between the grayscale intensity of the pixel and the mean grayscale intensity of the lesion region neighboring background (for the supported channels). Difference between the grayscale intensity of the pixel and the mean grayscale intensity of the slice background (for the supported channels).
Atlas Spatial Priors	Pixel WM Probability Pixel GM Probability Pixel CSF Probability

Table 4-4 : Segmented Pixel Features Vector

4.7 Pipeline 3 – Segmentation of MS in Multi-Sectional Views Multi-channel Slices

The MS lesion volumes can be viewed from the different sectional views of the brain MRIs. Figure 4-16 shows an MS lesion volume in the three sectional views. A lesion region in the axial view is pointed to by a cursor in Figure 4-16(a), its corresponding lesion region in the sagittal view in Figure 4-16(b) and its corresponding lesion region in the coronal view in Figure 4-16(c). The proposed technique aims to benefit from the multiple views in improvement the quality of MS lesions segmentation. Our segmentation technique strategy is to superimpose the segmentations, in which slice segmentation are performed on axial, sagittal and coronal views and the resultant segmentations are aggregated to generate the final segmentation.
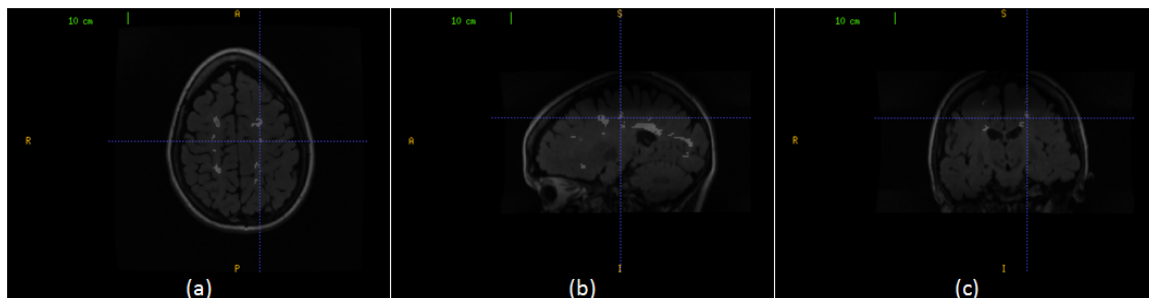
Figure 4-16: An MS lesion volume in the three sectional views. (a) The cursor points to an MS lesion region in the axial view. (b) The cursor points to the corresponding MS lesion region in the sagittal view. (c) The cursor points to the corresponding MS lesion region in the coronal view.

A pipeline was developed based on the preprocessing and textural based SVM processing core to implement the multi-sectional views segmentation. This pipeline is composed of preprocessing, three instances of textural based SVM core and multi-views' segmentation aggregation modules as shown in Figure 4-17. The input studies are preprocessed offline for registration and co-registration. The three textural based SVM core are similar engines; each is applied to a sectional view (Axial, Sagittal, and Coronal views). Through each engine, the multi-channels MRI slices of the brain are preprocessed for intensity correction to remove the effect of noise and variations in brightness and contrast among corresponding sequences of different subjects. The next step in each sectional view engine is the processing module, which is used for the detection of initial MS lesions regions based on textural features. The output segmentations of the sagittal and coronal view engines are rotated to be in the axial view. The output segmentations of the axial, sagittal, and coronal sectional views engines are noted as A, B, and C, respectively. The multi-sectional views' segmentations aggregation step is applied to A, B, and C to generate the final output segmentation. No post processing module is included in this pipeline.
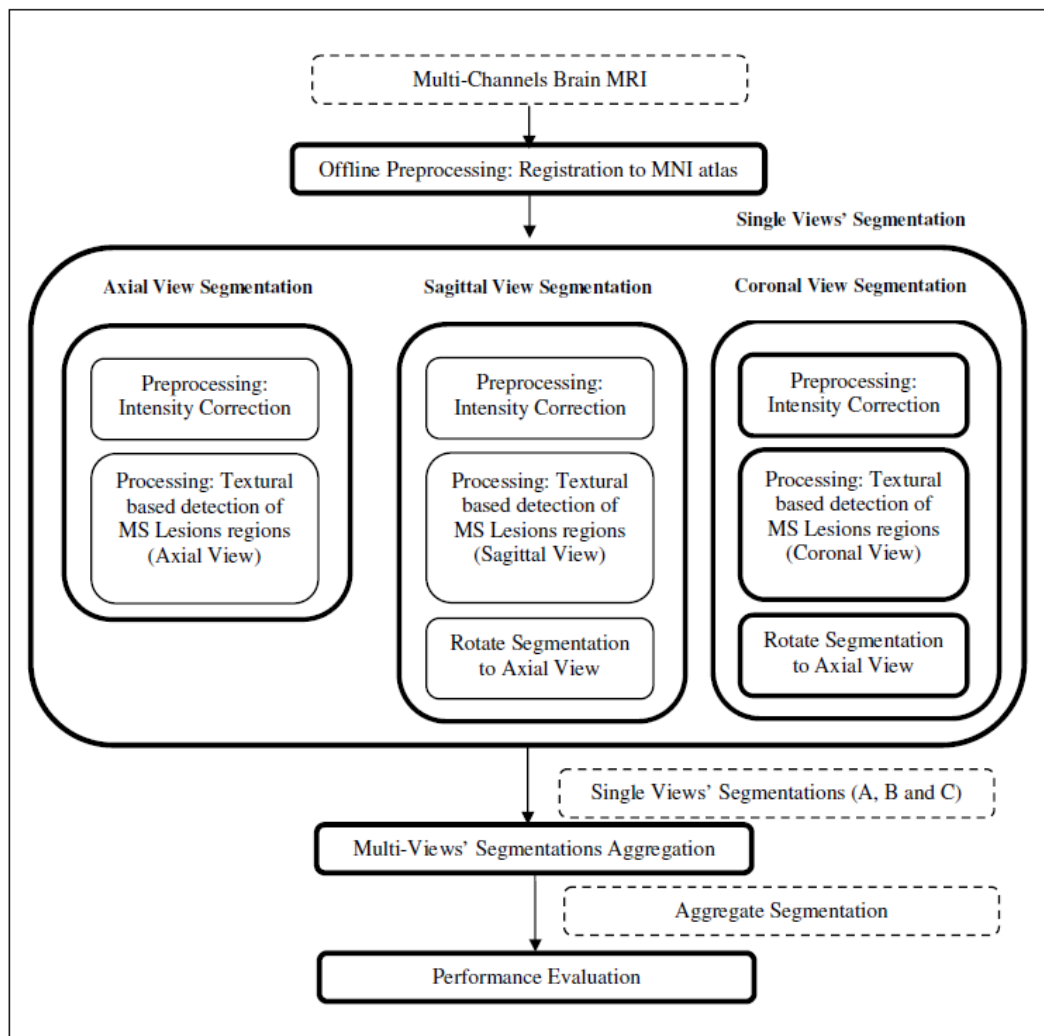
Figure 4-17: MS lesions Multi-sectional views – Multi-Channels MS lesions segmentation framework (Pipeline 3).

The block division is done in a non-overlapping manner to reduce computation time taking into consideration that the segmentation is done three times, once for each sectional view.

4.7.1 Changes in Pre-Processing Module

It uses the same pre-processing module used in pipeline 2 with only three channels (T1, T2, and FLAIR are supported).

For the MS Lesion Segmentation Challenge datasets, all datasets were rigidly registered to a common reference frame and re-sliced to isotropic voxel spacing, with resolution 512x512x512, using B-spline based interpolation. The images were then aligned on the mid-sagittal plane [34]. The template of the MNI atlas is a T1 sequence, thus the T1 sequence of the MRI datasets is used for registration with the atlas.

For the datasets acquired at the University of Miami Miller School of Medicine, have different resolutions across the channels. In this pipeline, these datasets are used in MS Challenge settings. In MS-Challenge settings, all the datasets were re-sliced to be in the same resolution conditions of MS Lesion Segmentation Challenge datasets to be tested by the models trained by MS Lesion Segmentation Challenge training data. Then they were registered to the MNI atlas using Automated Image Registration (AIR) software [35].

We have created a spatial priors atlas for the BrainWeb datasets analog to the standard MNI atlas. The BrainWeb database provides twenty anatomical models for normal brain. These anatomical models consist of a set of fuzzy tissue membership volumes, one for each tissue class, i.e., white matter, gray matter, cerebrospinal fluid, fat. The voxel values in these volumes reflect the proportion of tissue present in that voxel, in the range [0, 1]. Since we are interested in this dissertation in the white matter, gray matter, and CSF tissues probabilities, the corresponding tissues membership volumes in

the anatomical models are averaged over the twenty brain templates to get a spatial priors atlas to be used when testing the BrainWeb data.

4.7.2 Changes in core "textural based SVM" Module

The textural based SVM is cloned three times for the three sectional-views. For each sectional brain view (axial, sagittal or coronal), a trained classifier is designed to segment the MS lesions in the corresponding view. The details discussed in this subsection apply to the three classifiers. The single view classifiers are identical in the structure, criteria, and algorithm while the difference is only in the slices view from which the features are extracted.

It uses the same feature vector used in pipeline 2 listed in Table 4-2 (except that only PD channel based feature (no. 29) is not included).

The training takes place the same as the single views' pipelines (1 and 2), the only difference that it is performed in pipeline 3 for all sectional views. The segmentation is little bit different due to using non-overlapping blocks to reduce computation time as explained below:

Segmentation

In the selected brain view, each of the slices of the subject dataset to be segmented is divided into non-overlapping square blocks of size w**x**w pixels. The features vector for each block is calculated. The trained SVM for the corresponding brain view is used to predict the class labels for all the non-overlapping blocks. The block division is done in a non-overlapping manner to reduce computation time taking into consideration the high resolution of each slice and performing the segmentation three times, once for each

sectional view. For any block classified as MS block, assuming true positive classification, this does not mean that all pixels of the block should be classified as MS pixels because the SVM engine is trained to detect the blocks that contains MS pixels completely or partially. However, during segmentation, if any block is classified as MS block (y=1), all pixels inside the block are marked as MS voxel while the final decision for the individual pixels taken by the aggregate segmentation step. For the sagittal and coronal sectional-view segmentations, the output segmentation is rotated to be in the axial view. The output segmentations of the axial, sagittal, and coronal sectional views segmentation engines are noted as A, B, and C, respectively.

Figure 4-18 shows segmentation of sample slices from subject (CHB_train_Case01) in multi-sectional views (axial, sagittal, and coronal). Figure 4-18 (A) demonstrate axial sectional view segmentation: Figure 4-18 (A1) shows the original FLAIR slice. Figure 4-18 (A2) shows the ground truth for the lesions generated through manual segmentation. Figure 4-18 (A3) provides the colored evaluation of the segmentation. Similarly, Figure 4-18 (B) and Figure 4-18 (C) demonstrates sagittal and coronal views segmentations.
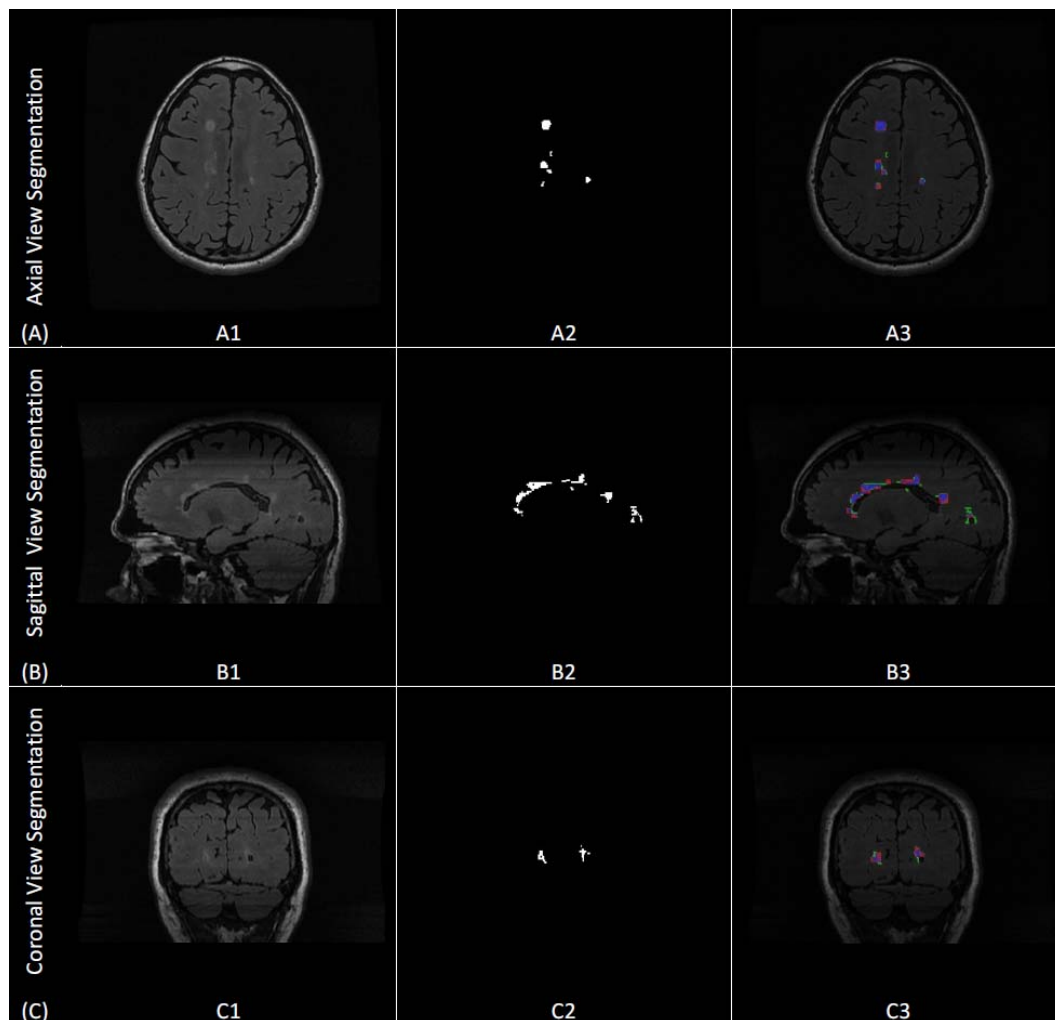
Figure 4-18: Using Textural SVM in Multi-Sectional Views.

(A) Axial Sectional-View Segmentation, (B) Sagittal Sectional-View Segmentation and (C) Coronal Sectional-View Segmentation. (A1, B1 and C1) Sample slice from CHB_train_Case07 in the corresponding View, (A2, B2 and C2) ground truth for the sample slice, (A3, B3 and C3) colored evaluation of the corresponding view segmentation.

4.7.3 Aggregate Segmentation

The set S includes all voxels in the dataset to be segmented. The resultant segmentations from the single view classification can be represented by sets A, B, and C for the axial, sagittal, and coronal segmentations, respectively. Each set includes the voxels that are classified as MS-voxel in the corresponding view segmentation. In the best case scenario, all the positive and negative classes are true, and A, B, and C should be identical. However, reality is that the three sets intersect with each other as shown in Figure 4-19. As the number of segmentation sets where a certain voxel is included increases, the higher the probability of this voxel to be true positives.
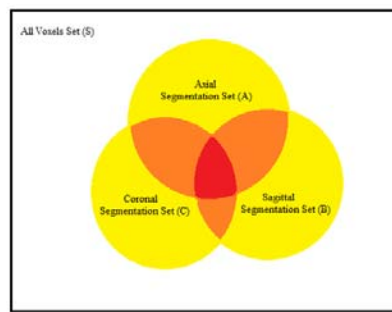
Figure 4-19: Segmentation sets of single views: Axial (A), Sagittal (B) and Coronal (C).

The aggregate segmentation function decides for each voxel $v \in S$ if it should be included in the aggregate segmentation set (G) according to the relation between v and the three sets of the sectional views segmentations A, B, and C. We model the aggregation as a binary classification problem that assigns for each voxel $v \in S$ either the class MS_1 (included in G) or the class MS_0 (not included in G).

For each voxel v, we define a discrete variable X_v that represents the number of segmentation sets where the voxel v is included. The variable X_v can take discrete symbolic values X_0, X_1, X_2, and X_3 according to the following definition:

$$X_v = \begin{cases} X_0, & v \in S - (A \cup B \cup C) \\ X_1, & v \in ([A - (B \cup C)] \cup [B - (A \cup C)] \cup [C - (A \cup B)]) \\ X_2, & v \in ([(A \cap B) - C] \cup [(B \cap C) - A] \cup [(A \cap C) - B]) \\ X_3, & v \in (A \cap B \cap C) \end{cases} \quad (4\text{-}6)$$

According to the above definition and the demonstration in Figure 4-19, $X_v = X_0$ if v falls inside the white area (v is not included in any of A,B or C), $X_v = X_1$ if v falls inside the yellow area (v is inside either A,B, or C), $X_v = X_2$ if v falls inside the orange area (v is included by any two sets of A,B, and C) and $X_v = X_3$ if v falls inside the red area (v is included in A,B, and C).

The Bayesian decision rule is applied to obtain the posterior probability $P(MS_1|X_v)$. This involves the calculation of the classes priors $P(MS_i)$ and the likelihood functions $P(X_v|MS_i)$ for (i=0 to 1). In order to obtain the prior probabilities and the likelihoods, statistical analysis was performed on the training datasets (the 20 training datasets from the MS Lesion Challenge). The prior probability $P(MS_1)$ is the percentage of voxels that are manually labeled as MS-voxels in the datasets ground truth while the prior probability $P(MS_0)$ is the percentage of voxels that are manually labeled as normal voxels. Then the training datasets were segmented using the three engines of the single view (Axial, Sagittal, and Coronal) segmentations to get the three sets A, B, and C respectively for the training datasets. The likelihood function $P(X_v = X_i|MS_1)$ is the percentage of voxels that are manually labeled as MS-voxels in the datasets ground truth

and in the same time $X_v = X_i$ according to the automatic segmentation for (i=0 to 3). In the same manner, $P(X_v = X_i | MS_0)$ is calculated for (i=0 to 3).

The posterior probability $P(MS_1 | X_v)$ is calculated using the Bayesian decision rule:

$$P(MS_1 | X_v) = (P(X_v | MS_1). P(MS_1)) / P(X_v) \tag{4-7}$$

where the evidence $P(X_v)$ is given by:

$$P(X_v) = P(X_v | MS_1). P(MS_1) + P(X_v | MS_0). P(MS_0) \tag{4-8}$$

Finally, the aggregate segmentation function uses the posterior probability $P(MS_1 | X_v)$ to select voxels from the the S to be included in the aggregate set G according the following rule:

$$G = \{v \mid P(MS_1 | X_v) \geq \tau \ \ \forall v \in S \} \tag{4-9}$$

with τ is a threshold that converts the posterior into a binary decision. The value 0.5 is selected for τ.

The aggregate segmentation of the multi-sectional views segmentations is demonstrated in Figure 4-20. Figure 4-20(a) shows a sample axial slice (257) from CHB_train_Case07. Figure 4-20(b) shows the axial view segmentation for this slice referred to by A_{257} (A stands to the axial segmentation set and the subscript 257 stands to the slice number). A segmented lesion region (R_A) is highlighted by yellow circle. Figure 4-20(c) shows sagittal sectional-view segmentation performed in a sagittal slice with the highlighted area intersects with (R_A). After performing sagittal view segmentation for all sagittal slices, the resultant segmentation is rotated to the axial view

(B). The result of this segmentation that corresponds to the original sample slice, referred to by B_{257} (B stands to the set of sagittal segmentation rotated to axial view and the subscript 257 stands to the slice number in axial view) is shown in Figure 4-20(d) with the highlighted area is a new segmentation for R_A namely (R_B). Similarly, Figure 4-20(e) shows coronal sectional-view segmentation performed in a coronal slice with the highlighted area intersects with (R_A). After performing coronal view segmentation for all coronal slices, the resultant segmentation is rotated to the axial view (C). The result of this segmentation that corresponds to the original sample slice, referred to by C_{257} (C stands to the set of coronal segmentation rotated to axial view and the subscript 257 stands to the slice number in axial view) is shown in Figure 4-20(f) with the highlighted area is a new segmentation for R_A namely (R_C). Figure 4-20(g) gives the aggregate segmentation of A_{257}, B_{257}, and C_{257} referred to by G_{257} (G stands to the aggregate segmentation set and the subscript 257 stands to the slice number in axial view). The highlighted lesion region (R_G) is the aggregate of the (R_A), (R_B), and (R_C). Figure 4-20(h) shows the colored evaluation of the aggregate segmentation with the highlighted lesion region (R_G) is colored mostly with blue (true positive voxels) along with small portions colored with red and green (false positive and false negative voxels respectively).
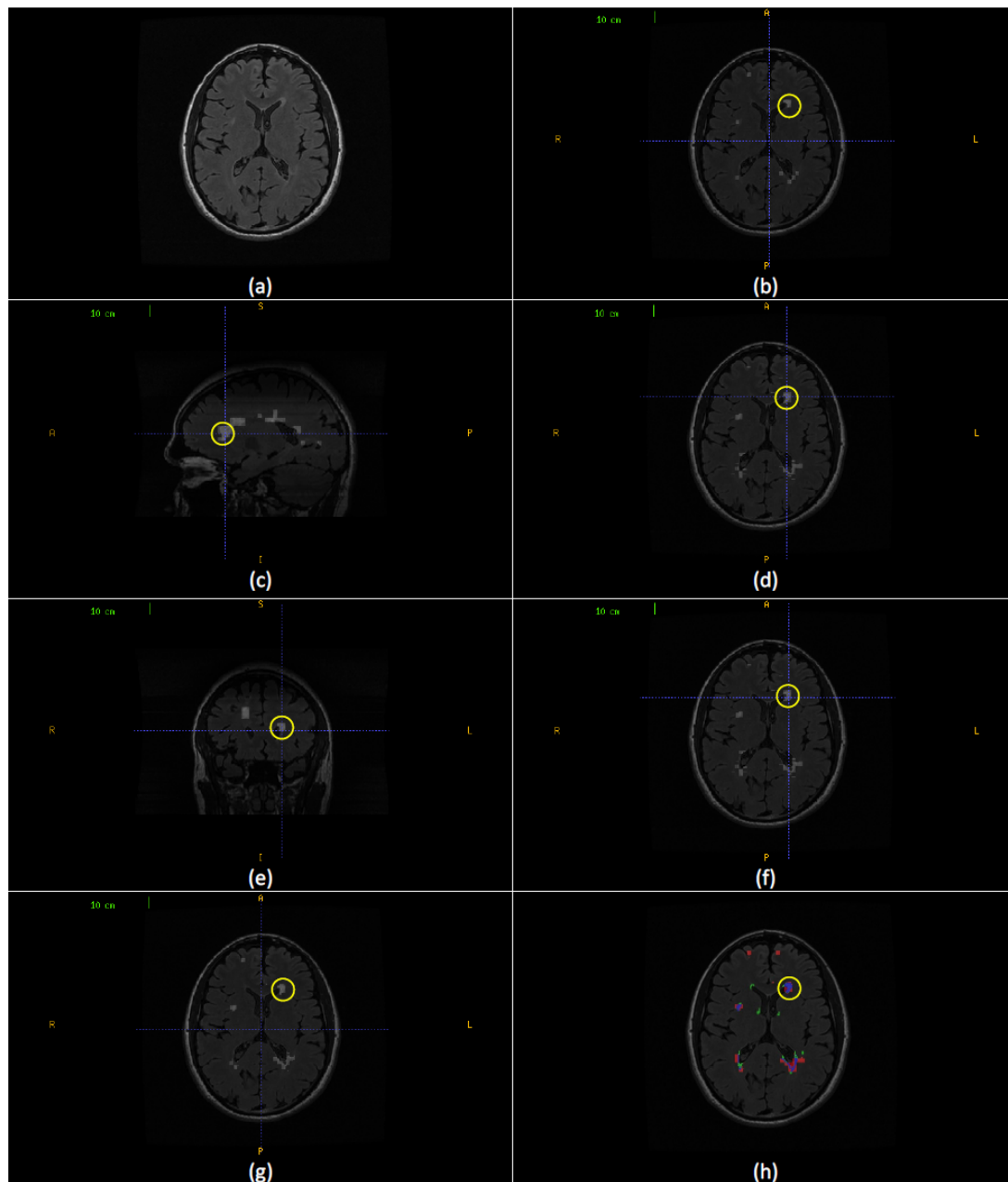
Figure 4-20: Aggregate segmentation of the multi-sectional views segmentations.

(a) Axial slice (257) from CHB_train_Case07. (b) Axial sectional-view segmentation of the slice (A_{257}) with a lesion region (R_A) highlighted in yellow. (c) Sagittal sectional-view Segmentation. The highlighted lesion region intersects with (R_A). (d) Segmentation of the original slice (B_{257}) in sagittal sectional-view segmentation rotated to axial view with lesion region (R_B) highlighted in yellow. (e) Coronal sectional-view segmentation. The highlighted lesion region intersects with (R_A). (f) Segmentation of the original slice (C_{257}) in coronal sectional-view segmentation rotated to axial view with lesion region (R_C) highlighted in yellow. (g) Segmentation of the original slice (G_{257}) by aggregates segmentation of (A_{257}), (B_{257}) and (D_{257}). The marked lesion region (R_G) is the aggregate of (R_A), (R_B) and (R_C). (h) Colored evaluation of the aggregate segmentation (G_{257}) and the highlighted area is the colored evaluation if (R_G).

4.8 Experimental Results

This section integrates all tests done to evaluate the different pipelines. Although the development of three segmentation pipelines are done in sequence, we can look to the technique as a single segmentation framework and each pipeline can be obtained by configuring the pre-processing, core, and post-processing modules to ease discussion of the testing results.

Possible Configurations of the Preprocessing Module

The segmentation pipeline uses the preprocessing module configured using the following options:

- Intensity Correction on selected brain sectional view: Axial, Sagittal or Coronal.

- Intensity Correction on selected Channels.

- Registration of T1 channel to the MNI T1 atlas can be enabled or disabled.

- Co-Registration among the different channels can be enabled or disabled.

Possible Configurations of the Textural based SVM Core Module

The segmentation pipeline uses the textural based SVM configured using the following options:

- Segmentation is done on selected brain sectional view: Axial, Sagittal or Coronal.

- Slice is divided into either overlapping or non-overlapping blocks.

- Blocks feature vector includes textural, position, and neighboring blocks features and:

  o Spatial Features (Tissues Probabilities) can be included or not.

- Other Channels Mean Intensities Features Can be:
  - None.
  - Means of the intensities of another 2 channels T1 and T2.
  - Means of the intensities of another 3 channels T1, T2, and PD.

Possible Configurations of the Post Processing Module

The segmentation pipeline uses the post-processing module configured using the following options:

- Step 1 False positive Lesion Regions Reduction – can be done using either:
  - SVM based Deleter module or
  - Deletion of false region based on location.
- Step 2: False Negative Lesion Regions Reduction by inter-slices comparison (No Configuration).
- Step 3 Lesion Region Shape Correction
  - A- False positive on boundary reduction- can be done using either:
    - Fuzzy based engine.
    - SVM based Trimmer.
  - B- False negative on boundary reduction by region growing - No Configuration.

C- False negatives inside region reduction by forcing lesion continuity - No Configuration.

In the following subsections, the core (main) classifier is verified, single view pipelines (1 and 2) are validates and the multi-views pipeline is evaluated.

4.8.1 Validation of the Main Classifier

Receiver operating characteristics (ROC) analysis was performed to evaluate the main processing classifier that generates the initial MS-blocks. Due to using overlapping blocks, each slice pixel is included in 64 blocks (in case of using square blocks of 8x8 pixels). A score is given to each pixel equals to the number of blocks that encloses the pixel and classified as MS-block. To draw the ROC curve, a threshold is defined as the number of positive blocks needed to consider the pixel as MS pixel. This threshold was changed from 64 blocks down to 0, and for each case the specificity (true negatives rate) and sensitivity (true positives rate) were calculated in order to create the ROC curve as plotted in Figure 4-21 where the false positive rate (1-specificity) is on x-axis and the sensitivity is on y-axis. Using very high threshold leads to zero false positives and very low sensitivity while very low threshold leads to both very high false positive rate and sensitivity. For all tests, the ROC curve falls above the diagonal indicating good classification.
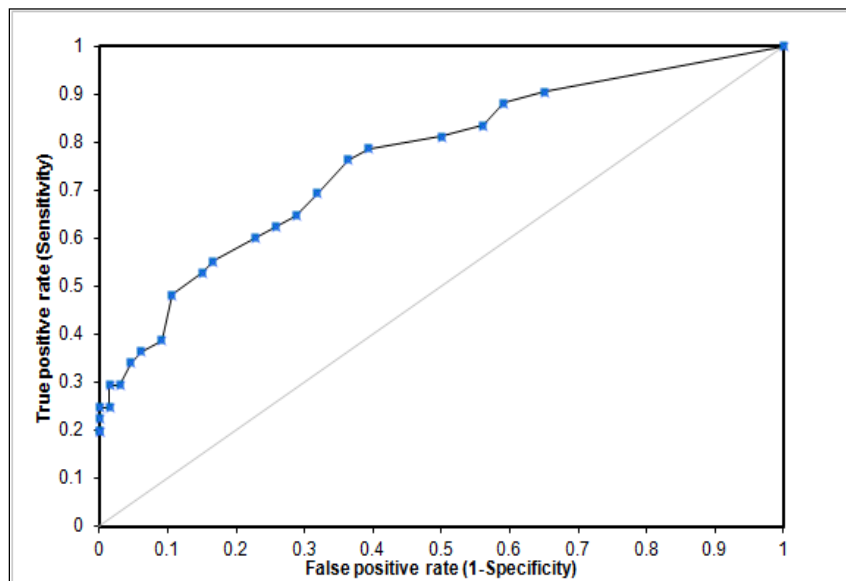
Figure 4-21: Receiver Operating Characteristics (ROC) curve for main processing performance.

To compare the ROC curve of the proposed technique with one of the state of the art methods that used ROC for evaluation, the free-response receiver-operating characteristic curve for overall performance of method proposed by Yamamoto et al. [6] is cloned in

Figure 4-22. Since their method does not involve threshold, they could not draw real ROC, but instead, the sensitivity is plotted versus the number of false positives per slice.
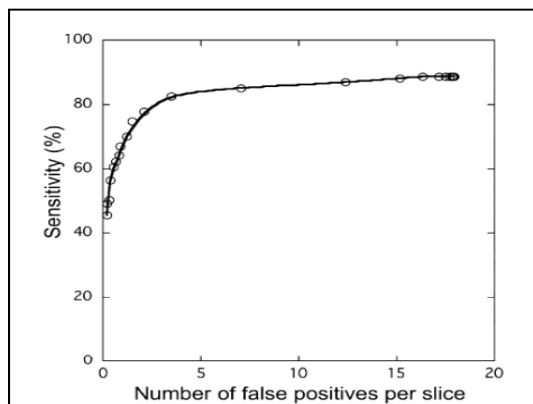
Figure 4-22: Free-response receiver-operating characteristic curve for overall performance of a state of the art method in detection of MS lesions.

4.8.2 Validation of the Single View Pipelines (Pipeline 1 and Pipeline 2)

The dataset of ten real FLAIR MRI axial sequences were used to evaluate the performance of the proposed single view pipelines (pipeline 1 and pipeline 2). The performance metrics detailed in section 4.2 were calculated.

In pipeline 1, the preprocessing, the textural based SVM core and post-processing modules are configured as following:

Pre-processing module configuration:

- Intensity Correction on *axial* view.

- Intensity Correction on *FLAIR* slices.

- Registration of T1 channel to the MNI T1 atlas is *disabled*.

- Co-Registration among the different channels is *disabled*.

Textural based SVM configuration:

- Segmentation is done on *axial* view.

- Slice is divided into *overlapping* blocks.

- Blocks feature vector includes textural, position and neighboring blocks features and:
  - Spatial Features (Tissues Probabilities) is *not* included.
  - Other Channels Mean Intensities Features is *not* included.

Post-Processing module configured using the following options:

- Step 1 False positive Lesion Regions Reduction – using ***Deletion of false region based on location***.

- Step 2: False Negative Lesion Regions Reduction by inter-slices comparison.

- Step 3 Lesion Region Shape Correction

  - o A- False positive on boundary reduction either ***Fuzzy based engine***.

  - o B- False negative on boundary reduction by region growing.

  - o C- False negatives inside region reduction by forcing lesion continuity.

In pipeline 2, the textural based SVM core and post-processing modules are configured as following:

Pre-processing module configuration:

- Intensity Correction on ***axial*** view.

- Intensity Correction on ***4 channels: FLAIR, T1, T2 and PD*** slices.

- Registration of T1 channel to the MNI T1 atlas is ***enabled***.

- Co-Registration among the different channels is ***enabled***.

Textural based SVM configuration:

- Segmentation is done on ***axial*** view.

- Slice is divided into ***overlapping*** blocks.

- Blocks feature vector includes textural, position and neighboring blocks features and:

  - o Spatial Features (Tissues Probabilities) is ***included***.

  - o Other Channels Mean Intensities Features include:

▪ Means of the intensities of another *__3 channels: T1, T2, and PD__*.

Post-Processing module configured using the following options:

- Step 1 False positive Lesion Regions Reduction – using *__SVM based Deleter__*.

- Step 2: False Negative Lesion Regions Reduction by inter-slices comparison.

- Step 3 Lesion Region Shape Correction

  o A- False positive on boundary reduction using *__SVM based Trimmer__*.

  o B- False negative on boundary reduction by region growing.

  o C- False negatives inside region reduction by forcing lesion continuity.

The segmentation results are summarized in Table 4-5. In this table, for each study subject, the dice similarity based on lesion regions DS_R, the dice similarity based on voxels DS_V, sensitivity, and detected lesion load are given. Overall average is given for each of these performance metrics.

Study	Pipeline 1 (FLAIR)				Pipeline 2 (Multi-Channels)			
	DS_R	DS_V	Sensitivity$_V$	DLL	DS_R	DS_V	Sensitivity$_V$	DLL
MS2	0.88	0.78	0.67	0.93	0.88	0.78	0.67	0.93
MS3	0.96	0.93	0.89	1.1	0.91	0.86	0.86	0.96
MS4	0.68	0.64	0.72	0.92	0.82	0.74	0.79	0.93
MS5	0.72	0.68	0.59	0.65	0.72	0.68	0.59	0.65
MS6	0.84	0.72	0.64	0.76	0.84	0.72	0.64	0.76
MS7	0.77	0.67	0.63	0.91	0.83	0.72	0.72	0.93
MS8	0.68	0.68	0.72	1.05	0.68	0.68	0.72	1.05
MS9	0.76	0.63	0.57	0.78	0.81	0.7	0.71	0.82
MS10	0.83	0.71	0.62	0.81	0.83	0.71	0.62	0.81
MS11	0.75	0.68	0.74	1.12	0.82	0.78	0.81	1.3
Average	0.79	0.71	0.68	0.90	0.814	0.737	0.713	0.914

Table 4-5 : Segmentation Result

The single-channel pipeline has average metrics of 0.79 for DS_R, 0.71 for DS_V, 0.68 for sensitivity, and 0.9 for percentage of detected lesion load. Better results are obtained for the multi-channel pipeline where the average metrics are 0.81 for DS_R, 0.73 for DS_V, 0.71 for sensitivity, and 0.91 for percentage of detected lesion load. The average detected lesion load indicates that the proposed method could detect the MS lesion with reasonable error rates. Although the average dice similarity based on voxels DS_V is exceeds the minimum value for reasonably good segmentation, there were drops in the performance for some of the studies. Based on the analysis of the results for these studies, the MS lesions were found to be very small for these studies (percentage of MS lesions volume in voxels to the total volume in voxels less than 0.1%).

Excellent result for the segmentation of the training set is a bottom line for accepting the technique. If the segmentation result of the training dataset (MS3) is removed from the average calculation, the average metrics would be lower but still in the good range. However, it is included for comparison with results of other techniques that includes the training set segmentation result in their averages.

The effect of the post processing steps on the overall performance is shown in Figure 4-23. For each study, the dice similarity based on Voxels (DS_V) is calculated before and after the use of the post processing step for both pipelines. The average improvement in dice similarity of the overall segmentation due to the post processing stage varies is 12% for pipeline 1 and 15% for pipeline 2.
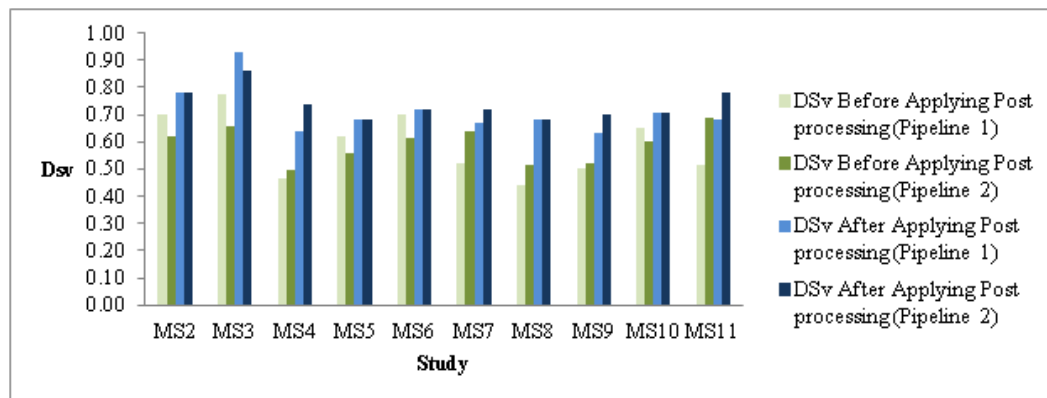
Figure 4-23: Effect of the Post processing steps on the overall segmentation performance.

One of the recent publications [**6**] provides a comparison table between different techniques for detection of the multiple sclerosis lesions according to the dice similarity based on lesion regions. We quote the table with in Table 4-6 with our results added as the last two line for the two single view pipelines. For each technique, the citation is referenced and the methods used in segmentation are provided along with the number of subjects used in the evaluation and the average dice similarity obtained using the technique. In the original publication of [**6**], the dice similarity was calculated based on the common regions between the manual segmentation and the automatic segmentation and similarly we also used the value of DS_R in the table 3. The table shows that our method has an average regional dice similarity of which is the highest among the past studies for both pipelines. This does not mean that our method is the best in terms of automatic segmentation performance because the comparative results in table 3 are dependent on the image properties of the datasets, which are different among the techniques included in the table. However, the comparison shows the success of our method for detecting MS lesions in real MRI datasets with competitive results.

Authors	Segmentation Method	#studies	Average DS$_R$
Boudraa et al. 2000 [72]	Fuzzy C-Means	10	0.62
Leemput et al. 2001 [60]	Stochastic model	50	0.51
Zijdenbos et al. 2002 [73]	Pipeline analysis	29	0.68
Khayati et al. 2008 [36]	AMM (adaptive mixtures method), MRF (Markov random field model)	20	0.75
Yamamoto et al. 2010 [6]	Region growing, LSM (level set method)	6	0.77
Proposed Pipeline1 [74]	Texture Analysis,SVM (Pipeline 1)	10	0.79
Proposed Pipeline2 [75]	Texture Analysis,SVM (Pipeline 2)	10	0.8

Table 4-6 : Comparison of the automated methods for detection of MS in MR images

4.8.3 Validation of the Multi-Sectional Views Pipeline (Pipeline 3)

Evaluation was performed using both synthetic data and real MRI data containing varying levels of MS lesion load at different locations in the brain.

In pipeline 3, the three pairs of pre-processing and textural based SVM cores are configured as following:

Pre-processing module configuration:

- Intensity Correction on ***axial, sagittal, and coronal (individually)*** view.

- Intensity Correction on ***3 channels: FLAIR, T, and T2*** slices.

- Registration of T1 channel to the MNI T1 atlas is ***enabled***.

- Co-Registration among the different channels is ***enabled***.

Textural based SVM configuration:

- Segmentation is done on **_axial, sagittal, and coronal (individually)_** view.

- Slice is divided into **_non-overlapping_** blocks.

- Blocks feature vector includes textural, position, and neighboring blocks features and:

  - Spatial Features (Tissues Probabilities) is **_included_**.

  - Other Channels Mean Intensities Features include:

Means of the intensities of another **_2 channels: T1, T2_**.

The following subsections provide details about the evaluation settings, metrics, result in comparison to other methods and comment on results.

Synthetic Data

The BrainWeb database provides three simulated subjects brains involving three levels of MS lesion; mild, moderate, and severe referred to in this dissertation as MSLES 1, MSLES2, and MSLES3, respectively. The features vector used in the proposed method uses multi-channels image intensities with textural features from the FLAIR sequence. However, the FLAIR sequence is not provided in the BrainWeb MRI data, but the Inversion recovery sequence (IR) provided in the BrainWeb MRI data is used instead since it is the closest sequence to FLAIR. Besides, the feature vector depends on spatial priors' atlas to provide white matter, gray matter, and CSF tissues probability for each voxel. For this purpose, a simulated atlas was generated using anatomical models provided for the twenty normal simulated brains in the BrainWeb database as described in preprocessing section.

In the training phase, two different engines were trained. The first engine (Engine_Moderate_0%) was trained using moderate level lesion subject MSLES2 with 0% of noise level. The second engine (Engine_Severe_0_3_7_9%) was trained using severe level lesion subject MSLES3 at noise levels 0%, 3%, 7%, and 9% with each noise level sharing 25% of the training set. In segmentation phase, all the three subjects were tested with all noise levels and the obtained results are shown in Table 4-7. For each case, the segmentation is evaluated using dice similarity (DS) and detected lesion load (DLL) for both trained engines.

It is clear from Table 4-7 that using the second engine (trained with the severe level of MS lesions and varying noise levels) provides better segmentation performance for the mild, moderate, and severe cases for different noise levels compared to the first engine (trained with the moderate MS case at 0% noise level) in terms of the dice similarity (DS) and detected lesion load (DLL). Even the moderate MS case has better performance for the high noise levels with the second engine than with the first engine which is trained with the same dataset at 0% of noise.

The results were compared to those obtained by Garcia-Lorenzo et al. [15] for the same dataset. The comparison for the mild, moderate, and severe cases is provided in Figure 4-24(a), Figure 4-24 (b) and Figure 4-24 (c) respectively. In [15], the training was made using the 0% noise level template for each case and the segmentation is tested using the other noise levels, excluding the 0% noise level, in contrast to our testing that involves intra subject evaluation (training with a subject template and testing another subject). Therefore, the 0% will have no results in the charts for the Garcia Lorenzo [15] data in Figure 4-24.

Besides, the results were compared to Leemput et al. [**25**] and Freifeld et al. [**35**] for the moderate dataset, which was the only MS lesion level provided by the BrainWeb at the time of publication of these two methods. The comparison for the different noise levels is shown in Table 4-8. In [**25**] and [**35**], the training was made using the 0% and 1% noise levels for the moderate case template and the segmentation is tested using the other noise levels excluding the 0% and 1% noise levels for the same case where our testing involves segmentation of all moderate case noise levels using the engine trained by severe case (Engine_Severe_0_3_7_9%). Therefore, the 0% and 1% will have no results for [**25**] and [**35**] data in Table 4-8.

Comparison with Gracia-Lorenzo et al. [**15**] in Figure 4-24 shows stability of the performance using the second engine against different noise levels even for the mild and moderate cases which are not included in the training set, which supports using different noise levels in the training set. Comparison with Leemput et al. [**25**] and Freifeld et al. [**15**] shows competitive performance while considering that we use an engine trained with a different brain template compared to the segmented brain data while both of techniques used the same brain template for training and segmentation with different noise levels.

Subjects	Noise Level	Using Engine_Moderate_0%		Using Engine_Severe_0_3_7_9%	
		DS	DLL	DS	DLL
MSLES1 (Mild)	0%	0.79	1.06	0.57	1.65
	1%	0.69	1.21	0.57	1.49
	3%	0.37	3.31	0.55	1.72
	5%	0.06	27.06	0.49	1.82
	7%	0.03	50.60	0.49	1.75
	9%	0.03	53.16	0.53	1.65
MSLES2 (Moderate)	0%	0.88	0.85	0.82	0.74
	1%	0.87	0.85	0.81	0.74
	3%	0.85	0.88	0.80	0.71
	5%	0.53	2.00	0.77	0.69
	7%	0.33	3.86	0.75	0.66
	9%	0.23	5.22	0.72	0.63
MSLES3 (Severe)	0%	0.94	0.92	0.94	0.93
	1%	0.94	0.92	0.93	0.92
	3%	0.89	0.93	0.93	0.90
	5%	0.55	2.18	0.92	0.88
	7%	0.35	3.87	0.90	0.85
	9%	0.32	4.11	0.88	0.80

Table 4-7 : Segmentation results for the BrainWeb subjects for different noise levels

Noise Level	Leemput 2001 et al.[16]	Freifeld 2008 et al. [42]	Proposed Method[76]
0%	N/A	N/A	0.82
1%	N/A	N/A	0.81
3%	0.80	0.79	0.80
5%	0.73	0.79	0.77
7%	0.61	0.78	0.75
9%	0.47	0.76	0.7

Table 4-8 : Comparison between the proposed method and state of the art methods on BrainWeb moderate lesion level dataset for different noise levels based on of dice similarities.
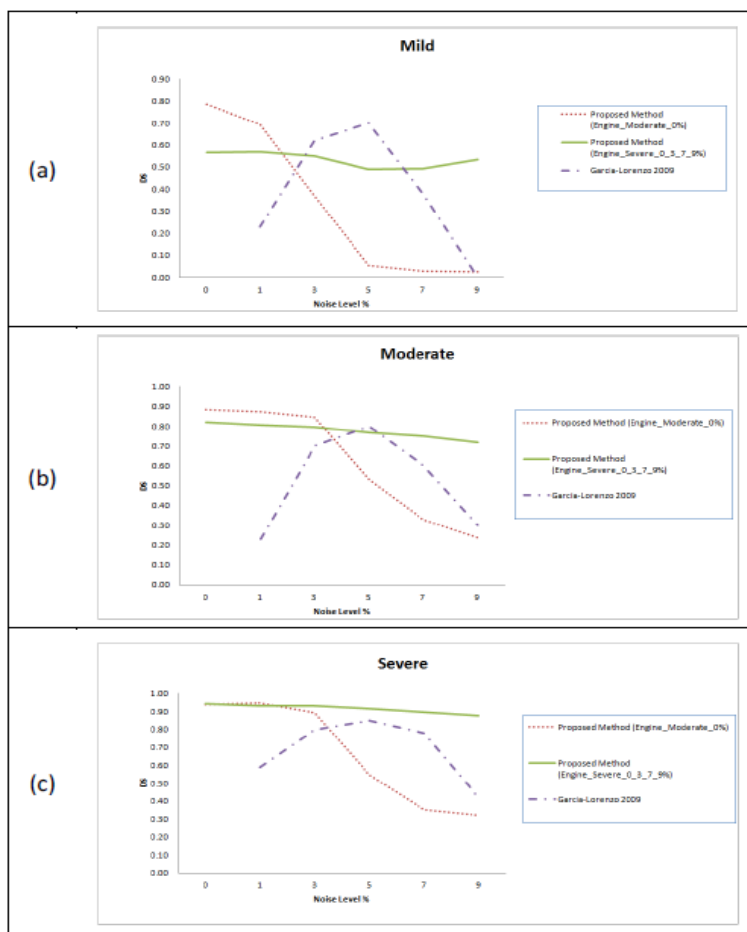
Figure 4-24: Comparison between the proposed method and a state of the art method.

BrainWeb datasets are used for different noise levels based on dice similarities for different subject templates. (a) Mild lesion level subject. (b) Moderate lesion level subject. (c) Severe lesion level subject.

Real Data

Datasets of 61 cases were used to verify the segmentation technique proposed in this dissertation. The sources of these datasets are the workshop of MS Lesion Segmentation Challenge 2008 (51 subjects) and real MRI datasets acquired for MS subjects at the University of Miami Miller School of Medicine (10 subjects). The subjects datasets of the MS Lesion Segmentation Challenge are categorized as twenty subjects provided for

training (10 from CHB and 10 from UNC) and thirty one subjects provided for testing (18 from CHB and 13 from UNC) and labels are not provided for the testing set.

In the training phase, two different engines were trained. The first engine (Engine_CHB) was trained using four subjects of the training dataset (CHB_train_Case01, CHB_train_Case02, CHB_train_Case06 and CHB_train_Case10) with each subject sharing 25% of the training set. The second engine (Engine_UNC) was trained using four subjects of the training dataset (UNC_train_Case02, UNC_train_Case03, UNC_train_Case09 and UNC_train_Case10) with each subject sharing 25% of the training set. As recommended by Anbeek et al. [20], CHB_train_Case04, CHB_train_Case05, CHB_train_Case09, UNC_train_Case01, UNC_train_Case05 and UNC_train_Case06 were avoided due to image and manual segmentation quality. For the UNC training set, only the manual segmentations of the CHB rater were used.

In the segmentation phase, three groups of subjects were tested. The first group is composed of the MRI dataset acquired for MS subjects at the University of Miami Miller School of Medicine. The second group is composed of the training set of CHB provided by MS Lesion Challenge. The third group is composed of the testing cases provided by MS Lesion Challenge. For each group, the segmentation is evaluated using the metrics that match those used in other methods using the MS Lesion Segmentation Challenge datasets to facilitate the comparison with other method for the purpose of the evaluation.

MS Subjects MRI data

The ten MRI studies for MS subjects acquired at University of Miami, referred to as MS2 to MS11, have been re-sliced to have the same resolution of the training set of Engine_CHB and registered to the MNI atlas as described in the preprocessing section. The segmentation of these datasets was performed to check the robustness of the proposed technique when training is done using one source of data with set of conditions of acquisition and segmentation is performed using a different source of data with different set of conditions. The dice similarity for the segmentation of the ten subjects is shown in Figure 4-25.

The X-axis provides the subjects ordered by the total lesion load (TLL) in ml to show the effect of this parameter on the segmentation performance. An approximate total lesion load is calculated by counting the number of MS voxles in the ground truth and multiplying by the voxel volume.
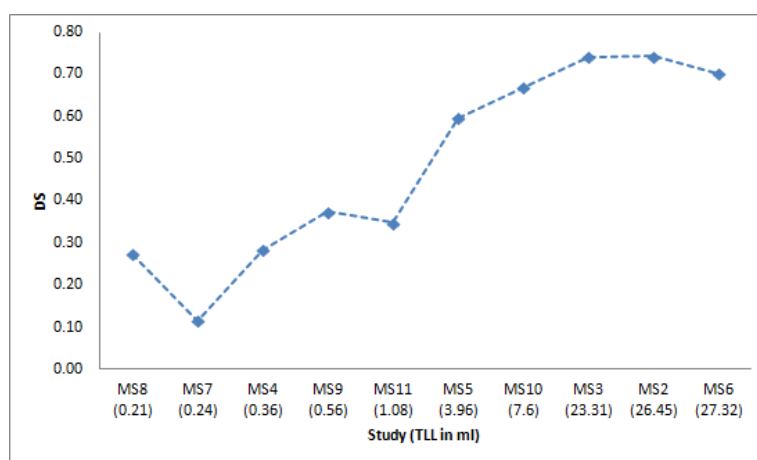
Figure 4-25: Dice similarity versus the total lesion load.

X-Axis gives the study name ordered by the total lesion load in ml.

The results shown in Figure 4-25 demonstrate the success of the proposed technique to deal with the case when training is done using one source of data and conditions of acquisition and segmentation is performed using a different source of data and different conditions. Results for these datasets show weak performance when dealing with MRI studies that have very low total lesion load (TLL) which leads to the conclusion that extra effort is still needed to specifically handle the mild MS lesions cases.

MS Lesion Segmentation Challenge Training Set Data

The ten CHB subjects provided by the MS Lesion Segmentation Challenge datasets are segmented by the proposed technique. For the purpose of comparison, segmentation results are evaluated using TPR and PPV in the same manner reported by Geremia et al. [13] who provides results for segmenting the training datasets. Table 4-9 shows a comparison between the proposed technique, Geremia et al. [13] and Souplet et al. [43] (the best result in the MS Lesion Segmentation Challenge at that time). The value marked in bold is the best metric value obtained.

Study Case	TPR			PPV		
	Souplet et al. 2008 [77]	Geremia et al. 2010 [13]	Proposed Method [76]	Souplet et al. 2008 [77]	Geremia et al. 2010 [13]	Proposed Method [76]
CHB_train_Case01	0.22	0.49	**0.73**	0.41	**0.64**	0.48
CHB_train_Case02	0.18	**0.44**	0.02	0.29	**0.63**	0.56
CHB_train_Case03	0.17	**0.22**	0.14	0.21	**0.57**	0.06
CHB_train_Case04	0.12	0.31	**0.48**	0.55	**0.78**	0.04
CHB_train_Case05	0.22	0.4	**0.44**	0.42	**0.52**	0.10
CHB_train_Case06	0.13	**0.32**	0.15	0.46	**0.52**	0.42
CHB_train_Case07	0.13	**0.4**	0.29	0.39	0.54	**0.54**
CHB_train_Case08	0.13	0.46	**0.76**	0.55	**0.65**	0.47
CHB_train_Case09	0.03	**0.23**	0.18	0.18	**0.28**	0.09
CHB_train_Case10	0.05	0.23	**0.38**	0.18	0.39	**0.43**

Table 4-9 : Comparison of the proposed method segmentation results with state of the art methods.

MS Lesion Segmentation Challenge Testing Set Data

After testing the technique with the available datasets that contain the ground truth, the test cases provided by MS Lesion Segmentation Challenge are segmented using the proposed method with the CHB test cases segmented using the Engine_CHB and the UNC test cases segmented using Engine_UNC for the purpose of comparison and competition with other competitors have done the same. The segmentation of the test

cases were uploaded to the MS Lesion Segmentation Challenge to get the automatic evaluation of the segmentation with team name (UM-ECE_team) and the results were posted and can be accessed from the official MS Lesion Segmentation Challenge results section[78]. A snapshot of results is depicted in Figure 4-26 with the performance metrics used explained in subsection 2.5.4. Currently, the proposed technique segmentation score is included in the best ten scores knowing that the proposed technique is the only technique that was tested using 31 test cases where the others were tested using only 25 test cases.

Ground Truth	UNC Rater								CHB Rater								Total
All Dataset	Volume Diff. [%]	Score	Avg. Dist. [mm]	Score	True Pos. [%]	Score	False Pos. [%]	Score	Volume Diff. [%]	Score	Avg. Dist. [mm]	Score	True Pos. [%]	Score	False Pos. [%]	Score	
UNC test1 Case01	66.2	90	14.7	70	18.6	62	63.6	71	50.6	93	10.1	79	25.0	66	31.8	90	78
UNC test1 Case02	-1.0	0	-1.0	0	-1.0	0	-1.0	0	-1.0	0	-1.0	0	-1.0	0	-1.0	0	0
UNC test1 Case03	55.3	92	3.8	92	16.2	61	26.0	94	42.3	94	3.2	93	19.1	62	12.0	100	86
UNC test1 Case04	100.0	85	128.0	0	0.0	51	0.0	100	100.0	85	128.0	0	0.0	51	0.0	100	59
UNC test1 Case05	86.0	87	11.7	76	19.0	62	31.2	91	68.3	90	9.3	81	30.4	69	37.5	87	80
UNC test1 Case06	99.9	85	80.7	0	0.0	51	100.0	49	99.5	85	82.0	0	0.0	51	100.0	49	46
UNC test1 Case07	96.2	86	28.0	42	3.3	53	20.0	97	91.1	87	20.9	57	10.0	57	20.0	97	72
UNC test1 Case08	44.8	93	9.6	80	19.1	62	53.3	77	9.8	99	6.0	88	55.6	83	46.7	81	83
UNC test1 Case09	256.2	62	48.2	1	0.0	51	100.0	49	402.6	41	53.4	0	0.0	51	100.0	49	38
UNC test1 Case10	100.0	85	128.0	0	0.0	51	0.0	100	100.0	85	128.0	0	0.0	51	0.0	100	59
UNC test1 Case11	98.2	86	15.3	68	3.2	53	25.0	94	98.4	86	13.2	73	4.8	54	0.0	100	77
UNC test1 Case12	100.0	85	128.0	0	0.0	51	0.0	100	100.0	85	128.0	0	0.0	51	0.0	100	59
UNC test1 Case13	143.5	79	29.3	40	0.0	51	100.0	49	153.9	77	16.1	67	33.3	70	83.3	59	62
UNC test1 Case14	103.3	85	10.6	78	44.4	77	64.7	70	113.4	83	15.2	69	25.0	66	82.4	60	73
CHB test1 Case01	243.7	64	8.1	83	45.3	77	78.0	62	391.0	43	10.5	78	77.4	95	83.9	59	70
CHB test1 Case02	445.1	35	9.6	80	77.3	95	92.7	53	132.3	81	4.4	91	84.2	99	87.6	56	74
CHB test1 Case03	112.0	84	15.0	69	50.0	80	93.5	53	2.4	100	12.6	74	53.3	82	91.9	54	74
CHB test1 Case04	80.2	88	19.0	61	27.3	67	76.5	63	90.5	87	24.2	50	16.7	61	76.5	63	68
CHB test1 Case05	11422.6	0	17.4	64	70.4	91	98.5	50	2085.8	0	11.8	76	78.3	96	97.8	50	53
CHB test1 Case06	173.6	75	3.6	93	75.0	94	96.9	51	185.9	73	3.8	92	45.5	77	98.3	50	75
CHB test1 Case07	140.2	79	7.7	84	41.7	75	75.0	64	46.1	93	2.8	94	50.0	80	50.0	79	81
CHB test1 Case08	19.7	97	20.4	58	18.5	62	71.4	66	46.2	93	21.4	56	11.8	58	66.7	69	70
CHB test1 Case09	775.8	0	4.9	90	80.5	97	84.4	58	638.6	6	4.4	91	67.3	90	86.7	57	61
CHB test1 Case10	754.2	0	10.2	79	68.4	90	90.1	55	317.7	53	3.6	93	69.0	91	77.2	63	65
CHB test1 Case11	1120.7	0	14.0	71	56.8	84	96.0	51	294.7	57	7.6	84	51.7	81	92.0	54	60
CHB test1 Case12	32.4	95	2.9	94	27.7	67	60.7	73	32.7	95	3.0	94	28.2	68	69.2	68	82
CHB test1 Case13	12.8	98	5.8	88	40.0	74	59.3	74	46.6	93	3.3	93	28.6	68	7.4	100	86
CHB test1 Case15	337.4	51	4.5	91	83.6	99	90.3	55	476.8	30	5.6	89	93.6	100	95.1	52	71
CHB test1 Case16	62.8	91	6.0	88	22.5	64	50.0	79	68.0	90	4.1	92	40.0	74	47.2	81	82
CHB test1 Case17	123.0	82	7.3	85	32.1	70	58.9	74	9.4	99	3.4	93	22.0	64	43.8	83	81
CHB test1 Case18	71.5	90	62.5	0	0.0	51	100.0	49	7.4	99	58.8	0	0.0	51	100.0	49	49
All Average	557.3	69	27.5	59	30.3	67	63.1	67	203.3	74	25.7	63	32.9	68	57.5	70	67
All UNC	96.3	79	45.3	39	8.8	53	41.6	74	102.1	78	43.7	43	14.4	56	36.6	77	62
All CHB	936.9	60	12.9	75	48.1	79	80.7	60	286.6	70	10.9	79	48.1	79	74.8	64	71

Figure 4-26: Snapshot of the results table generated automatically by the MS Lesion Segmentation Challenge workshop evaluation software for the segmentation of the test cases.

For the test done on the datasets obtained from MS Lesion Challenge, two engines were trained namely Engine_CHB and Engine_UNC for the purpose of comparison and competition with other competitors have done the same. However, having a unified engine in clinical practice will provide better performance when dealing with other subjects.

Chapter 5
Customized MV-T-SVM for Lesions and Tissues Detection in MRI

This chapter presents the enhancements developed to customize multi-sectional views textural based SVM (pipeline 3) approach presented in the end of Chapter 4 to provide optimized performance in segmentation of tissues and lesions in MRI. The proposed technique is referred to as customized multi-sectional views textural based SVM (MV-T-SVM).

In the customized MV-T-SVM, enhancements are needed to overcome the limitations of the pipeline 3. First, pipeline 3 shows low performance when tested on datasets with low TLL. Also, the technique does not take into consideration the anatomic properties of the brain area that can be determined from the MNI atlas and thus using the probability based on statistics of having MS lesion in a specific area. Lesion regions shape improvement in post-processing does not take the contouring properties into consideration. The textural features are based on FLAIR sequence and does not benefit from the lesion pattern in the other sequences in multi-channels MRI. Performance of using SVM to learn lesion properties is affected by imbalanced training set due to the relative size of the lesion with respect to the normal brain tissues. Finally, using SVM to learn lesion blocks properties within a slice does not benefit from the 3D information of the MRI in learning.

125

The enhancements involve benefiting from the tissues and lesions specific information in the training and prediction phases of the classifier. The customized MV-T-SVM uses 3D nature of the MRI information, template tissue information provided for images which are registered to an anatomical atlas as well as textural information from the multi-channels images. Also, SVM learning procedure is revised to address imbalanced classes and to synchronize the learning of the three views engines. In addition, the multi-views aggregate function is replaced by variable aggregate operator instead of the naive Bayesian formula.

Although the customizations made in the MV-T-SVM are made to enable the technique to best deal with all tissues and lesions detection in MRI, but the concept is mainly applied on MS-Lesions which is the core of this dissertation without loss of generality. The twenty real subjects' datasets provided by MS Lesion Challenge are used to train, get statistics, priors, criteria, and thresholds of the customized MV-T-SVM while the ten real subjects' datasets acquired at University of Miami are used for testing the technique. The ability of the customized MV-T-SVM to deal with other tissues and lesions detection is tested using the publicly available datasets in the CAUSE07 challenge [**79**] which aim to get best performance in detection of caudate.

5.1 Statistical Textural Features in Multi-Channels Images

The statistical textural features used in the three segmentation pipelines introduced in Chapter 4 are extracted from FLAIR images. These textural features extraction formulas are provided in section 2.4. Changes proposed in features extraction to benefit from the multi-channels information are mentioned in this section.

We assume that the medical (brain) image is available in L channels and that features are calculated for each square block of area w^2. The intensity of each image channel k is given by $f_k(x,y,z)$ of three space variables x, y, and z and the function f can take any value i=0,1,…. $(G_k -1)$ where G_k is the number of intensity levels in the image channel k. For simplicity the dimension z is neglected in the formulas given that the textural features are extracted for a slice in certain section view which leads to constant z.

In the following subsections, the updates proposed to the definitions of textural features provided in section 2.4 are explained to each category of features.

5.1.1 Multi-Channels Histogram based Features

This category of textural features was updated to be extracted from all channels instead of single channel. The intensity level histogram is a function $h_k(i)$ providing, for each intensity level i, the number of pixels in the whole square block having this intensity in the image channel k:

$$h_k(i) = \sum_{x=0}^{w-1}\sum_{y=0}^{w-1} \delta(f(x,y),i) \; ; \delta(i,j) = \begin{cases} 1, & j = i \\ 0, & j \neq i \end{cases} \tag{5-1}$$

The histogram is a concise and simple summary of the statistical information contained in the image. Dividing the histogram $h_k(i)$ by the total number of pixels in the image provides the approximate probability density of the occurrence of the intensity levels $p_k(i)$, given by:

$$p_k(i) = h_k(i) / NM \tag{5-2}$$

The following set of textural features is calculated from the normalized histogram for each image channel for a certain square block:

$$\text{Mean: } \mu_k = \sum_{i=0}^{G-1} i p_k(i) \tag{5-3}$$

$$\text{Variance: } \sigma_k^2 = \sum_{i=0}^{G-1} (i-\mu)^2 p_k(i) \tag{5-4}$$

5.1.2 Multi-Channels Gradient based Features

This category of textural features was updated to take the neighboring slices and the other channels into consideration.

The gradient matrix element g(x,y) is defined for each pixel in the image square block based on the neighborhood size. For a 3x3x3 pixels neighborhood, g is defined as follows:

$$\Delta_x = \sum_{k=1}^{L} \left| f_k(x-1,y,z) - f_k(x+1,y,z) \right|$$
$$\Delta_y = \sum_{k=1}^{L} \left| f_k(x,y-1,z) - f_k(x,y+1,z) \right| \tag{5-5}$$
$$\Delta_z = \sum_{k=1}^{L} \left| f_k(x,y,z-1) - f_k(x,y,z+1) \right|$$
$$g(x,y) = \sqrt{\Delta_x^2 + \Delta_y^2 + \Delta_z^2}$$

The following set of textural features is calculated from the updated gradient matrix:

$$\text{Mean of absolute gradient (GrMean)} = \frac{1}{w^2} \sum_{x=0}^{w-1} \sum_{y=0}^{w-1} g(x,y) \qquad (5\text{-}6)$$

$$\text{Variance of absolute gradient (GrVariance)} = \frac{1}{w^2} \sum_{x=0}^{w-1} \sum_{y=0}^{w-1} (g(x,y) - GrMean)^2 \qquad (5\text{-}7)$$

5.1.3 Multi-Channels Run length matrix based Features

The run length matrix is defined for a specific direction. Usually a matrix is calculated for the horizontal, vertical, $45°$, and $135°$ directions. The matrix element r(i,j) is originally defined as the number of times there is a run of length j having gray level i.

We made the following changes to the definition of Run length matrix:

- For a certain image channel k, the Gray levels G_k are divided into R_k ranges (Each range covers G_k / R_k gray levels).

- A pixel in a block can fall in ranges R_1, R_2, …. R_L. These ranges for a single pixel are defined as a gray level range intersection.

- The number of possible gray level ranges intersections R is given by:

$$R = \prod_{k=1}^{L} R_k \qquad (5\text{-}8)$$

- The matrix element r(i,j) is defined as the number of times there is a run of length j falling in the gray level intersection i.

The following set of textural features calculated from the multi-channels run length matrix is used in the feature vector of customized MV-T-SVM:

$$\text{Multi-channels Gray level non-uniformity (GLevNonUni)} = (\sum_{i=0}^{R}(\sum_{j=1}^{N_r}r(i,j))^2)/C \quad (5\text{-}9)$$

Multi-channels Run length non-uniformity (RLNonUni)

$$= (\sum_{j=1}^{N_r}(\sum_{i=0}^{R}r(i,j))^2)/C \quad (5\text{-}10)$$

where the normalization coefficient C is defined as follows:

$$C = \sum_{i=0}^{R}\sum_{j=1}^{N_r}r(i,j) \quad (5\text{-}11)$$

5.1.4 Multi-Channels Co-occurrence matrices based Features

The co-occurrence matrix is defined for certain angle θ and certain distance d. The matrix element $h_{d\theta}(i,j)$ is the number of times $f(x1,y1) = i$ and $f(x2,y2) = j$ where $(x2,y2)=(x1,y1) + (d\cos\theta, d\sin\theta)$.

Instead of extracting features for each square block based on a co-occurrence matrix for single channel image, we calculated a co-occurrence matrix for each channel $CO_1, CO_2, \ldots CO_L$ and modified the formulas of textural extraction to use all of the matrices.

The following set of textural features calculated from the multi-channels co-occurrence matrices are used in the feature vector of customized MV-T-SVM:

$$\text{Multi-channels Absolute Value} = \sum_{k=1}^{L}\left(\sum_{i=0}^{G-1}\sum_{j=0}^{G-1}|i-j|(Co_k(i,j))^2\right) \quad (5\text{-}12)$$

$$\text{Multi-channels Contrast} = \sum_{k=1}^{L}\left(\sum_{i=0}^{G-1}\sum_{j=0}^{G-1}(i-j)^2\,(Co_k(i,j))^2 \right) \qquad (5\text{-}13)$$

$$\text{Multi-channels Entropy} = -\sum_{k=1}^{L}\left(\sum_{i=0}^{G-1}\sum_{j=1}^{N_r}co(i,j)\log_2(co_k(i,j)) \right) \qquad (5\text{-}14)$$

5.2 Regional Segmentation

The MNI atlas provides anatomical labels for each voxel in the brain as demonstrated in Figure 5-1 with each anatomical label is represented by a unique color. Since the datasets used in this dissertation are already registered to this atlas, the anatomical information can be used to enhance the proposed technique to get more accurate results.

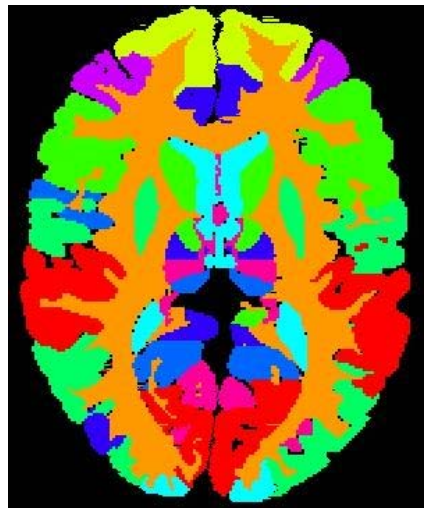**Figure 5-1: Atlas Discrete Labels for a brain slice.**

Applying statistical analysis on the ground truth of the real subjects' datasets provided by MS Lesion Challenge using the anatomical labels atlas; results in the distribution of MS voxels in different regions of the brain as shown in Figure 5-2. Most of the MS lesions (66%) occur in the White Matter tissue, which is the highest rate of MS

lesions voxels among all labels. (In Figure 5-2, White Matter is removed from the chart to avoid dominating the other percentages).

Some brain regions have high percent of MS lesions voxels. Other regions of the brain have no MS voxels and this is logical due to the anatomical properties of these regions which are not vulnerable to such type of lesions. On the other hand, some brain regions have non-zero percentage of MS voxels although the anatomical structure cannot hold such lesions but the MS lesions can occur in the adjacent tissues while the atlas provides wrong label due to errors in the registration.
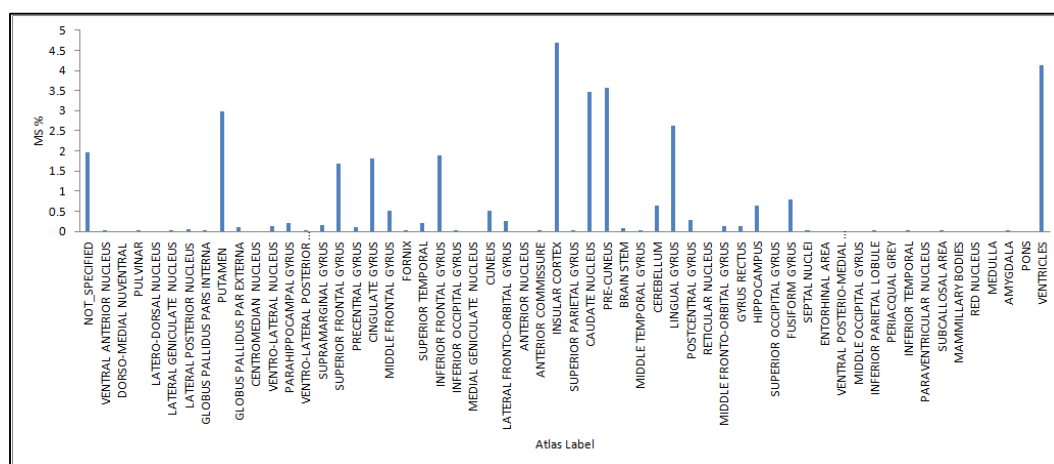
Figure 5-2: Distribution of MS lesion on the brain regions according to analysis made on training dataset registered to MNI atlas.

We used this percent as the likelihood $P(\text{Tissue}(v)|MS_1)$ where $\text{Tissue}(v)$ is the label assigned to voxel v using the MNI atlas. Using the naive Bayesian decision rule, the posterior $P(MS_1|\text{Tissue}(v))$ can be calculate based on the likelihood obtained from the statistical analysis on the twenty real subjects. Assuming two-class problem with the classes MS_1 (lesion voxel) and MS_0 (healthy voxel), the posterior is given by:

$$P(MS_1|Tissue(v)) = \frac{P(MS_1).P(Tissue(v)|MS_1)}{P(MS_1).P(Tissue(v)|MS_1) + P(MS_0).P(Tissue(v)|MS_0)} \qquad (5\text{-}15)$$

By calculating posterior for all tissues, we get $P(MS_1|Tissue(v))$ as shown in Figure 5-3.

Figure 5-3: Calculated posterior based on the anatomical tissues likelihood.

Errors in registration are expected due to difference in brain sizes and difference in region sizes from subject to another, we do not base the classification directly on the calculated posterior but it is used as a threshold in the aggregation step in the customized MV-T-SVM as presented later in this chapter.
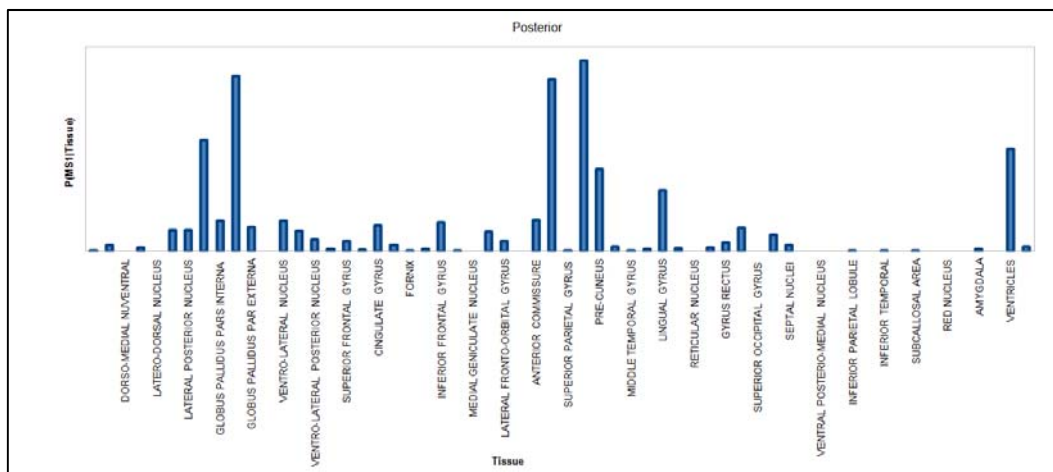
5.3 MS Lesion Customized Learning Engine

In this section modification are proposed to update the learning of the classification engine to be customized to the MS lesions problem. The purpose from the first modification is to enhance the integration between the three sectional views classification engines. The other modification is done to overcome the imbalanced data effect on learning.

5.3.1 Overlapping Learning of the Multi-Views SVM's

In multi-sectional views pipeline explained in Chapter 4, the three SVM engines are trained independently using the training set obtained by dividing the slices of the corresponding views into blocks with the texture features of each block are mainly used to create a feature vector and the class assigned to each block is either MS-block or non-MS block.

In the aggregation step after the three views segmentation, the variable X_v assigned to each voxel v can take the discrete values X_0, X_1, X_2 or X_3 which indicates being classified as MS voxel either 0, 1, 2 or 3 times in the different views segmentations respectively. The value assigned to X_v highly affects the inclusion of the voxel v to the aggregate set G. The worst case classification occur when the true class of the voxel is negative while $X_v=3$ (worst case false positive) and similarly when the true class of the voxel is positive while $X_v=0$ (worst case false negative). The reason of considering these cases as worst case scenario is the difficulty of recovering the error with later steps in the segmentation framework due to the misguiding voting resulting from the three views segmentation.

The proposed modification to the learning scheme aims to minimize the number of times that the three views classification engines result in misguided voting (False positive with $X_v= X_3$ or False negative with $X_v= X_0$). We will refer to this count as Misguiding Voting Error (MVE).

According to our approach in pipeline 3, the three sectional-views engines were trained to classify square blocks of area d^2 pixels. In the training phase, no information about the single pixel was considered. The geometrical intersection from 3 blocks from 3 different views results in a single voxel.

Initially, assuming that the sagittal and coronal SVM were trained, the current activity will be training the axial SVM to accurately classify block of axial slices while minimizing the misguiding voting error (MVE). Assuming that the two intersecting blocks S and C from sagittal and coronal slices respectively were classified by the corresponding two SVM's, the intersection of these two blocks generates a line SC as shown in Figure 5-4. According to multi-view segmentation, before axial SVM classification, X_v of each voxel of the line SC may be assigned a value from X_0, X_1, or X_2 according to the other two views segmentation.
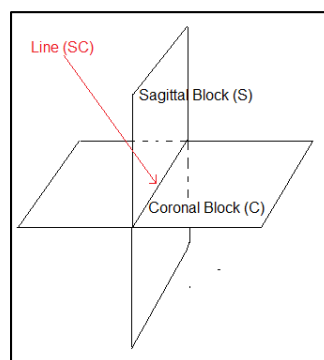

Figure 5-4: Line SC resulting from intersection of Sagittal Block (S) and Coronal Block (C).

When axial blocks of axial slices intersecting with the line SC are classified by the corresponding SVM, the variable X_v corresponding to the intersection voxel v may be incremented to the higher discrete level or remains the same according to the class assigned to the axial block.

The decision taken for the axial block by the axial classifier is taken based on the engine trained by the features vector extracted per block. It is required to bias the axial classifier to classify the axial block that intersects with the line SC in a way that minimizes MVE. For the block that intersects with SC in a voxel v, if the current $X_v = X_2$ while the ground truth of this specific voxel is non-MS, then it is required from axial classifier to classify the axial block as non-MS block to avoid having $X_v = X_3$ for non-MS voxel. Similarly, for the block that intersects with SC in a voxel v, if the current $X_v = X_0$ while the ground truth of this specific voxel is MS, then it is required from axial classifier to classify the axial block as MS block to avoid having $X_v = X_0$ for MS voxel. This can be represented by the variable $Bias_v$ given by the following equation:

$$Bias_v = \begin{cases} +1, & GT(v) = 1 \ \& \ X_v = X_0 \\ -1, & GT(v) = 0 \ \& \ X_v = X_2 \\ 0, & otherwise \end{cases} \tag{5-16}$$

where GT(v) is the ground truth of the voxel v. This variable indicates the bias required to be taken into consideration in the remaining view segmentation based on the current X_v resulting from segmentation from the other two views.

For axial block i composed of d^2 pixels, each of these pixels should be taken into consideration to minimize the overall MVE.

To train the axial classifier, taking into consideration the sagittal and coronal classifier, the subject training dataset is segmented first by the trained sagittal and coronal dataset.

The effect of segmentation of pixels of the block i using the other segmentation engines can be represented by Bias Count Modifier variable (BCM_i) which can be calculated by:

$$BCM_i = sign(\sum_{v \in i} Bias_v)$$
(5-17)

This variable will be positive if majority of voxels of the block are biasing the block class to vote as positive to rescue them from having $X_v = X_0$ while they are positive. Similarly, BCM_i will be negative if majority of voxels of the block are biasing the block class to vote as negative to rescue them from having $X_v = X_3$ while they are negative. The variable will be zero if all voxels are not in the critical situation or it is balanced biasing where the number of voxels biasing to positive equals the number of voxels biasing to negative.

When the subject dataset is used to train the axial SVM, each axial block i having feature vector x_i, the ground truth class of the block y_i is originally has value 1 (MS-block or contains at least 1 MS pixel) or -1 (non-MS or contains non-MS pixel). To minimize MVE, the block class y_i need to be updated according to the following formula:

$$y_i = sign(y_i + 2BCM_i)$$
(5-18)

According to (5-18), if BCM_i is zero, the block class y_i will remain the same, while the class block will be biased to +1 if BCM_i is positive and similarly it will be biased to -1 if BCM_i is negative whatever the original y_i.

After training the axial sectional view SVM, a new iteration of learning is performed to modify the coronal sectional view SVM to minimize MVE based on results of segmentation of the training set by the axial SVM and sagittal SVM considering the individual voxels ground truth. The same will be done to update the sagittal view SVM based on the other two sectional views classifier. More training iterations can be made and derivations to stopping condition can be done, but the time complexity of the processing will be infeasible; thus we recommend using fixed number of iterations.

Modes of learning

In order to provide the axial SVM with trained sagittal SVM and coronal SVM, the last two engines should be trained in non-overlapping learning mode while the axial engine is trained in overlapping mode. The overall learning of MV-T-SVM is overlapping scheme but the individual engines should support the two modes of learning.

In non-overlapping learning, the input training entries are pairs (x_i, y_i) where x_i is the feature vector of the block in the corresponding sectional view and y_i is the class of the block which is either MS block(class +1) if at least one pixel is MS in ground truth and non-MS block (class -1) if all pixels are non-MS in ground truth. In the non-overlapping learning process, y_i is never modified.

In overlapping learning, the input training entries are composed of three groups:

- Pairs (x_i, y_i) where x_i is the feature vector of the block in the corresponding sectional view and y_i is the class of the block which is either MS block(class +1) if at least one pixel is MS in ground truth and non-MS block (class -1) if all pixels are non-MS in ground truth.

- The class of the individual pixels of each block entry provided from the ground truth of the subject dataset.

- The segmentation result of the training subject dataset using the other two sectional views' engines.

In the overlapping learning process, y_i is modified using equation (5-18).

5.3.2 Dealing with Imbalanced Data

Customization of the learning scheme to best fit the MS lesion detection includes dealing with the imbalanced data problem. In lesion or tissue detection, the positive class is always much less than the negative class which represents healthy voxels (in case of lesion detection) or the remaining of the organ (in case of tissue detection). We used the recommendation of the libsvm authors to use two different penalty parameters in the SVM formulation. Thus the formulation of the optimization problem of the SVM given in equation (2-31) is updated to the following equation:

$$minimize_{w,b} \qquad \frac{1}{2}\|w\|^2 + C^+ \sum_{y_i=1} \varepsilon_i + C^- \sum_{y_i=-1} \varepsilon_i \qquad (5\text{-}19)$$

$$subject\ to: y_i(W^T X_i + b) \geq 1 - \varepsilon_i, \ \varepsilon_i \ \geq 0, i = 1, \dots, n$$

The learning of each SVM is updated to solve the optimization problem (5-19) instead of single penalty parameter problem (2-31).

5.4 Rule based Multi-Sectional Views Aggregate function

The aggregate function used in multi-sectional views pipeline (pipeline 3) uses the threshold function defined in equation (4-9) that is based on the posterior $P(MS_1|X_v)$ calculated using the naive Bayesian decision rule in equation (4-7). More customization is made to the multi-sectional views segmentation technique by updating the aggregate function.

The updates proposed to the aggregate function aim to deal with the following problems:

1. Minimize the false positive in case of low TLL subjects which dominate the segmentation performance.
2. Take in consideration the posterior probability of MS lesion region based on brain anatomical region given by equation (5-15)

In case of subjects of low TLL, the low performance of the segmentation is mainly because the effect of false positives that dominate the true positive voxels. Although similar rate of false positive occurs with segmentation of high TLL but the domination of true positive results in good performance in terms of dice similarity. According to tests made on the training sets, the segmentation performance of the low TLL subjects was improved by at least 10% in terms of dice similarity when the AND operator is applied on the three segmentation sets A, B, and C.

An estimate of the total lesion load (TLL) can be determined before applying the aggregate function by taking the average of the number of voxels segmented as MS voxels in the three segmentation sets A,B, and C.

The aggregate function of customized MV-T-SVM depends on the posterior $P(MS_1|X_v)$, the posterior $P(MS_1|Tissue(v))$ and estimate of TLL of the dataset to be segmented. Each of the three factors is compared to thresholds specific for each parameter to enable rule based decision. $P(MS_1|X_v)$ thresholds are PX_{VH} (very high), PXV_H (high) and PX_{VL} (very low). Similarly, $P(MS_1|Tissue(v))$ thresholds are PT_{VH} (very high) and PT_{VL} (very low). The average TLL threshold is TLL_{VL} (very low).

The new aggregate function is given by the rule based function listed in Figure 5-5. Rule 1 makes the aggregate function include the voxel v in the aggregate set G only if the posterior $P(MS_1|X_v)$ is very high in case of low TLL or in case of voxels falling in brain regions that has very low posterior $P(MS_1|Tissue(v))$. Rule 2 makes the aggregate function include the voxel in the aggregate set G only if the posterior $P(MS_1|Tissue(v))$ is very high in cases of low $P(MS_1|X_v)$. The goal of this rule is to give opportunity to voxels appearing in only one sectional view segmentation to be added to the aggregate set if its posterior probability based on registered tissue is highly to have MS lesion. Rule 3 is to apply the same rule proposed in equation (4-9) in the common cases.
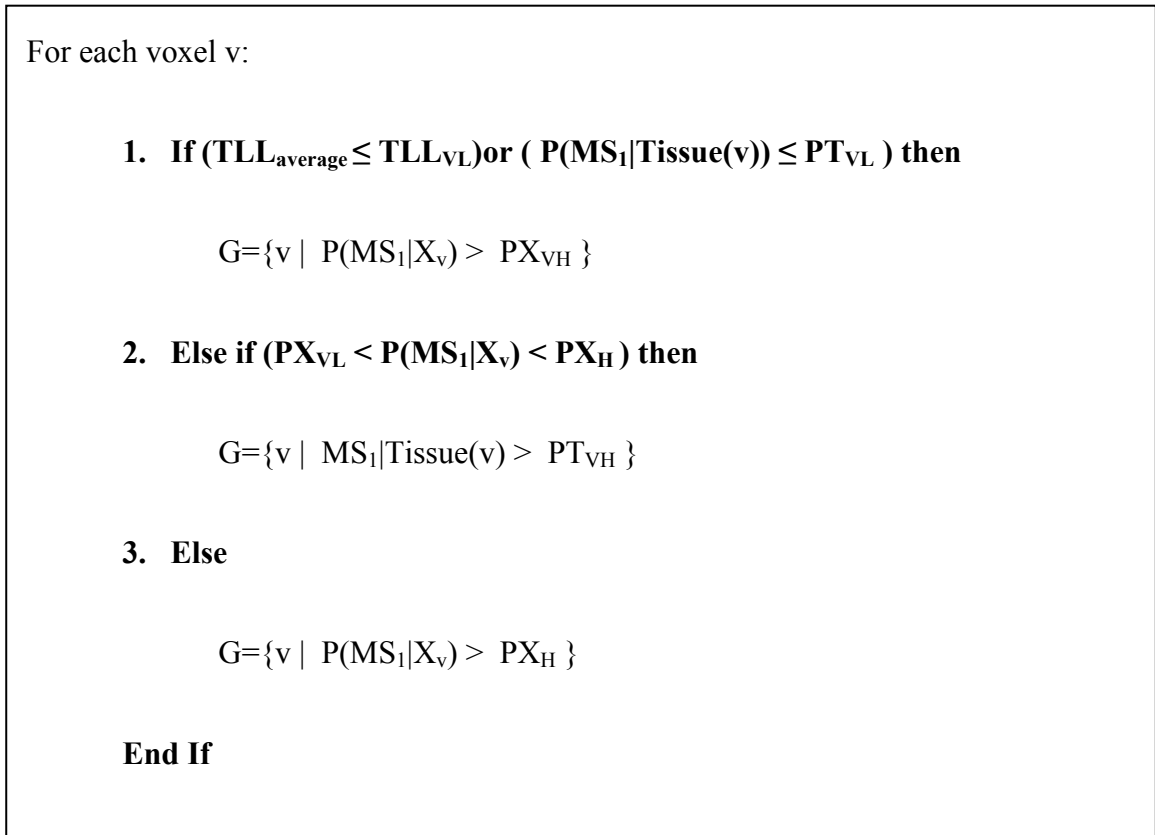
For each voxel v:

    1. If $(TLL_{average} \leq TLL_{VL})$ or ($P(MS_1|Tissue(v)) \leq PT_{VL}$) then

        $G=\{v \mid P(MS_1|X_v) > PX_{VH}\}$

    2. Else if $(PX_{VL} < P(MS_1|X_v) < PX_H)$ then

        $G=\{v \mid MS_1|Tissue(v) > PT_{VH}\}$

    3. Else

        $G=\{v \mid P(MS_1|X_v) > PX_H\}$

    End If

Figure 5-5: Rule based Multi-View Aggregate Function.

5.5 Automatic Contouring Algorithm

According to our approach in generating the final segmentation of the MS lesions of the subject test case, the connected pixels are grouped and form a lesion region. Due to the square block based segmentation in all sectional views, the lesion region contour is not smooth in most case but forms staircase contour with some random lines outing from the lesion region due to multi-view aggregation.

We propose a contouring module to improve the shape of the resulting lesion regions. The purpose of the contouring module other than having smoother lesion region

is also to trim extra segmented pixels (false positives) on the lesion region boundary and to recover missing pixels adjacent to the lesion region boundary (false negatives).

The proposed contouring algorithm determines for each lesion region three basic contours. The first is the initial contour which is the boundary of the lesion region in the aggregate segmentation. The second contour is the smallest possible contour which is the boundary of the lesion region resulting from applying AND operator on the three views segmentation. The third contour is the largest possible contour which is the boundary of the lesion region resulting from applying OR operator on the three views segmentation. The determination of the basic contours for a lesion region is illustrated in Figure 5-6.

The contouring algorithm processes the initial contour pixels. The processing results in reformation of the initial contour without crossing the smallest contour to the interior when shrinking and without crossing the largest contour to the exterior when expanding. The pseudocode of the proposed contouring algorithm is listed in Figure 5-7.

The routines BoundaryTest(b,M1,M2) compares the grayscale of the pixel b with the region mean (M1) and the neighborhood mean(M2). It takes decision of Keep if grayscale of b is suitable to be boundary between M1 and M2. BoundaryTest takes decision Grow when b is closer to M2 than M1 which means that b should be exterior pixel. BoundaryTest takes decision Shrink when b is closer to M1 and also M2 is close to M1 which means that b should be an interior pixels.
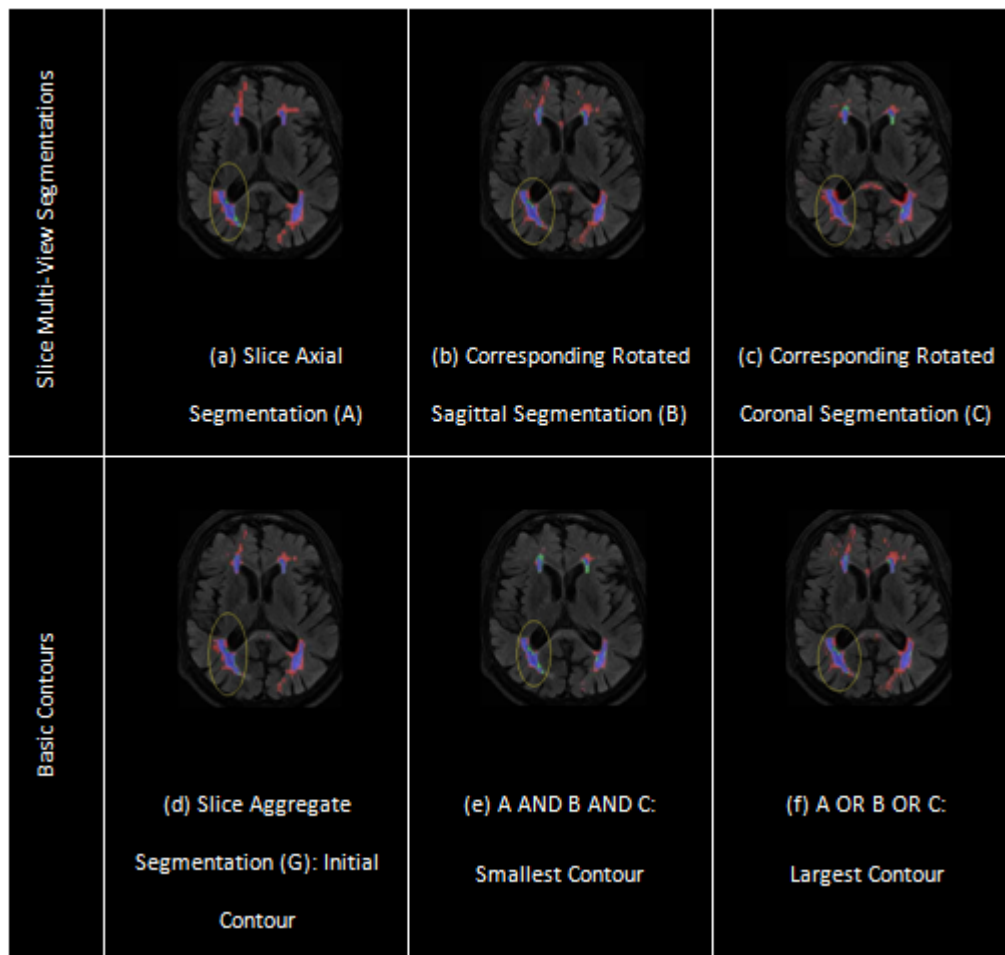
Figure 5-6: Multi-Views segmentations (A, B and C) sand Basic Contours determination.

The routine StopRegion represents the stopping condition and it stops when no longer formation is possible to the boundary of the lesion region without crossing the I_{min} or I_{max}.

The efficient application of the contouring algorithm results in minimizing the false positive and false negative pixels on the boundary of the lesion regions. Besides, it gives *clinic-friendly* output. An example of the output of applying the contouring algorithm on a sample slice is shown in Figure 5-8.

For each slice:

1   Label the lesion regions from the connected MS-Pixels in the multi-views' segmentation A,B and C and in the aggregate segmentation G.

2   For each lesion region:

    2.1 Determine the Initial Contour $I_{initial}$, I_{min} and I_{max}:

        2.1.1 $I_{initial}$ = Boundary (Lesion Region in G)

        2.1.2 I_{min} = Boundary (Lesion Region in A AND B AND C)

        2.1.3 I_{max} = Boundary (Lesion Region in A OR B OR C)

    2.2 Calculate the mean GS of the lesion region(G) in FLAIR: M1

    2.3 Calculate the mean GS of the neighborhood of the lesion region(G) in FLAIR: M2

    2.4 Form a linked list of the boundary pixels of $I_{initial}$.

    2.5 For each consecutive pixels a, b and c in the linked list of $I_{initial}$:

        2.5.1   if BoundaryTest(b,M1,M2) is Keep then

            2.5.1.1 Keep b

        2.5.2   Else if BoundaryTest(b,M1,M2) is Shrink then

            2.5.2.1 Select an internal pixel d from the lesion region that can connect a and c to be a boundary pixel

            2.5.2.2 If d is outside the smallest contour then

                2.5.2.2.1   Replace b with d

                2.5.2.2.2   Remove b from lesion region

            End If

        2.5.3   Else if BoundaryTest(b,M1,M2) is Grow then

            2.5.3.1 Select an external pixel d from the lesion region that can connect a and c to be a boundary pixel

            2.5.3.2 If d is inside the largest contour then

                2.5.3.2.1   Add d to lesion region

                2.5.3.2.2   Replace b with d

            End If

        End if

    2.6 Repeat step 2.4 until **StopRegion(lesion region) is true**

    2.7 Mark the holes in the I_{final} as MS pixel based on Lesion Continuity Concept.

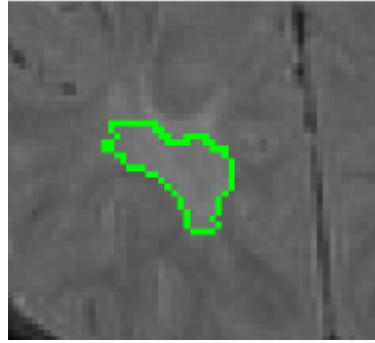**Figure 5-7: Proposed Lesion Region Contouring Algorithm.**

Figure 5-8: MS Lesion boundary generated by the proposed automatic contouring algorithm.

5.6 Experimental Results

In this section the segmentation performance of customized MV-T-SVM is evaluated. The twenty real datasets of the MS Lesion Segmentation Challenge were used for training and setting the thresholds while the ten real datasets acquired at University of Miami (UM datasets) were used for testing. To test the ability of the technique to be generalized to do other tissues segmentation, the caudate segmentation was tested using MV-T-SVM with settings updated to match the dataset settings.

5.6.1 MS Lesions Segmentation

The effect of the enhancements applied to pipeline 3 to have the customized MV-T-SVM segmentation framework was evaluated through the testing of UM datasets. First, the overlapping learning scheme of the three SVM engines was evaluated by analysis of its effect on $P(X_v| MS_1)$ and $P(X_v| MS_0)$ as a measure of minimizing the MVE. In Figure 5-9 (a), the $P(X_0| MS_1)$ is reduced which reduces misguiding voting in multi-view segmentation that results in reporting an MS voxel as non-MS in all views. Similarly, In Figure 5-9 (b), the $P(X_3| MS_0)$ is reduced which reduces misguiding voting in multi-view segmentation that results in reporting an non-MS voxel as MS voxel in all views. In

Figure 5-9 (b), the $P(X_0| MS_0)$ is removed from the chart because of its very large value (most voxels are healthy and segmented as healthy) to avoid dominating the other values.

Second, the effect of replacing the single channel based textural features with the proposed multi-channels textural based features was evaluated by drawing the ROC curve for the axial views segmentation SVM and comparing it with the ROC curve drawn before in section 4.8.1. In Figure 5-10, the comparison between customized MV-T-SVM and pipeline 3 shows the improvement of using the proposed multi-channels textural features on the blocks classification performance.
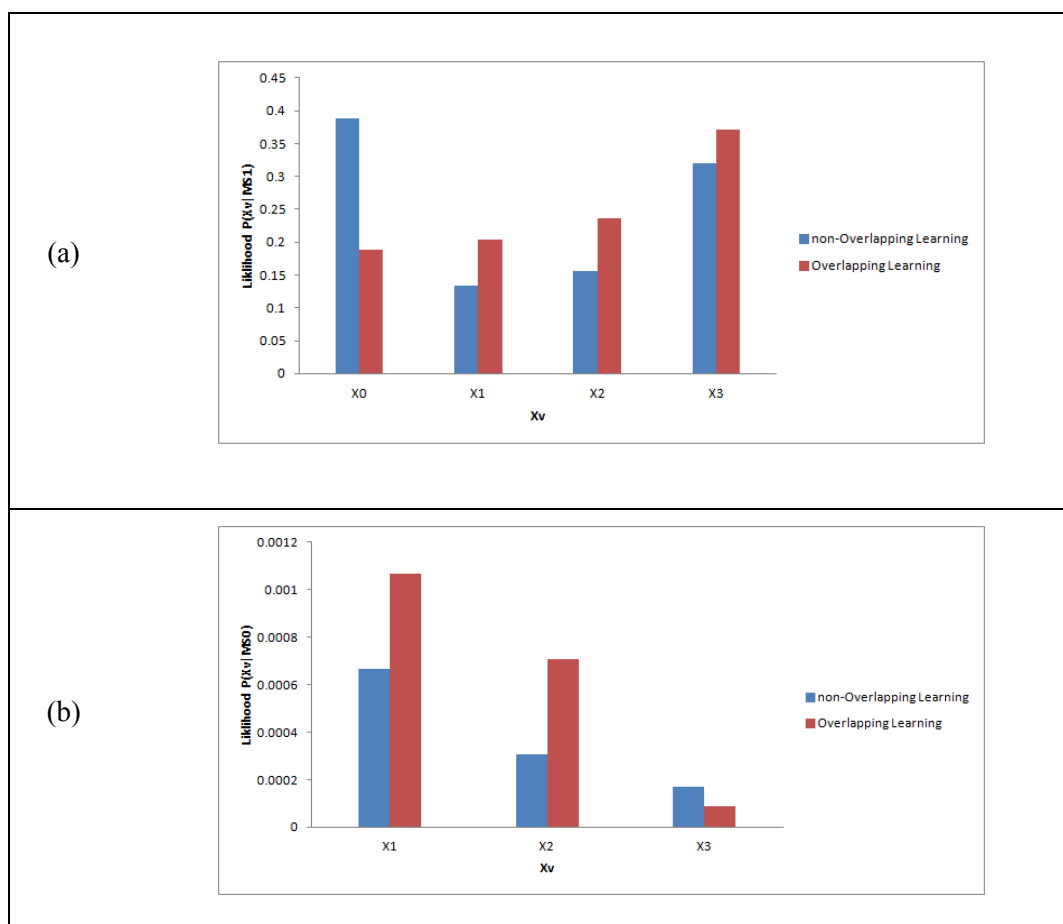

Figure 5-9: Effect of overlapping learning of three views engines on MVE.
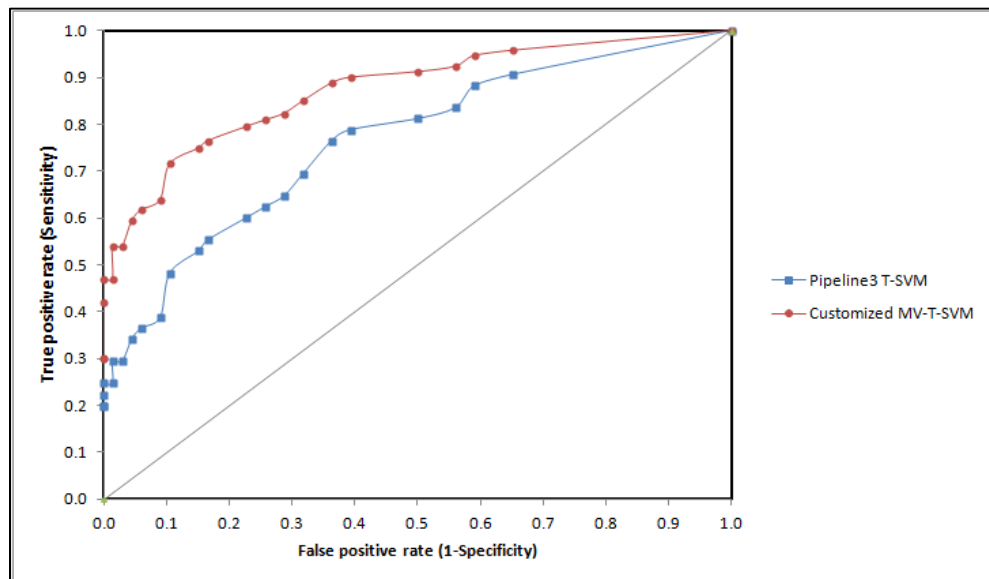**(a) $P(X_v|MS_1)$ and (b) $P(X_v| MS_0)$.**

Figure 5-10: Effect of using multi-channels textural features on the ROC curve compared with Pipeline 3.

Third, the effect of applying the proposed contouring algorithm on the aggregate segmentation lesion region was evaluated quantitatively by measuring the amount of false positive voxels removed from boundary versus the false negative voxels added when the contouring algorithm results in shrinking of the lesion region. Also evaluation involves measuring the amount of false negative voxels removed from boundary versus the false positive voxels added when the contouring algorithm results in growing the lesion region. The effect of contouring algorithm on the lesion region boundary is shown in Figure 5-11. The y-axis percentage is the percentage of the total number of voxels removed or added to the total areas of lesion regions. For shrinking function, the percentage of false positive voxels removed from boundary is 20% versus 5% of false negative voxels added. For growing function, the percentage of false negative voxels recovered is 7% versus 2% of false positive voxels added.
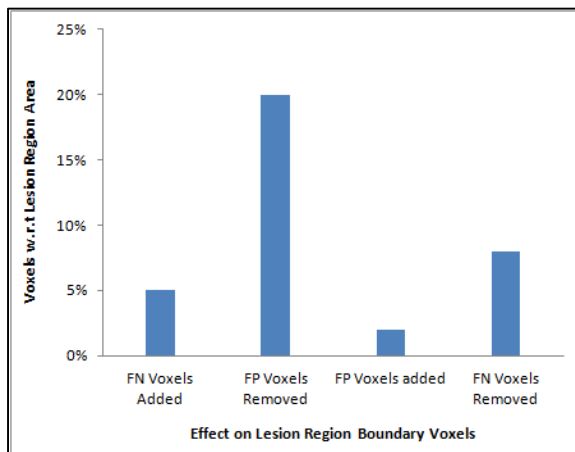
Figure 5-11: Effect of contouring algorithm on lesion region boundary voxels.

The segmentation of the ten real subjects acquired at University of Miami using customized MV-T-SVM was evaluated by measuring the dice similarity in comparison with the previous measure done using the pipeline 3. Figure 5-12 shows the result and comparison. Better result is obtained for most of studies with average 9%. The effect of low TLL on the performance is reduced.
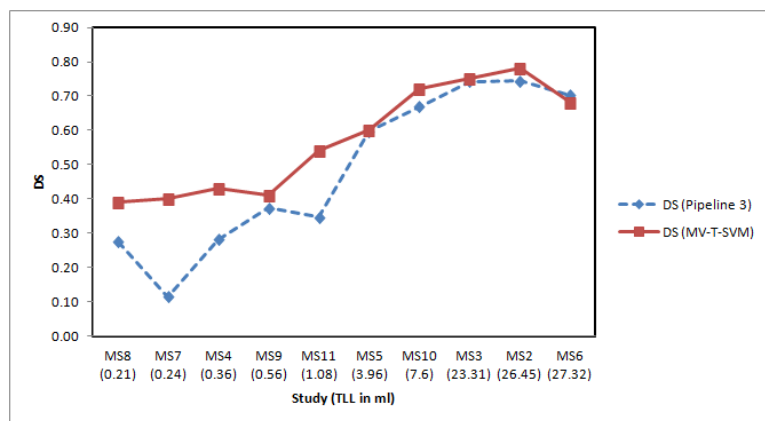
Figure 5-12: Segmentation results of UM datasets using customized MV-T-SVM versus pipeline 3.

5.6.2 Caudate Segmentation

In this section, we test the MV-T-SVM in segmentation of a context other than the MS lesions. The datasets publicly available on CAUSE07 were used to test the performance of using MV-T-SVM to segment the caudate nucleus. CAUSE07 is a competition that was held as part of the workshop 3D Segmentation in the Clinic. The goal of this competition was to compare different algorithms to segment the caudate nucleus from brain MRI scans. The ideal segmentation of caudate is shown in Figure 5-13.

Figure 5-13: Caudate segmentation.

Dataset

The caudate data consisted of a total of 15 training datasets from healthy controls and subjects in a Schizoptypal Personality Disorder (SPD) study provided by Psychiatry Neuroimaging Laboratory at the Brigham and Women's Hospital, Boston. The datasets are referred in results with notation (BWH_PNL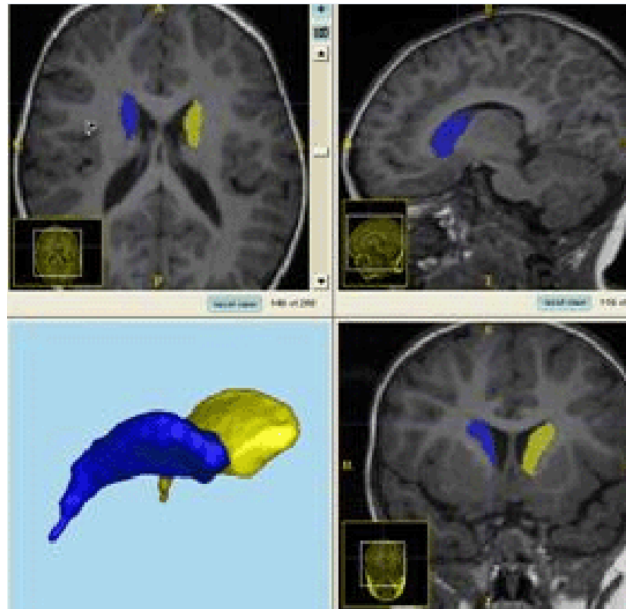_XX_MRI_RAI) with XX is replaced by two digits integer from 01 to 15. Caudate manual segmentation protocols often differ in regard to the exact separation of caudate to the nucleus accumbens anteriorly and the level of inclusion of the vanishing tail posteriorly. Such variability in caudate definition is common place and part of clinical routine when comparing studies.

All datasets were scanned with an Inversion Recovery Prepped Spoiled Grass sequence on a variety of scanners (GE, Siemens, Philips, all 1.5 Tesla). Some data sets were acquired in axial direction, others in coronal direction. All data sets were re-oriented to axial orientation, but were not aligned in any fashion. The resolution of the datasets is at $0.9375 \times 0.9375 \times 1.5$ mm.

Multi-sectional Views Segmentation

In order to apply MV-T-SVM to segment the caudate, the following settings are set to match the input dataset properties:

- Intensity correction is enabled

- Registration is disabled

- Textural Features : Single Channel

- Block size is set 4mm x 4mm

- Atlas prior features and Atlas Tissue based posteriors are disabled

- Contouring algorithm is enabled

As shown in Figure 5-14, the pair of caudate may be visualized and hence segmented from a single axial or coronal sectional view where they are separated in sagittal view. In our testing, the segmentation is done through separating the right and left half of the brains and then result segmentations are added up to result in the final segmentation.

Results on the training datasets

In training, the BWH_PNL_01_MRI_RAI and BWH_PNL_02_MRI_RAI were used in training while all of the 15 datasets were used in segmentation. The result of segmentation is given in Table 5-1. The segmentation performance is evaluated using the dice similarity metric for the left and right nucleus separately and the total segmentation dice similarity is given. The average dice similarity values for left and right halves segmentation are 0.67 and 0.68 which is a good result for using segmentation framework mainly designed for MS segmentation for segmenting other parts in brain. Better results can be achieved if multi-channels images are provided and if features are customized for caudate segmentation.
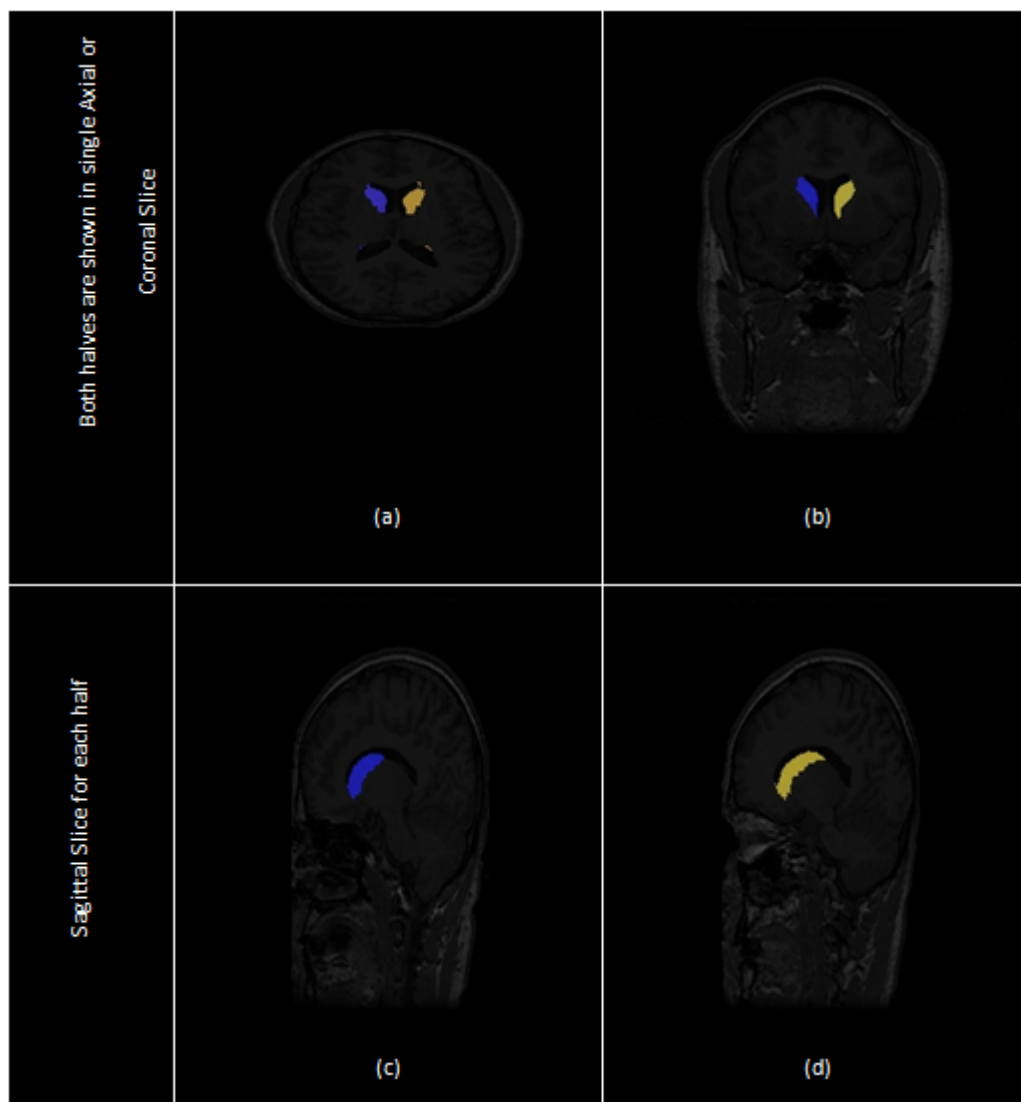
Figure 5-14: Caudate analysis using multi-views segmentation.

Study	Dice Similarity		
	Left	Right	Total
BWH_PNL_01_MRI_RAI	0.84	0.82	0.83
BWH_PNL_02_MRI_RAI	0.8	0.79	0.795
BWH_PNL_03_MRI_RAI	0.71	0.73	0.72
BWH_PNL_04_MRI_RAI	0.63	0.67	0.65
BWH_PNL_05_MRI_RAI	0.65	0.62	0.635
BWH_PNL_06_MRI_RAI	0.45	0.51	0.48
BWH_PNL_07_MRI_RAI	0.62	0.63	0.625
BWH_PNL_08_MRI_RAI	0.69	0.71	0.7
BWH_PNL_09_MRI_RAI	0.74	0.73	0.735
BWH_PNL_10_MRI_RAI	0.65	0.68	0.665
BWH_PNL_11_MRI_RAI	0.32	0.27	0.295
BWH_PNL_12_MRI_RAI	0.83	0.86	0.845
BWH_PNL_13_MRI_RAI	0.73	0.78	0.755
BWH_PNL_14_MRI_RAI	0.64	0.63	0.635
BWH_PNL_15_MRI_RAI	0.7	0.71	0.705
Average	0.67	0.68	0.67

Table 5-1 : Segmentation Results – Segmenting Caudate from training subjects of CAUSE07.

Chapter 6
Conclusion

6.1 Summary

In this dissertation, a set of segmentation frameworks is developed for the purpose of automated classification of multiple sclerosis (MS) lesions in MRI. Specifically, three pipelines are progressively developed to use the statistical texture as a feature vector and the support vector machines (SVM) as a learning technique in the MS lesions segmentation. We proposed two generic configurable components: a processing core module that locates the areas in the brain that potentially forms MS lesions and a post-processing module that adds or removes from these areas to have more accurate segmentations. Based on these configurable modules, single view segmentation and multi-sectional views segmentation pipelines are proposed. The single view segmentation pipeline has two versions: FLAIR channel (pipeline 1) and multi-channels MRI (FLAIR / T1 / T2 / PD) inputs (pipeline 2). Multi-Sectional views segmentation pipeline takes multi-channels MRI (FLAIR / T1 / T2) inputs (pipeline 3). The single views pipelines were evaluated on 10 real datasets. The multi-view pipeline was evaluated by 3 simulated datasets and 61 real datasets. The segmentation average results were both good and competitive.

The multi-sectional views pipeline (pipeline 3) is customized to solve limitations found in the segmentation results and to make best use of the properties and nature of MS lesion in MRI. These customizations and enhancements lead to development of the

155

customized MV-T-SVM. In the customized MV-T-SVM involves the use of the anatomical tissue provided by the MNI atlas in segmentation and the use of the proposed textural features extracted from multi-channel images instead of single channel textural features. Besides, the learning of the SVM of the different views is updated to be overlapping learning scheme instead of non-overlapping learning. In the customized MV-T-SVM, the aggregate function is rule based to differentiate between voxels classification based on the brain anatomical region and the estimated total lesion load of the subject dataset. A contouring algorithm is proposed and incorporated in MV-T-SVM to improve the smoothness of lesion regions and to provide better performance quantitatively. The customized MV-T-SVM was trained with 20 real datasets and tested with another 10 real datasets. The proposed customization applied in this approach results in improvement of the average segmentation performance.

The generality of the proposed segmentation framework is tested by applying the technique with different settings on healthy datasets to segment the caudate. Good results were obtained with minor changes in the segmentation technique.

6.2 Findings:

According to the obtained results for the three pipelines, we concluded a set of findings. The proposed technique would be viable for use in clinical practice for the detection of MS lesions in MRI. It provides good segmentation when tested with datasets acquired at a source different from the source of training datasets with in terms of the acquisition conditions. The textural based SVM (C_3 and C_4) provides good coarse segmentation that can be used as an initial step in any MS segmentation framework. Slice division into

blocks enables the technique to benefit from the textural features (C_3) in detection of the candidate lesion region without manually labeled ROIs. In the same time, slice division into blocks makes the technique avoid voxel based learning which results in high rate of false positives and scattered automatic MS lesions. Using overlapping blocks provide more accurate segmentation with cost of computation time. It is only suitable for low-resolution datasets. Basing the classifier on SVM (C_4) as a machine learning technique provides a robust classifier. Feature selection (C_3) based on explicit use of human visible features and trying to emulate expert non-intended features aid the technique to find lesion areas that has high similarity (Dice similarity / Similarity Index) with the manually labeled areas. Training the blocks classifier in such a way where all blocks that contains at least 1 MS voxel are marked as MS-blocks aids that classifier to find that blocks that are completely or partially are MS blocks. Using more sequences in MRI (C_1) improves the performance of MS Segmentation. Using Tissues Probabilities based on the MNI atlas (C_2 and C_3) improves the performance of MS Segmentation. The use of post-processing module (C_5) that deals with higher level of lesion which the "MS Lesion Region" provides more *Clinic-friendly* outputs as long the improvement in the numeric metrics. The proposed post-processing module (C_5) is generic enough to be added to any other segmentation framework to improve the quality of the segmentation output. Performance evaluation (C_6) that adds metrics such as Dice Similarity for regions (DS_R) and Detected Lesion Load (DLL) ensures the clinically relevant performance. The multi-view segmentation pipeline (C_7) is suitable in cases of high resolution 3D Images that provides Lesion Visibility from all sectional views and it adds to the accuracy of the final segmentation.

According to the obtained results for the customized MV-T-SVM, we concluded a set of findings. Using rule based aggregate function allows improving the performance when tested on datasets with low TLL without effect on datasets with high TLL. Also, the rule based aggregate function reduces false positives in areas in anatomical regions statistically have low likelihood to have MS lesions. Lesion regions shape improvement using the proposed contouring algorithm (C_5) provides more *Clinic-friendly* outputs as long the improvement in the numeric metrics even better than using the post-processing modules used in pipeline 1 and pipeline 2. Using the proposed textural features (C_3) extracted from the multi-channels MRI provided better block segmentation. The overlapping learning (C_4) of the three views classification engines improves the quality of the aggregate segmentation in terms of voxel segmentation.

6.3 Future Work

More efforts are still needed to improve the proposed segmentation framework both in performance and in functionality. In the following, we mention the work that can be applied on the segmentation framework to provide more clinical relevant technique:

1. Textural based segmentation of 3D blocks instead of 2D blocks in multi-views slices should be analyzed.

2. Analysis of using advanced texture models such as Marcov Random Fields texture models [80] to represent the texture and study of the effect on improvement the discrimination between healthy and MS lesion areas.

3. Intensity correction based on the anatomical region of the brain should be analyzed.

4. Analysis of using the Diffusion Tensor Imaging (DTI) for MS segmentation and study the effect on the segmentation performance.

5. In order to provide better clinical relevant output, the monitoring of the MS lesion volume, location and gray level over the time need to be incorporated with the segmentation framework.

6. Study the ability of the segmentation technique to segment other lesions or tissues in MRI or Computed Tomography (CT) scans. Specifically using MV-T-SVM for liver tumor segmentation in MRI and vessels segmentation in lungs is included in our current research activity.

Bibliography

[1] BrainWeb: Simulated Brain Database. [Online]. http://www.bic.mni.mcgill.ca/brainweb/

[2] M Styner, J Lee, B Chin, M Chin, O Commowick, H Tran, S Markovic-Plese, and V Jewells, "3D Segmentation in the Clinic: A Grand Challenge II: MS lesion segmentation," *MIDAS Journal*, pp. 1–5, Sep. 2008.

[3] The Multiple Sclerosis International Federation (MSIF). [Online]. http://www.msif.org.

[4] C H Polman, S C Reingold, G Edan, M Filippi, H P Hartung, F D Lublin, L M Metz, H F McFarland, P W O'Connor, M Sandberg-Wollheim, A J Thompson, B G Weinshenker, and J S Wolinsky, "Diagnostic Criteria for Multiple Sclerosis: 2005 Revisions to the 'McDonald Criteria'," *Annals of Neurology*, vol. 58, pp. 840-856, 2005.

[5] D H Miller, R I Grossman, S C Reingold, and F McFarlan, "The Role of Magnetic Resonance Techniques in Understanding and Managing Multiple Sclerosis," *Brain*, vol. 121, pp. 3–24, 1998.

[6] Daisuke Yamamoto, Hidetaka Arimura, Shingo Kakeda, Taiki Magome, Yasuo Yamashita, Fukai Toyofuku, Masafumi Ohki, Yoshiharu Higashida, and Yukunori Korogi, "Computer-Aided Detection of Multiple Sclerosis Lesions in Brain Magnetic Resonance Images: False Positive Reduction Scheme Consisted of Rule-Based, Level Set Method, and Support Vector Machine," *Computerized Medical Imaging and Graphics*, vol. 34, no. 5, pp. 404-413, July 2010, DOI: 10.1016/j.compmedimag.2010.02.001.

[7] Leo Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001.

[8] C J Wallace, T P Seland, and T C Fong, "Multiple Sclerosis: The Impact of MR Imaging," *American Journal of Roentgenol*, vol. 158, pp. 849–857, 1992.

[9] S Wiebe, D H Lee, S J Karlik, M Hopkins, M K Vandervoort, C J Wong, L Hewitt, G PA Rice, G C Ebers G C, and J H Noseworthy, "Serial Cranial and Spinal Cord Magnetic Resonance Imaging in Multiple Sclerosis," *Annals of Neurology*, vol. 32, pp. 643–650, 1992.

[10] L Truyen, "Magnetic Resonance Imaging in Multiple Sclerosis: A Review," *Acta Neurologica Belgica*, vol. 94, pp. 98–102, 1994.

[11] F Fazekas, F Barkhof, M Filippi, R I Grossman, D KB Li, W I McDonald, H F McFarland, D W Paty, J H Simon, J S Wolinsky, and D H Miller, "The Contribution of Magnetic Resonance Imaging to the Diagnosis of Multiple Sclerosis," *Neurology*, vol. 53, pp. 448–456, 1999.

[12] V N Vapnik, *The Nature of Statistical Learing Theory*. New York: Springer-Verlag, 1995.

[13] Ezequiel Geremia, Bjoern Menze, Olivier Clatz, Ender Konukoglu, Antonio Criminisi, and Nicholas Ayache, "Spatial Decision Forests for MS Lesion Segmentation in Multi-Channel MR Images," in *Medical Image Computing and Computer-Assisted Intervention â MICCAI 2010*, vol. 6361, 2010, pp. 111-118.

[14] Ayelet Akselrod-Ballin, Meirav Galun, John Moshe Gomori, Massimo Filippi, Paola Valsasina, Ronen Basri, and Achi Brandt, "Automatic Segmentation and Classification of Multiple Sclerosis in Multichannel MRI," *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 10, Oct. 2009.

[15] Daniel Garcia-Lorenzo, Sylvain Prima, Sean P Morrissey, and Christian Barillot, "Multiple Sclerosis Lesion Segmentation using an Automatic Multimodal Graph Cuts," in *12th International Conference on Medical Image Computing and Computer Assisted Intervention*, London, 2009, pp. 584-591, DOI : 10.1007/978-3-642-04271-3.

[16] J Lecoeur, S P Morissey, J C Ferré, D Arnold, D Collins, and C Barillot, "Optimized Supervised Segmentation of MS Lesions from Multispectral MRIs," in *Proceedings of MICCAI workshop on Medical Image Analysis on Multiple Sclerosis (validation and methodological issues)*, 2009.

[17] J Lecoeur, S P Morissey, J C Ferré, D Arnold, D Collins, and C Barillot, "Multiple Sclerosis Lesions Segmentation using Spectral Gradient and Graph Cuts," in *Proceedings of MICCAI workshop on Medical Image Analysis on Multiple Sclerosis (validation and methodological issues)*, 2008.

[18] Erik Pernod, Jean-Christophe Souplet, Javier Rojas Balderrama, Diane Lingrand, and Xavier Pennec, "Multiple Sclerosis Brain MRI Segmentation Workflow deployment on the EGEE Grid," in *Olabarriaga, Lingrand and Montagnat (Eds.)*, New York, 9. 2008.

[19] Daniel Garcia-Lorenzo, Sylvain Prima, Sean P Morrissey, and Christian Barillot, "A Robust Expectation-Maximization Algorithm for Multiple Sclerosis Lesion Segmentation," in *The MIDAS Journal - MS Lesion Segmentation (MICCAI Workshop)*, 2008.

[20] P Anbeek, K L Vincken, and M A Viergever, "Automated MS-Lesion Segmentation by K-Nearest Neighbor Classification," *MIDAS Journal*, 2008.

[21] Ying Wu, Simon K Warfield, Leng Tan, William M Wells, Dominik S Meier, Ronald A van Schijndel, Frederik Barkhof, and Charles R G Guttmanna, "Automated Segmentation of Multiple Sclerosis Lesion Subtypes with Multichannel MRI," *NeuroImage*, vol. 32, pp. 1205 – 1215, 2006.

[22] Ruben Cardenes, Simon K Warfield, Elsa M Macıas, Jose Aurelio Santana, and Juan Ruiz-Alzola, "An Efficient Algorithm for Multiple Sclerosis Lesion Segmentation from Brain MRI," in *EUROCAST 2003, LNCS 2809*, Berlin Heidelberg, 2003, pp. 542–551.

[23] Ricardo J Ferrari, Xingchang Weia, Yunyan Zhanga, James N Scottb, and J Ross Mitchell, "Segmentation of Multiple Sclerosis Lesions Using Support Vector," in *SPIE 5032*, vol. 16, 2003, doi:10.1117/12.481377.

[24] Dirk-Jan Kroon, Erik van Oort, and Kees Slump, "Multiple Sclerosis Detection in Multispectral Magnetic Resonance Images with Principal Components Analysis," *The MIDAS Journal - MS Lesion Segmentation (MICCAI Workshop)*, 2008.

[25] K V Leemput, F Maes, D Vandermeulen, A Colchester, and P Suetens, "Automated Segmentation of Multiple Sclerosis Lesions by Model Outlier Detection," *IEEE Transactions on Medical Imaging*, vol. 20, no. 8, pp. 677–688, 2001.

[26] Marcel Prastawa and Guido Gerig, "Automatic MS Lesion Segmentation by Outlier Detection and Information Theoretic Region Partitioning," *The MIDAS Journal - MS Lesion Segmentation (MICCAI Workshop)*, 2008.

[27] Stephanie Bricq, Christophe Collet, and Jean-Paul Armspa, "MS Lesion Segmentation based on Hidden Markov Chains," *The MIDAS Journal - MS Lesion Segmentation (MICCAI Workshop)*, 2008.

[28] Stephanie Bricq, Christophe Collet, and Jean-Paul Armspa, "Lesions Detection on 3D Brain MRI using Trimmmed Likelihood Estimator and Probabilistic Atlas," in *Biomedical Imaging: From Nano to Macro. ISBI. 5th IEEE International Symposium*, Paris, 2008, pp. 93 - 96, DOI:10.1109/ISBI.2008.4540940.

[29] A Oliva and A Torralba, "The Role of Context in Object Recognition," *Trends in Cognitive Sciences*, vol. 11, no. 12, pp. 520–527, Dec. 2007.

[30] Jonathan H Morra, Zhuowen Tu, Arthur W Toga, and Paul M Thompson, "Automatic Segmentation of MS Lesions Using a Contextual Model," *The MIDAS Journal - MS Lesion Segmentation (MICCAI Workshop)*, 2008.

[31] Mark Scully, Vincent Magnotta, Charles Gasparovi, Peter Pelligrino, Delia Feis, and H Jeremy Bockholt, "3D Segmentation In The Clinic," *The MIDAS Journal - - MS Lesion Segmentation (MICCAI Workshop)*, 2008 (MSC).

[32] Ying Zhugea and Jayaram K Udupaa, "Multiple Sclerosis Lesion Quantification in MR Images by using Vectorial Scale-based Relative Fuzzy Connectedness," in *Medical Imaging 2004: Image Processing - Proceedings of SPIE*, vol. 5370, 2004, DOI: 10.1117/12.535655.

[33] Chaozhe Zhu and Tinazi Jiang, "Knowledge Guided Information Fusion for Segmentation of Multiple Sclerosis Lesions in MRI Images," in *SPIE03*, vol. 5032, 2003, DOI:10.1117/12.480312.

[34] M Ghazel, A Traboulsee, and R K Ward, "Optimal Filter Design for Multiple Sclerosis Lesions Segmentation from Regions of Interest in Brain MRI," in *IEEE International Symposium on Signal Processing and Information Technology*, 2006, pp. 1-5.

[35] Oren Freifeld, Hayit Greenspan, and Jacob Goldberger, "Multiple Sclerosis Lesion Detection Using Constrained GMM and Curve Evolution," *International Journal of Biomedical Imaging*, 2009, DOI:10.1155/2009/715124.

[36] R Khayati, M Vafadust, F Towhidkhah, and S M Nabavi, "Fully Automatic Segmentation of Multiple Sclerosis Lesions in Brain MR FLAIR Images using Adaptive Mixtures method and Markov Random Field Model," *Computers in Biology and Medecine*, vol. 38, pp. 379–390, 2008.

[37] David D Stark and William G Bradley, *Magnetic Resonance Imaging*, 3rd ed.: Mosby, Incorporated, 1999.

[38] B Johnston, M S Atkins, B Mackiewich, and M Anderson, "Segmentation of Multiple Sclerosis Lesions in Intensity Corrected Multispectral MRI," *IEEE Transactions on Medical Engineering*, vol. 15, pp. 154–169, 1996.

[39] B Mackiewich, *Master's thesis:Intracranial Boundary Detection and Radio Frequency Correction in Magnetic Resonance Images*. Burnaby, British Columbia: Simon Fraser Univ. Computer Science Dept., 1995.

[40] S M Smith, "Fast Robust Automated Brain Extraction," *Hum Brain Mapp*, vol. 17, no. 3, pp. 143-155, 2002.

[41] N K Subanna, M Shah, S J Francis, S Narayanan, D L Collins, D L Arnold, and T Arbel, "MS Lesion Segmentation using Markov Random," in *Workshop on Medical Image Analysis on Multiple Sclerosis - MIMAS*, 2009.

[42] Jing Zhang, Longzheng Tong, Lei Wang, and Ning Li, "Texture Analysis of Multiple Sclerosis: A Comparative Study," *Magnetic Resonance Imaging*, vol. 26, no. 8, pp. 1160-1166, Oct. 2008, DOI: 10.1016/j.mri.2008.01.016.

[43] J H Friedman, F Baskett, and L J Shustek, "An Algorithm for Finding Nearest Neighbors," *IEEE Transactions on Computers*, vol. C-24, pp. 1000–1006, 1975.

[44] S K Warfield, "Fast k-NN Classification for Multichannel Image Data," *Pattern Recognition Letters*, pp. 713–721, 1996.

[45] C M Bishop, *Neural Networks for Pattern Recognition*. Oxford, Great Britain: Oxford University Press, 1995.

[46] R O Duda, P E Hart, and D G Stork, *Pattern Classification*. New York, U.S.A.: John Wiley & Sons, Inc., 2001.

[47] L P Clarke, R P Velthuizen, S Phuphanich, J D Schellenber, J A Arrington, and M Silbiger, "MRI: Stability of Three Supervised Segmentation Techniques," *Magnetic Resonance Imaging*, vol. 11, pp. 95–106.

[48] S Vinitski, C Gonzalez, F B Mohamed, T Iwanaga, K Kamil Khalili, and J Mack, "Improved Intracranial Lesion Characterization by Tissue Segmentation based on a 3D Feature Map," *Magnetic Resonance In Medicine*, vol. 37, pp. 457–469, 1997.

[49] Yale Amit and Donald Y Geman, "Shape Quantization and Recognition with Randomized Trees," *Neural Computation*, vol. 9, no. 7, pp. 1545–1588, 1997.

[50] Jeremy Lecoeur, Jean-Christophe Ferre, and Christian Barillot, "Optimized Supervised Segmentation of MS Lesions from Multispectral MRIs," 2009, p. MICCAI workshop on Medical Image Analysis on Multiple Sclerosis (validation and methodological issues).

[51] C Cortes and V Vapnik, "Support Vector Networks," *Machine Learning*, vol. 20, pp. 1-25, 1995.

[52] CJC Burges, *A Tutorial on Support Vector Machines for Pattern Recognition*. Boston: Kluwer Academic Publishers, 1998, vol. 2.

[53] Douglas M Hawkins, *Identification of Outliers*.: Chapman and Hall, 1980.

[54] P Schroeter, J -M Vesin, T Langenberger, and R Meuli, "Robust Parameter Estimation of Intensity Distributions for Brain Magnetic Resonance Images," *IEEE Transactions on Medical Engineering*, vol. 17, no. 2, pp. 172–186, Apr. 1998.

[55] D Garcia-Lorenzo, L Lecoeur, D Arnold, and D L Collins, "Multiple Sclerosis Lesion Segmentation using an Automatic Multimodal Graph Cuts," in *MICCAI09*, 2009, pp. 584-591.

[56] Sushmita Datta, Balasrinivasa Rao Sajja, Renjie He, Rakesh K Gupta, Jerry S Wolinsky, and Ponnada A Narayana, "Segmentation of Gadolinium-Enhanced Lesions on MRI in Multiple Sclerosis," *Magnetic Resonance Imaging*, vol. 25, pp. 932–937, 2007.

[57] Frithjof Kruggel, Suresh Paul Joseph, and Hermann-Josef Gertz, "Texture-based Segmentation of Diffuse Lesions of the Brain's White Matter," *NeuroImage*, vol. 39, no. 3, pp. 987-996, Feb. 2008, DOI: 10.1016/j.neuroimage.2007.09.058.

[58] R K.-S Kwan, A C Evans, and G B Pike, "An Extensible MRI Simulator for Post-Processing Evaluation," *Visualization in Biomedical Computing (VBC'96). Lecture Notes in Computer Science*, vol. 1131, pp. 135-140, 1996.

[59] D L Collins, A P Zijdenbos, V V. Kollokian, J G Sled, N J Kabani, C J Holmes, and A C Evans, "Design and Construction of a Realistic Digital Brain Phantom," *IEEE Transactions on Medical Imaging*, vol. 17, no. 3, pp. 463-468, June 1998.

[60] K V Leemput, *PhD Thesis: Quantitative Analysis of Signal Abnormalities in MR Imaging for Multiple Sclerosis and Creutzfeldt-Jakob Disease*. Leuven, Belgium: Katholieke Universiteit Leuven, 2001.

[61] L J Rosner and S Ross, *Multiple Sclerosis*. New York: Simon and Schuster, 1992.

[62] C A Cocosco, V Kollokian, R K.-S Kwan, and A C Eva, "BrainWeb: Online Interface to a 3D MRI Simulated Brain Database," in *NeuroImage, Proceedings of 3-rd International Conference on Functional Mapping of the Human Brain*, Copenhagen, 1997.

[63] R K.-S Kwan, A C Evans, and G B Pike, "MRI Simulation-based Evaluation of Image-Processing and Classification Methods," *IEEE Transactions on Medical Imaging*, vol. 18, no. 11, pp. 1085-1097, Nov. 1999.

[64] MS Lesion Segmentation Challenge 2008. [Online]. http://www.ia.unc.edu/MSseg/

[65] Akmal Younis, Mohamed Ibrahim, Mansur Kabuka, and Nigel John, "An Artificial Immune-Activated Neural Network Applied to Brain 3D MRI Segmentation," *Journal of Digital Imaging*, 2008.

[66] Jan Rexilius, Horst K Hahn, Holger Bourquain, and Heinz-Otto Peitgen, "Ground Truth in MS Lesion Volumetry −A Phantom Study," *Medical Image Computing and Computer-Assisted Intervention - MICCAI03 Lecture Notes in Computer Science*, vol. 2879, pp. 546-553, 2003, DOI: 10.1007/978-3-540-39903-2_67.

[67] Chih-Chung Chang and Chih-Jen Lin, "LIBSVM: A Library for Support Vector Machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1-27, 2011, Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[68] D Goldberg-Zimring, H Azhari, S Miron, and A Achiron, "3-D Surface Reconstruction of Multiple Sclerosis Lesions using Spherical Harmonics," *Magnetic Resonance Imaging*, vol. 46, pp. 756–766, 2001.

[69] Zheru Chi, Hong Yan, and Turan Pham, *Fuzzy Algorithms: With Application to Image Processing and Pattern Recognition*, 1st ed. Singapore: World Scientific Publishing Co. Pte. Ltd, 1996.

[70] A C Evans, D L Collins, S R Mills, E D Brown, R L Kelly, and T M Peters, "3D Statistical Neuroanatomical Models from 305 MRI Volumes," in *IEEE-Nuclear Science Symposium and Medical Imaging Conference*, 1993, pp. 1813–1817.

[71] R P Woods, S T Grafton, C J Holmes, S R Cherry, and J C Mazziotta, "Automated Image Registration: General Methods and Intrasubject, Intramodality Validation," *Journal of Computer Assisted Tomography*, vol. 22, pp. 39-152, 1998.

[72] A O Boudraa, S MR Dehakb, Y M Zhu, C Pachai, Y G Bao, and J Grimaud, "Automated Segmentation of Multiple Sclerosis Lesions in Multispectral MR Imaging using Fuzzy Clustering," *Computers in Biology and Medecine*, 2000.

[73] A P Zijdenbos, R Forghani, and A C Evans, "Automatic "pipeline" Analysis of 3-D MRI Data for Clinical Trials: Application to Multiple Sclerosis," *IEEE Transactions on Medical Imaging*, pp. 1280–1291, 2002.

[74] Bassem A. Abdullah, Akmal A Younis, Pradip M Pattany, and Efrat Saraf-Lavi, "Textural based SVM for MS Lesion Segmentation in FLAIR MRIs," *Open Journal of Medical Imaging*, vol. 1, no. 2, Dec. 2011.

[75] Akmal A Younis, Bassem A. Abdullah, Pradip M Pattany, and Efrat Saraf-Lavi, "Multiple Sclerosis Lesion Segmentation in Multi-Channel MRIs using Textural-based SVMs," *Journal of the Digital Imaging*, 2011, In Review.

[76] Bassem A. Abdullah, Akmal A Younis, and Nigel John, "Multi-Sectional Views Textural based SVM for MS Lesion Segmentation in Multi-Channels MRIs," *The Open Biomedical Engineering Journal*, vol. 6, no. 1, pp. 5-23, Dec. 2011.

[77] J C Souplet, C Lebrun, N Ayache, and G Malandain, "An automatic Segmentation of T2-FLAIR Multiple Sclerosis Lesions," *The MIDAS Journal - - MS Lesion Segmentation (MICCAI Workshop)*, 2008.

[78] Results Page - MS Lesion Segmentation Challenge 2008 (Team Name: UM-ECE_team). [Online]. http://www.ia.unc.edu/MSseg/results_table.php

[79] CAUSE07:Caudate Segmentation Evaluation 2007. [Online]. http://www.cause07.org/

[80] George R Cross and Anil K Jain, "Markov Random Field Texture Models," *IEEE Transactions Pattern Analysis and Machine Intelligence*, vol. 5, no. 1, pp. 25 - 39, Jan. 1983.