

Textural Based SVM for MS Lesion Segmentation in FLAIR MRIs

Bassem A. Abdullah¹, Akmal A. Younis¹, Pradip M. Pattany², Efrat Saraf-Lavi²

¹Department of Electrical and Computer Engineering, University of Miami, Miami, USA

²Department of Radiology, Miller School of Medicine, University of Miami, Miami, USA

E-mail: b.abdullah@umiami.edu

Received October 17, 2011; revised November 8, 2011; accepted December 3, 2011

Abstract

In this paper, a new technique is proposed for automatic segmentation of multiple sclerosis (MS) lesions from brain magnetic resonance imaging (MRI). The technique uses textural features to describe the blocks of each MRI slice along with position and neighborhood features. A trained support vector machine (SVM) is used to discriminate between the blocks in regions of MS lesions and the blocks in non-MS lesion regions based on mainly the textural features with aid of the other features. The MRI slice blocks' classification is used to provide an initial segmentation. A comprehensive post processing module is then utilized to refine and improve the quality of the initial segmentation. The main contribution of the proposed technique described in this paper is the use of textural features to detect MS lesions in a fully automated process without the need to manually define regions of interest (ROIs). In addition, the post processing module is generic enough to be applied to the results of any other MS segmentation technique to improve the segmentation quality. This technique is evaluated using ten real MRI data-sets with 10% used in the training of the textural-based SVM. The average results for the performance evaluation of the presented technique were 0.79 for dice similarity, 0.68 for sensitivity and 0.9 for the percentage of the detected lesion load. These results indicate that the proposed method would be useful in clinical practice for the detection of MS lesions from MRI.

Keywords: MRI, Texture Analysis, Segmentation, Multiple Sclerosis, SVM, ROI, Post-Processing, Fuzzy Rules

1. Introduction

Multiple sclerosis (MS) is a chronic idiopathic disease that results in multiple areas of inflammatory demyelization in the central nervous system (CNS) [1]. 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 [2,3]. 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 [4]. Automatic Segmentation offers an attractive alternative to manual segmentation which remains a time-consuming task and suffers from intra- and inter-expert variability [5]. 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 [6-9]. This makes the automatic segmentation of MS lesions a challenging problem.

Because of the importance of computer-aided MS lesion detection, a number of semi-automated and automated methods have been proposed for segmenting MS lesions in MR images [4,5,10-16]. 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. Textural features have been used [17] to differentiate between lesion white matter (LWM), normal white matter (NWM) and normal appearing white matter (NAWM). Texture classification was also used for the analysis of multiple sclerosis [1]. The use of textural features is promising to provide good results in MS lesion segmentation, especially if the Fluid Attenuated Inversion Re-

covery (FLAIR) sequence is used where the textural attributes of the MS lesions are different from other regions in the brain.

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

In this paper, we propose a technique that uses textural features to describe the blocks of each MRI slice along with position and neighborhood features. A trained classifier is used to discriminate between the blocks and detect the blocks that potentially include MS lesions mainly based on the textural features with aid of the other features. The blocks classification is used to provide an initial coarse segmentation of the MRI slices. The textural-based classifier is built using Support Vector Machine (SVM), one of the widely used supervised learning algorithms that have been utilized successfully in many applications [18,19]. Classification errors can occur in this initial coarse segmentation. False positives can arise because of the similarity in textural attributes between some healthy regions in the brain and the MS lesions used in training the textural-based classifier. On contrary, false negatives can arise from any source of noise in the MR images that may corrupt the textural attributes of the MS lesions. To overcome both types of errors, a comprehensive post processing module is added to improve the quality of the initial segmentation by addressing each type of error individually to generate the final MS lesion segmentation. All the sequences of MRI are used in the task of MS segmentation. FLAIR images especially axial slices are selected in our system. Due to the higher accuracy of the FLAIR imaging sequence in revealing MS lesions and assessing the total lesion load [20,21], axial FLAIR MRI were used in this paper.

The paper consists of five sections including this introduction section. In Section 2, the details of the proposed segmentation technique are illustrated. The experimental results are presented in Section 3 and discussed in Section 4. The paper conclusion is stated in Section 5. For completeness, Appendix A provides the details for calculating the textural features.

2. Materials and Methods

2.1. Dataset

The dataset used in this paper involves FLAIR MRI sequences for ten subjects (4 males, age range: 50 - 72 and

6 females, age range: 30 - 59). On average, each FLAIR MRI sequence consists of thirty seven slices that cover the whole brain. All the subjects were referred for brain MRI studies based on an earlier diagnosis of MS. The MS lesions were manually labeled by a neuroradiologist. The ten MRI studies were acquired using a 3.0T MR scanner under a human subject's protocol approved by the institutional review board of the University Of Miami Miller School Of Medicine (Florida, USA). The axial FLAIR sequences used in this paper were acquired using the following imaging parameters: 9000/103/2500/256 × 204/17/123 (repetition time ms/echo time ms/inversion time/matrix size/echo time length/imaging frequency). In clinical practice, the T2-weighted and/or FLAIR images have been an established routine sequence for diagnosis of MS lesions [22], and thus FLAIR MR imaging was employed in this paper. All images were acquired with a slice thickness of 3 mm, an interslice gap of 0.9 mm, and a field of view of 175 × 220 mm. Pixels are sampled to 16-bits and resolution of 408 × 512 (pixel size 0.43 × 0.43 mm).

2.2. MS Lesions Segmentation Framework

The proposed MS lesions segmentation framework is described in **Figure 1**. 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 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 corrects the MS lesions for each slice by removing voxels based on distance and grayscale features, and adding neighboring voxels using region growing based on grayscale features and adding voxels by removing holes in lesion regions using lesion continuity fact.

2.3. Preprocessing

Due to different operating conditions from a subject to another, brightness and contrast of 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

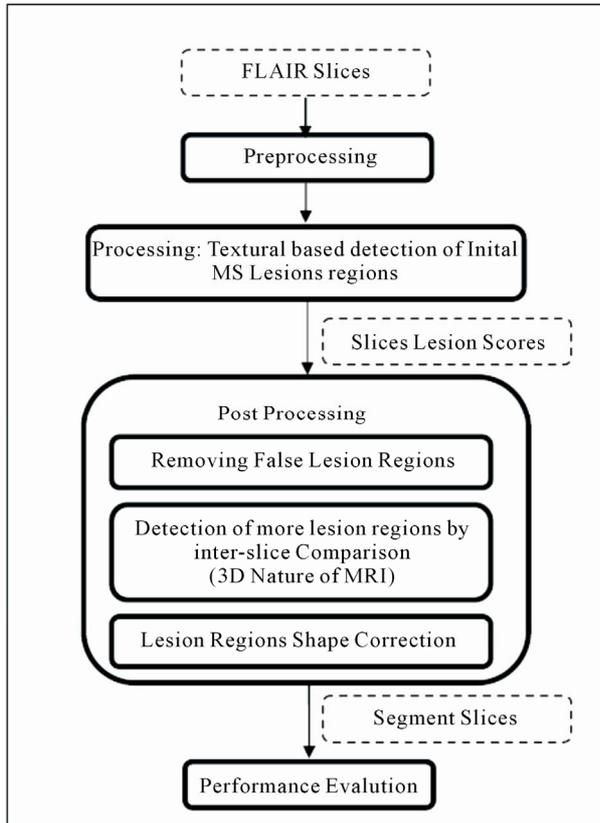


Figure 1. MS lesion segmentation framework.

be segmented and the training set will lead to error reduction.

We used our preprocessing technique used before in [23] 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 avoid empty histogram bins. Figure 2 shows the effect of the preprocessing technique

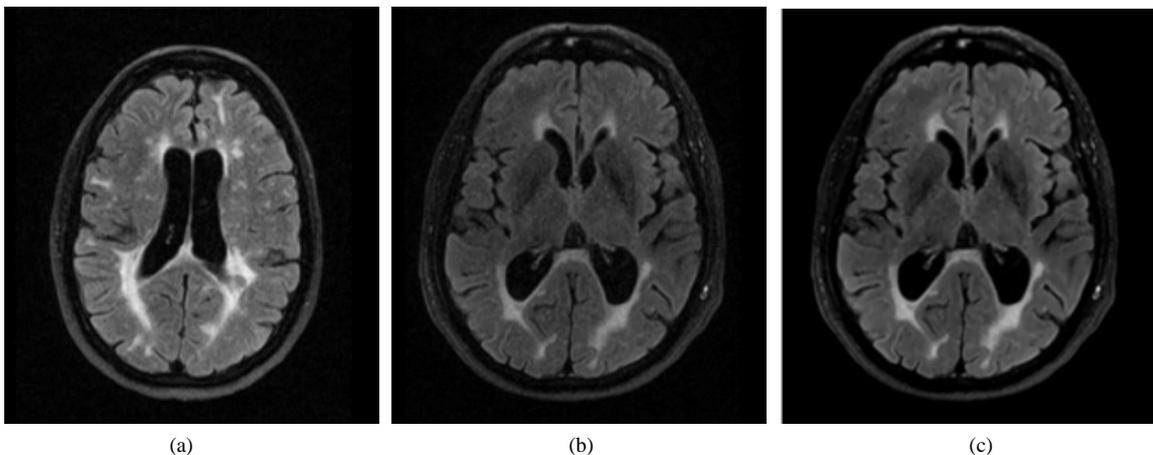


Figure 2. Preprocessing: (a) A slice from the reference subject MS3 (used in training); (b) A slice from subject MS6 before preprocessing; (c) The same slice of MS6 after preprocessing.

on improving a slice from MS6 (subject dataset to be segmented) with reference to MS3 (subject dataset used in training).

In addition, the preprocessing step includes the detection of the center of mass and the sagittal plane (central line passing through the center in the same direction of the slice orientation) for each slice. These geometrical parameters are needed in feature extraction.

2.4. Textural Based Detection of Initial MS Lesions Regions

Each preprocessed MRI slice, is processed by a trained detector to get initial MS lesions regions. The detector engine in our method is implemented using support vector machine. Training the detector machine is done by processing the training dataset and dividing its slices into square blocks and assigning binary class for each block. If the block contains at least one pixel 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.

2.4.1. 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 13.5 mm [24]. For any input MRI studies, the square $w \times w$ blocks are selected automatically to tightly cover the smallest possible diameter as a function of the given pixel sizes calculated from the input resolution and field of view. For the size

of pixels in our studies, square 8×8 (pixel²) blocks (3.4×3.4 mm²) tightly cover the smallest possible MS lesion diameter.

2.4.2. Feature Vector

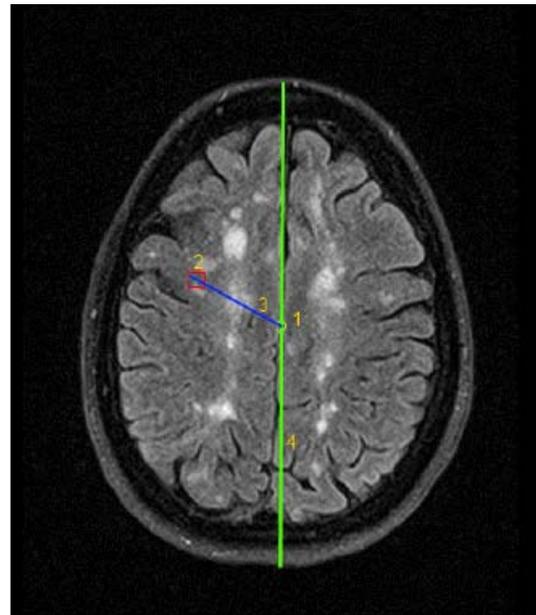
To describe each square block of the FLAIR MRI slice, a feature vector of thirty four features is calculated. The block features are categorized into three categories: twenty four textural features, two position features and eight neighboring blocks features. The features are summarized in **Table 1**. The 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 Appendix A.

The position features are the slice relative location with reference to the bottom slice and the radial Euclidean distance between the block top pixel and the center of the slice normalized by dividing it by the longest diameter of the slice. A sample labeled slice to illustrate the position features is shown in **Figure 3**. In **Figure 3(a)**, the block is labeled by 2, the slice center is labeled by 1, the radial Euclidean distance is labeled by 3 and the longest diameter is labeled by 4. The center and the longest diameter of the slice are parameter geometrically calculated in the preprocessing step.

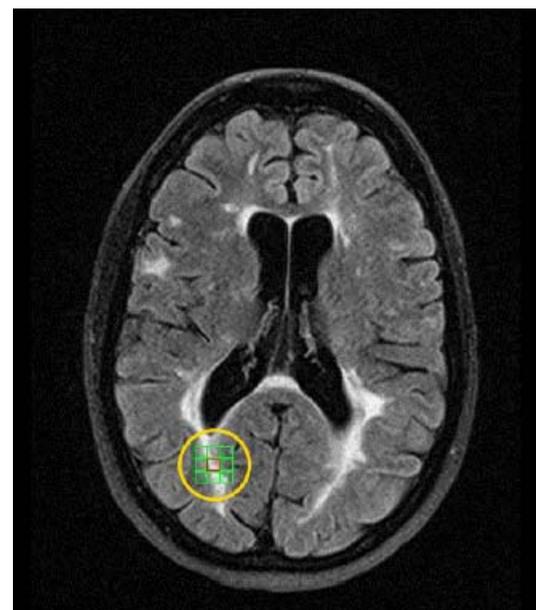
Table 1. Feature vector.

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-uniformity in the 4 directions (Run length based feature)	
	9 - 12 Run length non-uniformity 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 in grayscale between the block and each of the 8 neighboring blocks

The neighboring blocks features are the difference between the mean grayscale of the current block and the mean grayscale of each of the eight neighboring blocks. In **Figure 3(b)**, the 3×3 grid circled by yellow circle



(a)



(b)

Figure 3. Position and neighboring blocks features extraction on sample slices. (a) Calculation of the normalized radial distance between block (2) and slice center (1) (length of line 3/length of line 4); (b) 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).

demonstrates the current block colored with red color and its eight neighboring blocks colored with green color.

2.4.3. 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. 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 [25] include 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.

2.4.3.1. Training

The dataset of one or more subjects is used to generate the SVM training set. The slices of this training dataset are divided into n square blocks of size $w \times w$ pixels. SVM Training set T is composed of training entries $t_i (x_i, y_i)$ where x_i is the feature vector of the block b_i , y_i is the class label of this block for $i=1:n$ (number of blocks included in the training set). Our classification problem is binary, so 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 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 $w \times w$ 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 defined in our method as MS block.

Similarly, the blocks involved in the negative training entries (T_M) are generated by localizing the non-background pixels that are not manually labeled as MS pixels and dividing them into non-overlapping square blocks of size $w \times w$ pixels. 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: w^2 MS pixels. This helps the SVM engine to learn the features of the blocks that either partially or completely contain MS pixels.

In our case, we used one subject dataset (only 10% of the subjects) which consists of thirty seven slices as the training dataset. Since the training set entries are as many as the number of blocks, the training set will be large enough (134173 training entries against 34 features with ratio 3946:1) to avoid the curse of dimensionality, which is the problem that the performances of the pattern classification systems could deteriorate if the ratio of the number of training data to that of features used for the classifier is relatively small [26].

The training set entries were fed to the SVM engine to generate a MS classifier which is able to classify any square $w \times w$ block of a brain FLAIR MRI slice as MS block ($y = 1$) or non-MS block ($y = 0$) based on its feature vector (x).

2.4.3.2. Segmentation

Each of the slices of the datasets to be segmented is divided into overlapping square blocks of size $w \times w$ pixels. The feature vector for each block is calculated. The trained SVM is used to predict the class labels for all the overlapping blocks. The block division is done in an overlapping manner to detect any possible MS 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**. Thus, the score will be any value from 0 to w^2 .

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 5** shows segmentation of sample slice from subject (MS6). **Figure 5(a)** shows the preprocessed FLAIR slice. **Figure 5(b)** is the ground truth for the lesions generated through manual segmentation. **Figure 5(c)** is the initial segmentation by considering any pixel of score higher than zero as MS pixel. **Figure 5(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.

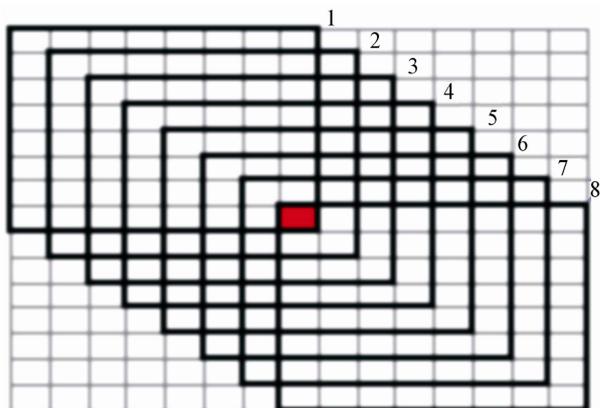


Figure 4. All possible overlapping blocks that contain a pixel: for 8×8 blocks ($w = 8$), the red pixel is part of $w^2 = 64$ blocks. The eight bold blocks are samples where the red pixel lies in the coordinates (8,8) of block 1, (7,7) of block 2 ... and (1,1) of block 8.

2.5. Post Processing

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 6** shows the initial segmentation of a sample slice from subject MS5 (**Figure 6(a)**) and the colored evaluation of the segmentation in which the false negatives and positives are marked in green and red colors, respectively (**Figure 6(b)**). 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 6(b)**).
- Type 2: False negatives resulting from incomplete MS lesion regions (labeled by 2 in **Figure 6(b)**).
- Type 3: False positives resulting from false MS lesion regions (labeled by 3 in **Figure 6(b)**).

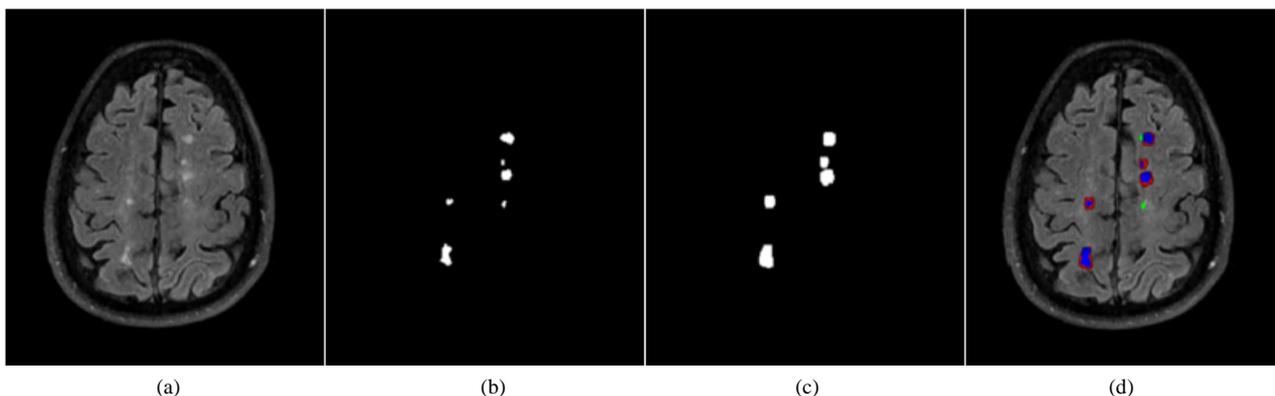


Figure 5. Initial MS lesions regions detection: (a) Preprocessed slice from MS6; (b) Ground truth; (c) Initial segmentation; (d) Colored evaluation of segmentation.

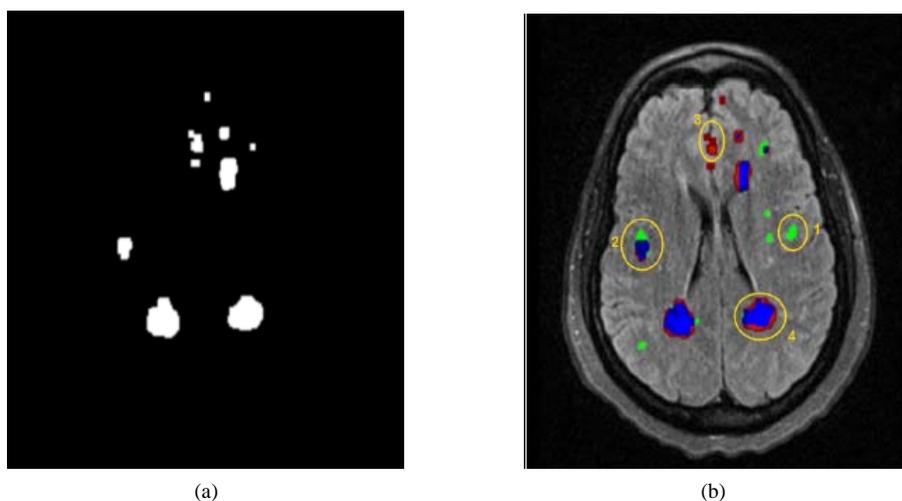


Figure 6. False negatives and positives in the textural segmentation (a) initial segmentation of a slice from MS5 and (b) colored evaluation of the initial segmentation where the different types of errors are labeled by a number matching the corresponding error type.

- Type 4: False positives resulting from false portions of true MS lesion regions (labeled by 4 in **Figure 6(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 7**.

Step1: 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 8**, step 1 of post processing is applied to the initial segmentation of a slice from subject MS6, shown in **Figure 8(a)** and color evaluated in **Figure 8(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 **(Figures 8(c) and (d))**.

essing is shown in **(Figures 8(c) and (d))**.

Step 2: 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 [28], 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

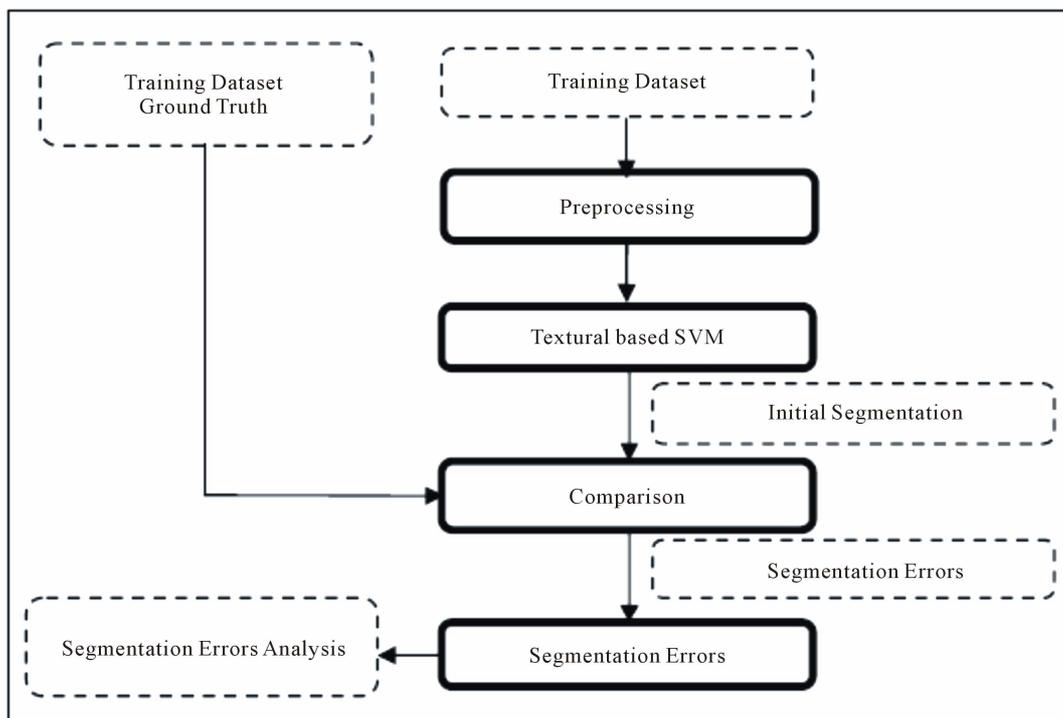


Figure 7. Formulation of criteria and thresholds used in post processing.

new lesion regions are part of non-detected lesions in the current slice, and they should be added as initial segmentations. In **Figure 9**, step 2 of post processing is applied to the initial segmentation of a slice from subject MS6. **Figure 9(a)** shows slice 12 and the circled green lesion is a sample for a completely non-detected MS lesion region. **Figures 9(b)** and **(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 9(d)**, they recover part of the non-detected lesion region in slice 12 as shown in **Figure 9(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.

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 11** will be used to illustrate the application of the different operations in post processing step 3 to a sample segmented slice from MS6. **Figures 11(i1)** and **(i2)** show the initial colored evaluation of the segmentation and the initial segmentation, respectively. The other parts of **Figure 11** will be used to illustrate the corresponding operations in the following discussion of step 3.

(a) Elimination of the false positives on the boundary of detected MS lesion regions.

The colored evaluation of the segmentation depicted in **Figure 11(i1)** shows that each lesion region 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

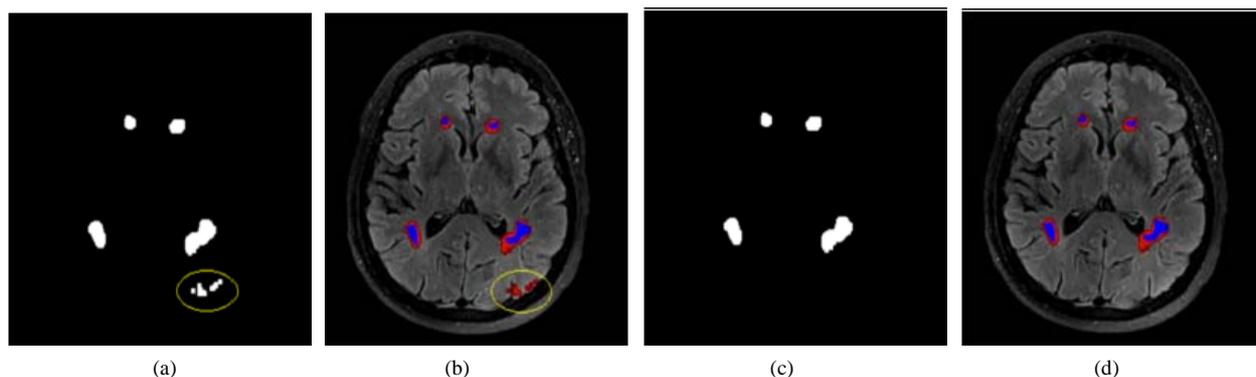


Figure 8. Post processing step 1: elimination of segmented lesion regions in odd locations. (a) and (b) are the slice segmentation and colored evaluation of the segmentation before applying step 1 (MS lesion regions in odd locations are circled in yellow). (c) and (d) are the slice segmentation and colored evaluation of the segmentation after applying step 1.

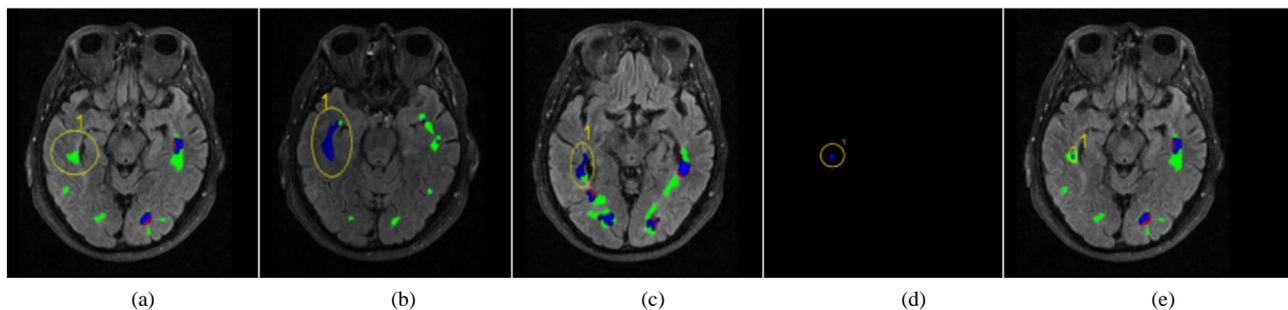


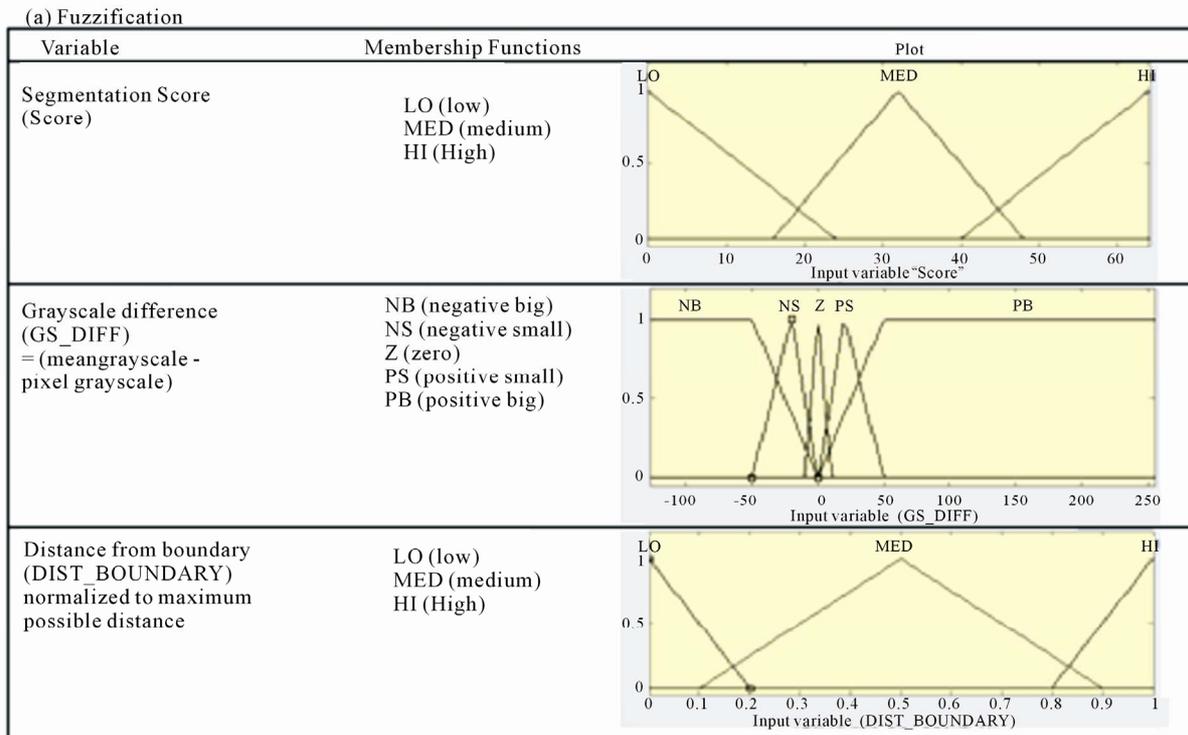
Figure 9. Post processing step 2: detection of the non-detected lesion regions using neighboring slices. (a) Non detected lesion in slice 12 of MS6. (b) and (c) detected lesions in the slices 11 and 13 respectively are intersected to recover part of the non detected lesion region. (d) Intersection provides part a new lesion region. (e) Portion of the lesion region in the slice 12 is recovered by adding (a) and (d).

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 thresh-old 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 10**. The fuzzy rules output is the decision which is binary variable have two values; either keep the pixel



(b) Fuzzy Rules

RULE	SCORE	GS DIFF	DIST BOUNORAY	DECISION
1	HI	×	×	KEEP
2	×	NB	×	KEEP
3	×	NS	×	KEEP
4	×	Z	×	KEEP
5	×	×	HI	KEEP
6	LO	PS	LO	REMOVE
7	LO	PS	MED	KEEP
8	LO	PB	LO	REMOVE
9	LO	PB	MED	REMOVE
10	MED	PS	LO	REMOVE
11	MED	PS	MED	KEEP
12	MED	PB	LO	REMOVE
13	MED	PB	MED	KEEP

(c) Defuzzification: Centroid Rule (Zheru Chi 1998 [29])

Figure 10. Fuzzy engine used in lesion regions shape correction (step 3a): 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).

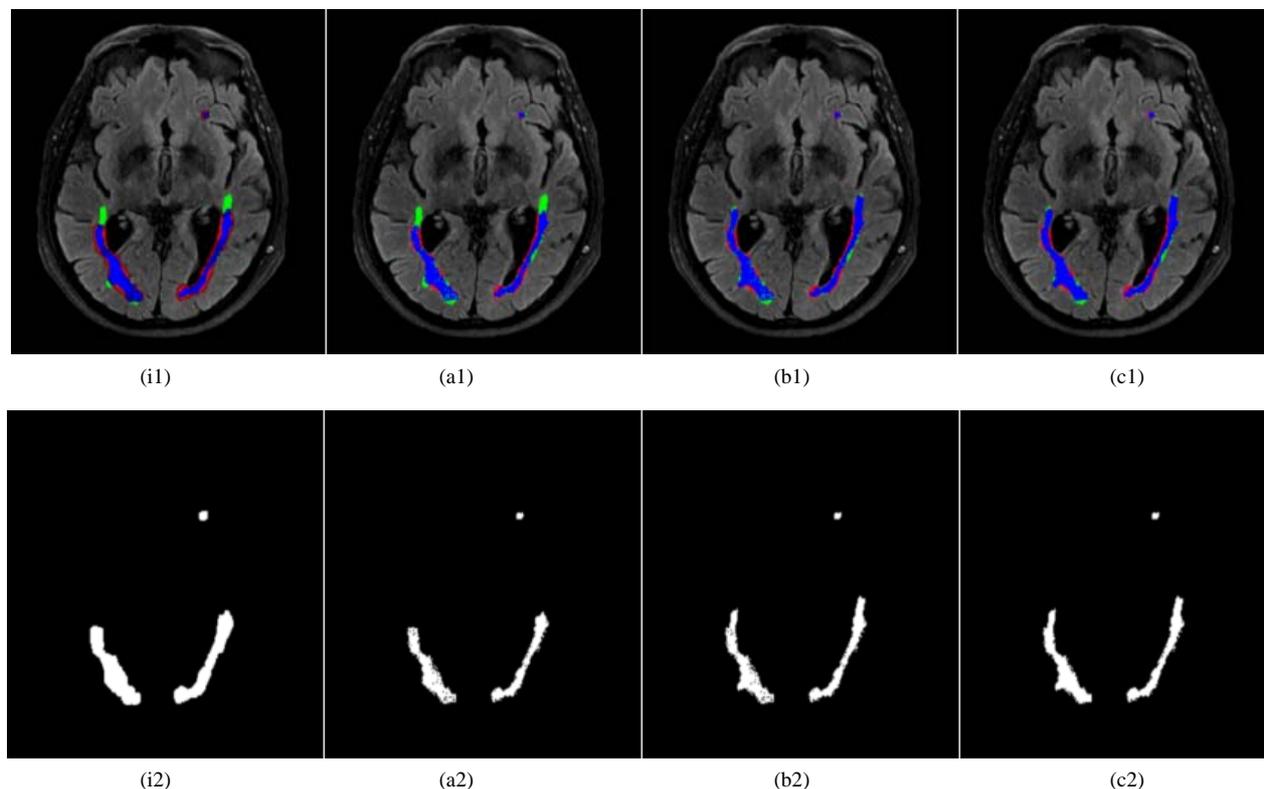


Figure 11. Post processing step 3: lesion regions shape correction. (i1) and (i2): The initial colored evaluation of the segmentation and the initial segmentation; (a1) and (a2): Effect of applying step 3a; (b1) and (b2): Effect of applying step 3b; (c1) and (c2): Effect of applying step 3c.

in the lesion region (KEEP) or remove it (REMOVE). The defuzzification is performed using the centroid rule which is used in case of classification [29]. **Figures 11(a1) and (a2)** show the effect of applying the fuzzy engine operation on a slice from MS6. 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 11-a1 and black voxels in **Figure 11(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.

(b) Elimination of the false negatives on the boundary of the lesion regions.

The colored evaluation of the segmentation depicted in **Figure 11(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. **Figures 11(b1) and (b2)** show the effect of applying this recursive region growing operation, where most of the false negative pixels were recovered.

(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. **Figures 11(c1) and (c2)** show the effect of applying this operation where all the holes were filled.

All the operations visualized in **Figure 11** 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.

2.6. Evaluation of the Proposed Method

To evaluate the performance of the proposed segmentation method, the dice similarity, sensitivity and percentage of the detected lesion load are calculated. 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: $DS = 2|X \cap Y| / (|X| + |Y|)$

As stated in [29,30], a DS score above 0.7 is generally considered as very good, especially when the segmented structures are small. Dice Similarity is calculated in our evaluation module twice. The first is calculated based on the similarity of voxels (DS_V) and the second is calculated based on the similarity of lesion regions (DS_R). DS_V uses the number of common MS voxels between manual and automatic segmentation for $|X \cap Y|$ and uses the number of MS voxels of manual and automatic segmentation for $|X|$ and $|Y|$ respectively. DS_R uses the number of common MS lesion regions between manual and automatic segmentation for $|X \cap Y|$ and uses the number of MS lesion regions of manual and automatic segmentation for $|X|$ and $|Y|$ respectively. In the context of DS_R , 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 [4]. For example, applying these definitions of DS_V and DS_R to the initial segmentations shown in **Figures 11(i1)** and **(i2)** yields values of 0.73 (good segmentation) and 1.0 (perfect segmentation), respectively.

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.

Percentage of 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.

3. Results

The dataset of ten real FLAIR MRI axial sequences were used to evaluate the performance of the proposed segmentation method. The performance metrics detailed in

Section 2.6 were calculated. The segmentation results are summarized in **Table 2**. 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.

The average metrics are 0.79 for DS_R , 0.71 for DS_V , 0.68 for sensitivity and 0.9 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 0.71, which 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 (0.77 for DS_R , 0.69 for DS_V , 0.66 for sensitivity and 0.88 for percentage of detected lesion load) which is still a very good result. 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 12**. For each study, the dice similarity based on Voxels (DS_V) is calculated before and after the use of the post processing step. The average improvement in dice similarity of the overall

Table 2. Segmentation result.

Study	Dice Similarity based on Number of Lesion Regions	Dice Similarity based on Number of Voxels	Sensitivity based on Voxels	Detected Lesion Load based on Voxels (%)
MS2	0.88	0.78	0.67	0.93
MS3	0.96	0.93	0.89	1.1
MS4	0.68	0.64	0.72	0.92
MS5	0.72	0.68	0.59	0.65
MS6	0.84	0.72	0.64	0.76
MS7	0.77	0.67	0.63	0.91
MS8	0.68	0.68	0.72	1.05
MS9	0.76	0.63	0.57	0.78
MS10	0.83	0.71	0.62	0.81
MS11	0.75	0.68	0.74	1.12
Average	0.79	0.71	0.68	0.90

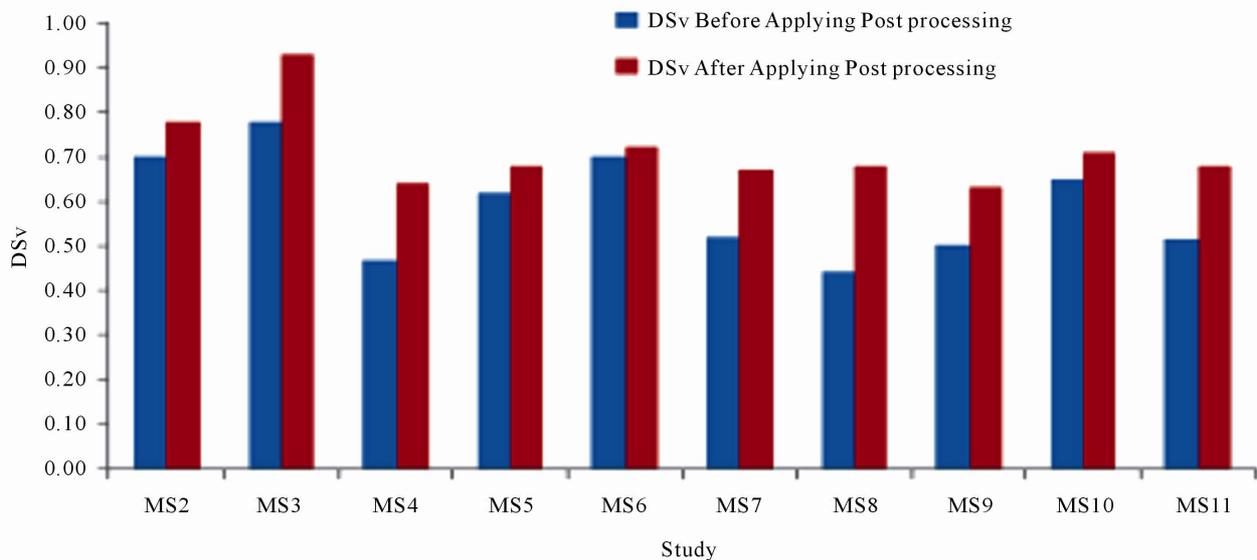


Figure 12. Effect of the post processing steps on the overall segmentation performance.

segmentation due to the post processing stage varies is 12%.

4. Discussion

A novel method for MS lesions segmentation in FLAIR sequence of brain MR images has been developed. The segmentation process goes through three steps. The first step is the preprocessing which is done to improve the brightness and contrast of all the FLAIR slices of the subject dataset. The second step is the main processing which involves using a trained SVM to detect the initial MS lesions from the individual slices using feature vector which are mainly composed of textural features. The third step is the post processing that aims to improve and refine the performance of the initial segmentation generated through the SVM-based textural segmentation. In that regard, the post processing step addresses all possible types of errors in the MS lesion segmentation results of the second step in order to reduce the overall errors including both false positives and false negatives.

The main processing classifier uses thirty four features in three categories. These categories of features are selected to have analogy with the features used non-intentionally by the expert in the task of manual labeling of MS areas. According to our observations, when the expert labels MS lesions in the FLAIR slice, the hyper intense areas are the potential areas to have the lesion. This is emulated in our technique by using the group of twenty four textural features. Candidate areas are filtered based on previous experience with the brain positions where most likely lesions occur; hence a group of two position based features is used. Besides, the expert takes into ac-

count the difference between the intensity of the lesion area and the neighboring areas intensities to take final decision and we emulate this by using the eight neighboring features. Although both textural features and neighboring features are based on intensity, no redundancy exists between them as the first group is used to aid the classifier in the detection of special pattern areas while the later is used to take into account the relation of the intensity of the area and the neighboring areas.

The main processing classifier uses an SVM engine. SVM Parameters selection and training set balancing directly affect 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 [25]. 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 $(C; \gamma)$ 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.

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 8×8 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 13** 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.

Our first contribution in this method is using textural features without manual selection of ROI which was an area for future research and improvement [1]. To the best of our knowledge, the common use of textural features in MS lesions detection was previously attempted with aid of manual selection of regions of interests (ROIs). In the

approach presented in this paper, overlapping square blocks, with adaptively determined size, are used to replace these ROIs. Connected blocks classified as MS blocks form a lesion region. In the post processing step, these lesion regions are trimmed to remove extra pixels and/or extended to include undetected pixels, as appropriate, to finally arrive at accurate, smooth, and complete lesion regions without the need for manual selection of ROIs.

The second contribution of the presented method is the post processing step that is both generic and modular which means it is independent on the previous steps. Thus, it can be generalized and applied with any other MS lesion segmentation technique to improve the performance. In this step, different types of errors in MS lesion segmentation are addressed based on logical handling of both false positives and false negatives. When used with any other technique, the score parameter used in our approach can be replaced by any other parameter that provides the probability of the pixel being part of an MS lesion based on the other technique.

One of the recent publications [4] 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 3** with our results added as the last line. 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 [4], the dice similarity was calculated based on the common regions between the manual segmentation and the automatic segmentation and similarly

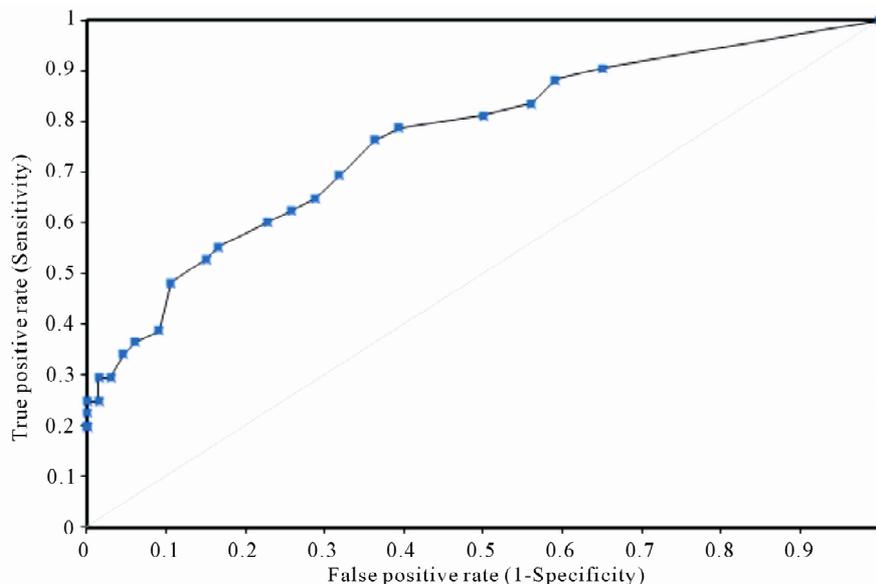


Figure 13. ROC (Receiver Operating Characteristics) curve for main processing performance.

Table 3. Comparison of the automated methods for detection of MS in MR images.

Authors	Segmentation Method	No. of studies	Average Dice Similarity (based on lesion regions)
Boudraa <i>et al.</i> 2000 [13]	Fuzzy C-Means	10	0.62
Leemput <i>et al.</i> 2001 [14]	Stochastic model	50	0.51
Zijdenbos <i>et al.</i> 2002 [15]	Pipeline analysis	29	0.68
Khayati <i>et al.</i> 2008 [21]	AMM (adaptive mixtures method), MRF (Markov random field model)	20	0.75
Yamamoto <i>et al.</i> 2010 [4]	Region growing, LSM (level set method)	6	0.77
Proposed method	Texture Analysis, SVM (support vector machines)	10	0.79

we also used the value of DS_R in **Table 3**. The table shows that our method has an average regional dice similarity of 0.79 which is the highest among the past studies. 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.

5. Conclusions

We have developed an automated method for detection of MS lesions in brain MR images using textural based SVM. The main contributions of the presented method are using textural features without manual selection of ROI and the comprehensive post processing step that handles different types of errors in MS lesions segmentation that can be generalized to improve the performance of any other MS segmentation technique. The method has been tested using ten real FLAIR MRI datasets. The performance evaluation and comparative results with other automated techniques show that our method provides competitive results for the detection of MS lesions.

6. Acknowledgements

This work was supported in part by National Institutes of Health (NIH), National Institute of Neurological Disease and Stroke (NINDS) through grant #R41NS060473.

7. References

- [1] J. Zhang, L. Z. Tong, L. Wang and N. Li, "Texture Analysis of Multiple Sclerosis: A Comparative Study," *Magnetic Resonance Imaging*, Vol. 26, No. 8, 2008, pp. 1160-1166. [doi:10.1016/j.mri.2008.01.016](https://doi.org/10.1016/j.mri.2008.01.016)
- [2] C. H. Polman, *et al.*, "Diagnostic Criteria for Multiple Sclerosis: 2005 Revisions to the 'McDonald Criteria'," *Annals of Neurology*, Vol. 58, No. 6, 2005, pp. 840-856. [doi:10.1002/ana.20703](https://doi.org/10.1002/ana.20703)
- [3] 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, No. 1, 1998, pp. 3-24. [doi:10.1093/brain/121.1.3](https://doi.org/10.1093/brain/121.1.3)
- [4] D. Yamamoto, *et al.*, "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, 2010, pp. 404-413. [doi:10.1016/j.compmedimag.2010.02.001](https://doi.org/10.1016/j.compmedimag.2010.02.001)
- [5] E. Geremia, *et al.*, "Spatial Decision Forests for MS Lesion Segmentation in Multi-Channel MR Images," *Medical Image Computing and Computer-Assisted Intervention*, Vol. 6361, 2010, pp. 111-118.
- [6] C. J. Wallace, T. P. Seland and T. C. Fong, "Multiple Sclerosis: The Impact of MR Imaging," *American Journal of Roentgenology*, Vol. 158, 1992, pp. 849-857.
- [7] S. Wiebe, *et al.*, "Serial Cranial and Spinal Cord Magnetic Resonance Imaging in Multiple Sclerosis," *Annals of Neurology*, Vol. 32, No. 5, 1992, pp. 643-650. [doi:10.1002/ana.410320507](https://doi.org/10.1002/ana.410320507)
- [8] L. Truyen, "Magnetic Resonance Imaging in Multiple Sclerosis: A Review," *Acta Neurologica Belgica*, Vol. 94, 1994, pp. 98-102.
- [9] F. Fazekas, *et al.*, "The Contribution of Magnetic Resonance Imaging to the Diagnosis of Multiple Sclerosis," *Neurology*, Vol. 53, 1999, pp. 448-456.
- [10] P. Anbeek, K. L. Vincken and M. A. Viergever, "Automated MS-Lesion Segmentation by K-Nearest Neighbor Classification," *MIDAS Journal*, 2008. <http://hdl.handle.net/10380/1448>
- [11] F. Rousseau, F. Blanc, J. de Seze, L. Rumbach and J. Armspach, "An a Contrario Approach for Outliers Segmentation: Application to Multiple Sclerosis in MRI," *5th IEEE International Symposium*, Paris, 2008, pp. 9-12.
- [12] B. Johnston, M. S. Atkins, B. Mackiewicz and M. Anderson, "Segmentation of Multiple Sclerosis Lesions in

- Intensity Corrected Multispectral MRI," *IEEE Transactions on Medical Imaging*, Vol. 15, No. 2, 1996, pp. 154-169. [doi:10.1109/42.491417](https://doi.org/10.1109/42.491417)
- [13] A. O. Boudraa, *et al.*, "Automated Segmentation of Multiple Sclerosis Lesions in Multispectral MR Imaging Using Fuzzy Clustering," *Computers in Biology and Medicine*, Vol. 30, No. 1, 2000, pp. 23-40.
- [14] 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, 2001, pp. 677-688. [doi:10.1109/42.938237](https://doi.org/10.1109/42.938237)
- [15] 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*, Vol. 21, No. 10, 2002, pp. 1280-1291. [doi:10.1109/TMI.2002.806283](https://doi.org/10.1109/TMI.2002.806283)
- [16] F. Kruggel, S. P. Joseph and H.-J. Gertz, "Texture-Based Segmentation of Diffuse Lesions of the Brain's White Matter," *NeuroImage*, Vol. 39, No. 3, 2008, pp. 987-996. [doi:10.1016/j.neuroimage.2007.09.058](https://doi.org/10.1016/j.neuroimage.2007.09.058)
- [17] W. F. Liu, X. X. Zhou, G. L. Jiang and L. Z. Tong, "Texture Analysis of MRI in Patients with Multiple Sclerosis Based on the Gray-Level Difference Statistics," *Education Technology and Computer Science*, Vol. 3, 2009, pp. 771-774.
- [18] A. Pozdnukhov and M. Kanevski, "Monitoring Network Optimisation for Spatial Data Classification Using Support Vector Machines," *International Journal of Environment and Pollution*, Vol. 28, No. 3-4, 2006, pp. 465-484.
- [19] M. Kanevski, M. Maignan and A. Pozdnukhov, "Active Learning of Environmental Data Using Support Vector Machines," *Conference of the International Association for Mathematical Geology*, Toronto, 21-26 August 2005.
- [20] R. R. Edelman, J. R. Hesselink, M. B. Zlatkin and J. V. Crues, "Clinical Magnetic Resonance Imaging," 3rd Edition, Elsevier, New York, 2006.
- [21] R. Khayati, M. Vafadust, F. Towhidkhan 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 Medicine*, Vol. 38, 2008, pp. 379-390.
- [22] W. I. McDonald, *et al.*, "Recommendation Diagnostic Criteria for Multiple Sclerosis: Guidelines from the International Panel on the Diagnosis of Multiple Sclerosis," *Annals of Neurology*, Vol. 50, 2001, pp. 121-127.
- [23] A. Younis, M. Ibrahim, M. Kabuka and N. John, "An Artificial Immune-Activated Neural Network Applied to Brain 3D MRI Segmentation," *Journal of Digital Imaging*, Vol. 21, Suppl. 1, 2008, pp. S69-S88.
- [24] J. Rexilius, H. K. Hahn, H. Bourquain and H.-O. Peitgen, "Ground Truth in MS Lesion Volumetry—A Phantom Study," *Lecture Notes in Computer Science*, Vol. 2879, 2003, pp. 546-553.
- [25] C.-C. Chang and C.-J. Lin, "LIBSVM: A Library for Support Vector Machines," *ACM Transactions on Intelligent Systems and Technology*, Vol. 2, No. 3, 2011, pp. 1-27.
- [26] A. K. Jain, R. P. W. Duin and J. Mao, "Statistical Pattern Recognition: Review," *IEEE Transactions Pattern Analysis and Machine Intelligence*, Vol. 22, No. 1, 2000, pp. 4-37. [doi:10.1109/34.824819](https://doi.org/10.1109/34.824819)
- [27] D. Garcia-Lorenzo, L. Lecoeur, D. Arnold and D. L. Collins, "Multiple Sclerosis Lesion Segmentation Using an Automatic Multimodal Graph Cuts," *Medical Image Computing and Computer Assisted Intervention*, London, 20-24 September 2009, pp. 584-591.
- [28] 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, 2001, pp. 756-766.
- [29] Z. R. Chi, H. Yan and T. Pham, "Fuzzy Algorithms: With Application to Image Processing and Pattern Recognition," World Scientific Publishing Co. Pte. Ltd, Singapore, 1996.
- [30] J. Lecoeur, *et al.*, "Multiple Sclerosis Lesions Segmentation using Spectral Gradient and Graph Cuts," *Proceedings of MICCAI Workshop on Medical Image Analysis on Multiple Sclerosis*, New York, September 2008.

Appendix A. Textural Features 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 , $x = 0,1,\dots, N - 1$ and $y = 0,1,\dots, M - 1$. The function $f(x,y)$ can take any value $I = 0,1,\dots,G - 1$ where G is total number of intensity levels in the image.

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.

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

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:

$$p(i) = h(i)/NM$$

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

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

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

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

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

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 3×3 pixels neighborhood, g is defined as follows:

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

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

Mean of absolute gradient

$$(GrMean) = \frac{1}{NM} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} g(x, y)$$

Variance of absolute gradient

$$(GrVariance) = \frac{1}{NM} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} (g(x, y) - GrMean)^2$$

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

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:

Short run emphasis inverse moments

$$(ShrtREmph) = \left(\sum_{i=0}^{G-1} \sum_{j=1}^{N_r} \frac{r(i, j)}{j^2} \right) / C$$

Long run emphasis moments

$$(LngREmph) = \left(\sum_{i=0}^{G-1} \sum_{j=1}^{N_r} j^2 r(i, j) \right) / C$$

Gray level non-uniformity

$$(GLvNonUni) = \left(\sum_{i=0}^{G-1} \left(\sum_{j=1}^{N_r} r(i, j)^2 \right) \right) / C$$

Run length non-uniformity

$$(RLNonUni) = \left(\sum_{j=1}^{N_r} \left(\sum_{i=0}^{G-1} r(i, j) \right)^2 \right) / C$$

where the normalization coefficient C is defined as follows:

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

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 $f(x_1, y_1) = i$ and $f(x_2, y_2) = j$ where $(x_2, y_2) = (x_1, y_1) + (dcos\theta, dsin\theta)$. Usually the co-occurrence matrix is calculated for $d = 1$ and 2 with angles $\theta = 0^\circ, 45^\circ, 90^\circ$ 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:

Angular second moment

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

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

$$\text{Correlation} = \frac{\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} ij co(i, j) - \mu_x \mu_y}{\sigma_x \sigma_y}$$

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

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