Feature Selection For Self-Organizing Feature Map Neural Networks With Applications in Medical Image Segmentation

By

Heather Erin Rickard
B.S.E.E., University of Louisville, 1999

A Thesis
Submitted to the Faculty of the
University of Louisville
Speed Scientific School
as Partial Fulfillment of the Requirements
for the Professional Degree

MASTER OF ENGINEERING

Department of Electrical Engineering

December 2001

Feature Selection For Self-Organizing Feature Map Neural Networks With Applications in Medical Image Segmentation

Submitted by: _____
Heather Erin Rickard

A Thesis Approved on

_____
(Date)

by the Following Reading and Examination Committee:

_____
Dr. Aly A. Farag, Thesis Director

_____
Dr. Jacek Zurada

_____
Dr. James Graham

ABSTRACT

This thesis presents a novel feature selection algorithm for medical image segmentation. A multiobjective optimization genetic algorithm is used to search among the candidate features to find an optimal subset which results in the highest segmentation. A self-organizing feature map serves as the classifier such that the fitness of the genetic algorithm is determined by two quality measures of the map, quantization error and topology preservation. The algorithm is applied to a 3D simulation model of the human brain and six MRI data sets, and shows promising results in comparison with using principal component analysis as the basis for feature selection. This indicates that tailoring a self-organizing feature map to a specific subset of features has the potential to increase the segmentation accuracy of medical images.

TABLE OF CONTENTS

NOMENCLATURE

$I(x, y)$	=	Intensity value of the pixel located at row x and column y in image I
$\mathbf{X}$	=	Feature vector
x_{i}	=	Feature number i
D	=	Dimension
μ	=	Mean
σ	=	Standard deviation
W	=	2D window of size w x w
d	=	distance
θ	=	direction
S	=	Sum histogram
D	=	Difference histogram
C_{θ°	=	Scaling factor for sum and difference histograms of direction θ
G_o	=	Gaussian kernel of standard deviation σ
η	=	Regular moments
m	=	Central moments
ϕ	=	Nonlinear combination of normalized central moments
P	=	Feature pattern
P'	=	Sample of feature vectors
$\mathbf{w}$	=	Weight vector

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

Modern medical imaging technology such as magnetic resonance imaging (MRI), x-ray computer tomography (CT), positron emission tomography (PET), and ultrasound has given physicians a non-invasive means to visualize internal anatomical structures and diagnose a wide variety of diseases. MRI is a particularly powerful and versatile modality. Compared to other such techniques, MRI has superior soft tissue differentiation [1], high spatial resolution and contrast [2], and does not use ionizing radiation which may be harmful to patients [3]. Such characteristics have shown MRI to be a valuable tool in the clinical and surgical environment [4].

A. Medical Image Segmentation

MR images are typically interpreted visually and qualitatively by radiologists. The need for quantitative information, however, is becoming increasingly important in the clinical and surgical environment [5]. The segmentation of MR images into anatomical tissues, structures, and fluids would allow their volumes to be calculated, providing quantitative measurements useful for a variety of applications. The delineation of tumors before and after radiation therapy for response evaluation, the assessment of drug treatment therapies, and the automatic or semiautomatic outlining of areas to be treated prior to surgery are a few possible areas of application [6], [7]. The quantitative measurement of brain atrophy and tumor volume also contribute to the study of certain pathologies, such as Alzheimer's disease, multiple sclerosis, and schizophrenia [7].

Alzheimer's disease, a degenerative disease of the brain [8], is the most common cause of dementia in the western world [9]. At present, it has no known cause and is very difficult to diagnose, with confirmation being obtained only after death by an autopsy. MR images of patients in the later stages of the disease show a decrease in brain size, especially around the hippocampus and the temporal lobes, as well as enlarged ventricles, sulcal broadening, and a reduction in the gyral size [9], [10]. It is hoped that MRI will make it possible to diagnose the presence of Alzheimer's disease before the manifestation of any physical symptoms and to discover the cause of the degeneration of the brain.

The routine quantitative analysis of MR images is impractical by manual methods, since manual segmentation is time-consuming, costly, and tedious [4]. Errors due to low tissue contrast, ambiguous tissue boundaries caused by partial volume effects (where individual pixels represent more than one tissue type), and poor hand-eye coordination commonly occur [7], [11]. The lack of a standard in MRI further complicates the problem; not only can images of the same subject obtained with two different MRI machines have different characteristics [12], but the intensity values for similar tissues obtained with a single MRI machine can vary across one image volume and within a single slice [7]. Although the automatic or semiautomatic segmentation of medical images is highly desirable, such a method is still not available for clinical use [13].

Several image segmentation systems have been reported in the literature [1], [2], [5]-[7], [13]-[16], [19]-[25]. These systems may be divided into four broad approaches: classical, statistical, fuzzy, and neural network techniques [13]. Classical approaches include such methods as edge- and region-based techniques [14], [15], or intensity

thresholding [16], and typically do not utilize any *a priori* information. Statistical classifiers, such as expectation-maximization (EM) algorithms [1], [5], maximum likelihood estimators [17], and Markov random fields [13], represent learned classification rules as mathematical formulas. The concept of a fuzzy set, which allows a gradual transition from membership to nonmembership [18], is the basis for fuzzy segmentation techniques [19], [20]. A multitude of neural network based approaches have been introduced [2], [6], [21]-[25], which rely on a neural network architecture for image segmentation.

The segmentation of MR images may be generalized as a type of pattern classification problem, where anatomically meaningful labels are assigned to each component of the segmented image according to its classification. The classification model for pattern recognition contains three parts: a transducer, a feature extractor, and a classifier [17]. In this case, the transducer is the MRI process, resulting in images suitable for processing. The feature extractor measures certain properties which should characterize one input pattern from another, and the classifier uses these input patterns to assign the data to one of a finite number of categories. In the case of image segmentation, there are potentially hundreds of possible features available with which to characterize the data set; feature selection aims to find an optimal subset of candidate features, and is an important part of the segmentation process. This thesis focuses primarily on the problem of feature extraction and feature selection.

Thesis Overview

In the feature extraction stage, four types of features are calculated: first order parameters, textural measures, multiscale features, and moment invariant features. First

order parameters describe the gray level of a single pixel or the gray level distribution in an area centered around the pixel. A type of second order statistic, textural measures take into account the spatial interdependence of the pixels. Multiscale features are created by decomposing an image into a set of descriptions at a specific scale. Moment invariant features are invariant under translation, scaling, and rotation, and are an indication of the center, spread, and skewness of the image.

Once the feature pattern has been constructed, the goal is to find an optimal subset of these features which will result in the most accurate image segmentation. To accomplish this, a multiobjective optimization genetic algorithm has been implemented to search among the candidate features. The classifier used in this system is a self-organizing feature map, a type of Kohonen neural network. The genetic algorithm evaluates the effectiveness of a particular subset of features on the self-organizing feature map based on two criteria, resolution and topology preservation of the map. Ideally, a subset of features which result in good feature map quality will also result in high segmentation accuracy. One of the goals of this thesis is to discover if such a relationship exists.

Similar methods have been reported [26]-[32] in the literature. The classification systems used have included a nearest neighbor classifier [26], a rule-based classification system [27], a Euclidean distance measure [28], and various types of supervised neural networks [29]-[32]. In all these cases, *a priori* knowledge of the data set was required. The novelty of the approach described in this thesis is the use of an unsupervised self-organizing feature map as the classification system. No *a priori* information about the data set is needed, since the fitness of the genetic algorithm is determined by the quality

of the map itself. Furthermore, the genetic algorithm employed uses multiobjective optimization in its development. Although multiobjective optimization has been tried in this type of problem before [32], the neural network used still relied upon *a priori* knowledge of the input data set.

A 3D geometrical model of the brain, proposed by M. N. Ahmed [33], has been used to test the performance of the segmentation algorithm. This model, similar in concept to the famous 2D Shepp-Logan head model, consists of ellipsoids of known dimension and gray levels. The segmentation algorithm was also applied to six real MRI data sets to estimate its usefulness in a clinical environment.

Thesis Organization

The remainder of this thesis proceeds as follows. Section II details each of the four types of features chosen for the feature extraction process and how the feature pattern is constructed. Section III discusses self-organizing feature maps, genetic algorithms, the multiobjective optimization requirement, and the quality measures used in this thesis. An overview of the system design, a discussion of the brain model used for testing, and details about the MRI data set are presented in Section IV. Segmentation results are reported in Section V, along with a comparison of the use of principal component analysis as a method for feature selection. Finally, Section VI contains the conclusions and recommendations of the thesis.

II. FEATURE EXTRACTION

The purpose of feature extraction is to reduce the original data set by measuring certain properties, or features, that distinguish one input pattern from another [17]. The selected features should provide the characteristics of the input type to the classifier by condensing the description of the relevant properties of the image into a feature space of D dimension. In this application, the pixels (short for "picture elements") of the gray level image I are transformed to a feature vector,

$$\mathbf{X} = [x_1, x_2, \cdots x_D] \tag{1}$$

where each x_i is a unique feature and D is the dimension of the feature vector.

A variety of approaches have been developed for feature extraction [4], [33], [34], including the Fourier transform, moment feature space, orthogonal polynomials, fractal dimension measures, Gabor functions, and Markov random fields. The features used here will be derived from first order gray level parameters, textural measures, multiscale features, and moment invariant features. Once these feature vectors are obtained, they will be sent to the feature selector to determine which are the most discriminating for image segmentation.

A. First Order Features

First order features describe the gray level of a single pixel or the gray level distribution in a 2D window, W, of size w x w, centered around the pixel. These features do not take the spatial interdependence of the pixels into account. In this category, three features were selected:

1. The gray level,

$$g = I(x, y) \tag{2}$$

2. The mean gray level,

$$mgl = \frac{1}{w^2} \sum_{(x,y)\in W} I(x, y) \tag{3}$$

3. The variance gray level,

$$vgl = \frac{1}{w^2 - 1} \sum_{(x,y)\in W} (I(x, y) - mgl)^2 \, . \tag{4}$$

A sliding window, w = 8, was used to calculate the mean and variance for each pixel in the image.

B. Textural Features

Although there is no agreement on a formal definition of texture, it can be described as the "repetition of a pattern or patterns over a region" [35]. These patterns may be qualitatively assessed as being fine, coarse, smooth, random, or lineated, for example [36]. Textural features are a type of second order statistic, where the differences between the gray levels of two pixels at different locations are compared. These features may be divided into one of two classes, statistical and structural. Statistical texture can be defined as the spatial distribution of intensity variations in an image according to some underlying probabilistic model, whereas structural texture is the spatial distribution of a set of primitives in an image based on some predefined rules [37].

Several approaches have been developed to derive textural features, such as first-order gradient distribution, the edge co-occurrence matrix, the run-length matrix, or the gray-level co-occurrence matrix [33]. For this thesis, textural features are acquired using a statistical method based on the use of sum and difference histograms, developed by

Unser [35], which is based on the gray level co-occurrence matrix described by Haralick et al [36].

The co-occurrence matrix is an estimate of the second order joint probability, $p(i, j)$, which is the probability of going from gray level i to gray level j, given the distance d between two pixels along a given direction θ. The matrix is square, where the dimension is the number of discrete gray levels in the image. Four directions are usually considered, θ = 0°, 45°, 90°, and 135°, although to guarantee that the texture features are invariant under rotation, Haralick et al [36] suggest using the average of the 4 directions. A distance of d = 1 or 2 is typically used as well. Two forms of the co-occurrence matrix exist, (1) symmetric, where pairs separated by d and $-d$ are counted, and (2) not symmetric, where only separation by a distance d is counted. Once the matrix is formed, it is normalized by the number of relationship pairs. The co-occurrence matrix is then analyzed and features (Haralick et al described 14 such features) are computed from the matrix.

Although this method is known for its good discrimination abilities [35], in practice it is slow and memory intensive. For each texture to be analyzed, a new co-occurrence matrix must be created, followed by a calculation of the value for the feature. Since it is only the features that are desired and not the co-occurrence matrix itself, Unser [35] devised a method for computing the features without creating the matrix, based on sum and difference histograms.

The sum histogram S is simply the histogram of the sums of all pixels d_x and d_y apart. For example, the gray level at $I(x, y)$ is added to the gray level at $I(x + d_x, y + d_y)$, and the histogram bin corresponding to that sum is incremented.

Similarly, the difference histogram D is the histogram bin of the gray level differences. For example, for the image show in Figure 1 with three gray levels, $g \in (0, 1, 2)$,

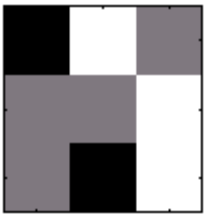


FIGURE 1 – Sample 3 x 3 Image With Three Gray Levels. This Image Is Used To Illustrate How Texture Measures Are Calculated.

the sum and difference histograms based on a distance $d = 1$, and direction $\theta = 90°$ are shown in Figure 2,

$$\begin{bmatrix} 0 & 2 & 1 \\ 1 & 1 & 2 \\ 1 & 0 & 2 \end{bmatrix}, \quad d = 1, \quad \theta = 90°$$

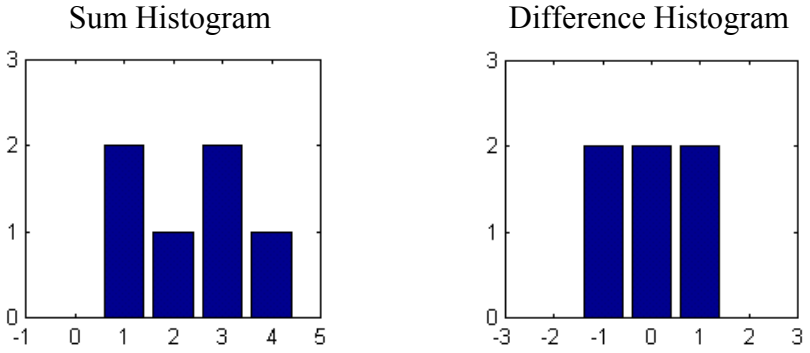


FIGURE 2 – Sample Sum And Difference Histograms For Figure 1 With Distance $d = 1$ And Direction $\theta = 90°$.

As with the co-occurrence matrix, the average of directions $\theta = 0°$, 45°, 90°, and 135° is used. The histograms S and D are then normalized so that their values become probabilities. For an image, I, of dimension M x N, the scaling factors are as follows [36],

$$
\begin{aligned}
C_{0^\circ} &= M(N-d) \\
C_{45^\circ} &= (M-d)(N-d) \\
C_{90^\circ} &= (M-d)N \\
C_{135^\circ} &= (M-d)(N-d)
\end{aligned}
\tag{5}
$$

Five features were calculated from these histograms:

1. Mean,
$$\frac{1}{2}\sum_i i \cdot S(i) \tag{6}$$

2. Contrast,
$$\sum_j j^2 \cdot D(j) \tag{7}$$

3. Homogeneity,
$$\sum_j \frac{1}{1+j^2} \cdot D(j) \tag{8}$$

4. Entropy,
$$-\sum_i S(i) \cdot \log(S(i)) - \sum_j D(j) \cdot \log(D(j)) \tag{9}$$

5. Energy,
$$\sum_i S(i)^2 \cdot \sum_j D(j)^2 \,. \tag{10}$$

This application is somewhat different from texture analysis in that the output is an image, preferably of the same size and with approximately the same spatial characteristics [38]. This may be handled by dividing the image to be analyzed into a number of square windows and to set the value of each pixel in the window to that of a calculated textural feature. For this thesis, a window size of 8 x 8 was used so that the window would be large enough to capture the textures, while being small enough to maintain an acceptable resolution.

C. Multiscale Features

The scale-space is a one-parameter family of blurred replicas of the input image, proposed by Witkin [39] and Koenderink [40]. Multiscale features are created by decomposing an image into a set of descriptions in the scale-space, at a specific scale. For an image *I*, the property,

$$\frac{\partial I}{\partial x} \otimes G_\sigma = \frac{\partial G_\sigma}{\partial x} \otimes I \tag{11}$$

shows that all image derivatives can be obtained by convolving the image with the corresponding derivative of a Gaussian kernel, G_σ [25],

$$G_\sigma(x, y) = \frac{1}{(\sigma\sqrt{2\pi})^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \tag{12}$$

(noting that images are 2D), where the one-parameter is the standard deviation ("width" or "scale") of the Gaussian kernel. The zero-th order Gaussian kernel, together with all its partial derivatives, form a complete family of scaled differential operators. It is the only such family satisfying the constraints of linearity, and invariance to shift, rotation, and scale, and has been established as the unique scale-space operator to change scale [41]. From these operators, image structures may be described completely up to any desired order at various scales. Furthermore, Gaussian filtering serves to regularize the image, making subsequent differentiation operations mathematically well-posed [41].

The Gaussian partial derivatives alone are not invariant under a particular coordinate transformation [42], so it is necessary to construct particular combinations of derivatives as invariant features. Two features each were constructed for four different values of σ.

1. Squared gradient of L,

$$L_i L_i = L_x^2 + L_y^2 \tag{13}$$

2. Laplacean of L,

$$L_{ii} = L_{xx} + L_{yy} \tag{14}$$

where,

$$L_x = I \otimes \frac{\partial G_\sigma}{\partial x}, \ldots \text{ and } L_{xx} = I \otimes \frac{\partial G_\sigma}{\partial x \partial x}, \ldots \tag{15}$$

D. Moment Invariant Features

One of the traditional methods for invariant feature extraction involves the use of moment invariants. The use of moments for image analysis and object representation was introduced in 1962 by Hu [4]. The regular moments of an image are defined by

$$\eta_{pq} = \sum_x \sum_y x^p y^q I(x, y) \tag{16}$$

where $p, q = 0, 1, 2, \ldots$. To make these moments invariant to translation, central moments are defined by,

$$m_{pq} = \sum_x \sum_y (x - x')^p (y - y')^q I(x, y) \tag{17}$$

where

$$x' = \frac{\eta_{10}}{\eta_{00}} \quad \text{and} \quad y' = \frac{\eta_{01}}{\eta_{00}}. \tag{18}$$

These central moments can then be normalized for scale invariance using

$$\mu_{pq} = \frac{m_{pq}}{m_{00}^\gamma} \tag{19}$$

where

$$\gamma = \frac{(p + q)}{2} + 1. \tag{20}$$

The following seven nonlinear combinations of normalized central moments developed by Hu [4] are invariant under translation, scaling, and rotation,

1. $\phi_1 = (\mu_{20} + \mu_{02})$ (21)

2. $\phi_2 = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2$ (22)

3. $\phi_3 = (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2$ (23)

4. $\phi_4 = (\mu_{30} + 3\mu_{12})^2 + (\mu_{21} + \mu_{03})^2$ (24)

5. $$\phi_5 = (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] + (3\mu_{21} + \mu_{03})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{12} + \mu_{03})^2]$$ (25)

6. $$\phi_6 = (\mu_{20} + \mu_{02})[(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03})$$ (26)

7. $$\phi_7 = (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] - (\mu_{30} - 3\mu_{12})(\mu_{12} + \mu_{03})[(3\mu_{30} + \mu_{12})^2 - (\mu_{12} + \mu_{03})^2]$$ (27)

In the first two equations, ϕ_1 and ϕ_2 provide scale and translation independence, equations $\phi_3 - \phi_6$ ensure rotation with reflection invariance, and ϕ_7 provides reflection discrimination in it's sign [4]. As with the textural features, the moment invariant features are calculated in an 8 x 8 window, and the value of each pixel in that window is set to that feature value.

E. Constructing the Feature Pattern

Once the features have been obtained for the entire image, they are ordered into a matrix *P*, where each row represents all the features for a single pixel, and each column represents a particular feature for all the pixels in the image. Thus,

$$\mathbf{X} = [x_1, x_2, \cdots x_D], \quad (28)$$

where *D* represents the number of features, and

$$P = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_N \end{bmatrix}, \tag{29}$$

where N represents the total number of pixels.

Since distance measures will be evaluated in the feature space, as discussed in Section III, the features need to be normalized so that no one feature dominates the others. To normalize the feature vectors to have zero mean and unit variance, the normalization

$$x_i' = \frac{x_i - \mu}{\sigma_x}, \tag{30}$$

where μ is the mean and σ_x is the standard deviation for feature x_i, has been used [25]. The complete listing of the features is detailed in Table I. These feature patterns are then sent to the feature selector to determine which are the most discriminating for image segmentation.

TABLE I
LIST OF TWENTY-THREE EXTRACTED FEATURES WHICH FORM THE PATTERN MATRIX *P*.

	Feature	**Feature Class**
x_1	Intensity	First-order
x_2	Mean	First-order
x_3	Variance	First-order
x_4	Mean	Texture
x_5	Contrast	Texture
x_6	Homogeneity	Texture
x_7	Entropy	Texture
x_8	Energy	Texture
x_9	$L_{\mathrm{i}}L_{\mathrm{i}}$, $\sigma = 0.5$	Multiscale
x_{10}	$L_{\mathrm{i}}L_{\mathrm{i}}$, $\sigma = 1.0$	Multiscale
x_{11}	$L_{\mathrm{i}}L_{\mathrm{i}}$, $\sigma = 2.0$	Multiscale
x_{12}	$L_{\mathrm{i}}L_{\mathrm{i}}$, $\sigma = 4.0$	Multiscale
x_{13}	L_{ii} , $\sigma = 0.5$	Multiscale
x_{14}	L_{ii} , $\sigma = 1.0$	Multiscale
x_{15}	L_{ii} , $\sigma = 2.0$	Multiscale
x_{16}	L_{ii} , $\sigma = 4.0$	Multiscale
x_{17}	ϕ_1	Moment Invariant
x_{18}	ϕ_2	Moment Invariant
x_{19}	ϕ_3	Moment Invariant
x_{20}	ϕ_4	Moment Invariant
x_{21}	ϕ_5	Moment Invariant
x_{22}	ϕ_6	Moment Invariant
x_{23}	ϕ_7	Moment Invariant

III. FEATURE SELECTION

In the case of image segmentation, there are potentially hundreds of possible features available with which to characterize the data set. Such a large number of features typically includes many "garbage features" [26], which are not only useless for classification purposes, but may degrade the performance of a classifier as well as increase the cost and running time of a system. At the same time, there is a potentially opposing need to include a sufficient number of features to achieve high classification accuracy.

In image understanding, finding a minimal set of features necessary for classification is an important part of designing an efficient and robust system [28]. This has led to the development of a variety of search techniques to find an "optimal" subset of features from a larger set of candidate features. The goal of feature selection is to find this optimal subset.

In this work, a multiobjective optimization genetic algorithm has been implemented to search among the candidate features for an optimal subset of features to obtain the highest classification accuracy. A self-organizing feature map will serve as the classifier for this system.

A. Self-Organizing Feature Maps

An artificial neural network (ANN) is an information-processing model or system that attempts to simulate the adaptive biological learning abilities of the mammalian brain. The architecture of a neural network is composed of a large number of highly

interconnected processing elements, referred to as *neurons* or *nodes*. As opposed to a digital model, in which all computations manipulate ones and zeros, the neural network operates by creating weighted connections between its nodes, which represent the knowledge of the system. Neural networks can be roughly categorized into two types in terms of their learning features: supervised learning algorithms, where networks learn to fit known inputs to known outputs, and unsupervised learning algorithms, where no desired output to a set of inputs is defined. ANN models are preferred for image segmentation applications because of their parallel processing, learning, and decision-making abilities [4].

Self-organizing feature maps (SOFMs) are a type of neural network developed by Kohonen. The SOFM uses an unsupervised learning method to map high dimensional data into a 1D, 2D, or at most 3D data space, subject to a topological ordering constraint. A major advantage is that the clustering produced by the SOFM retains the underlying structure of the input space, while the dimensionality of the space is reduced. As a result, a neuron map is obtained with weights encoding the stationary probability density function $p(x)$ of the input pattern vectors [43].

1. Network Topology

The SOFM consists of two main parts, the input layer and the output map. The dimensionality of the input layer is not restricted, while the output map has dimensionality 1D, 2D, or 3D. An example of a planar array of neurons with hexagonal neighborhoods is shown in Figure 3. Each node i of the map is connected to the input feature vector, $\mathbf{x} = [x_1\ x_2\ x_3\ x_4]$, with a weight vector, $\mathbf{w} = [w_{i1}\ w_{i2}\ w_{i3}\ w_{i4}]$.

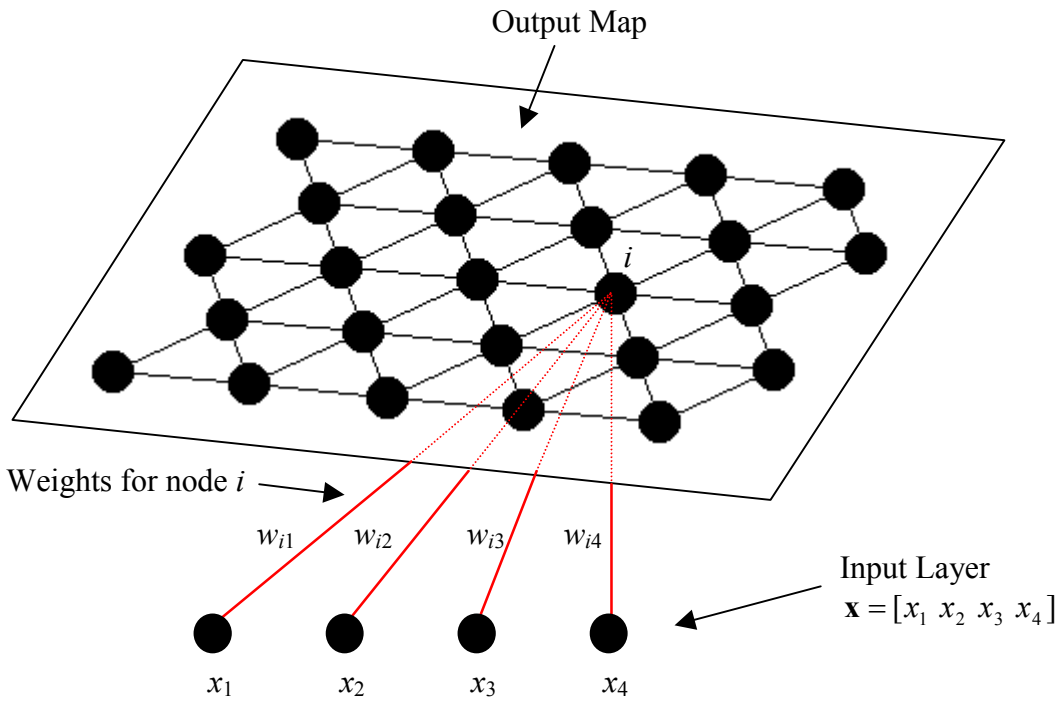


FIGURE 3 – Example Of Mapping Feature Vector **x** To SOFM. The Feature Vector Is Connected To The Output Map Via The Weight Vector, **w**.

The weight values between the input and output nodes are assigned prior to learning. Traditionally, weight values were often assigned randomly to demonstrate the strong self-organizing tendency of the SOFM; however, if the initial values for the weight vectors are assigned linearly, the computation of the SOFM can be made considerably faster. Since the SOFM is already approximately organized in the beginning, a narrower neighborhood function and smaller learning-rate factor may be used initially [44].

The learning procedure is a type of competitive learning, where only the winning node and its neighboring nodes are updated during the learning process. The topological neighborhood definition for the previous example is shown in Figure 4. The winning output node is determined by a similarity measure, which can be the Euclidean distance measure or the dot product of two vectors. The best match or the minimum distance

measure obtained defines the winning node in the output layer. For this thesis, the Euclidean distance measure,

$$\|\mathbf{x} - \mathbf{w}_m\| = \min_i \{\|\mathbf{x} - \mathbf{w}_i\|\} \tag{31}$$

where $\mathbf{w}_i$ represents the winner neuron, has been utilized.

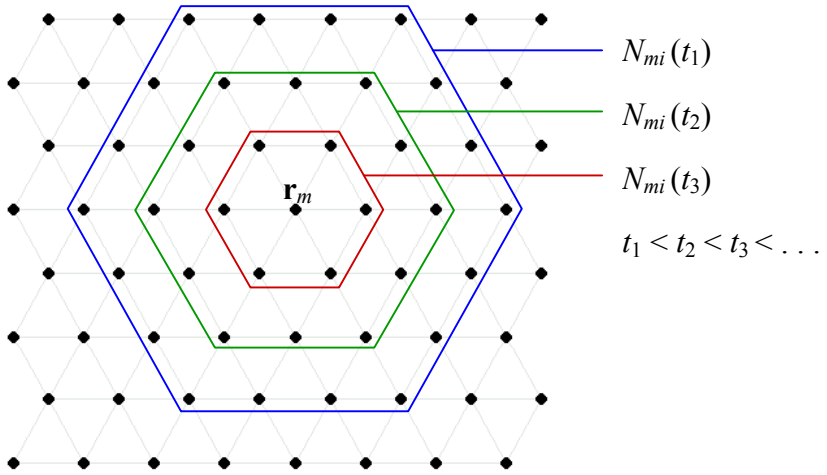


FIGURE 4 – Example Of Topological Neighborhood Definition For A 2D SOFM With Hexagonal Neighborhoods.

The SOFM update rule for the weight vector of unit *i* is

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + N_{mi}(t)[\mathbf{x}(t) - \mathbf{w}_i(t)] \tag{32}$$

where *t* denotes time, $\mathbf{x}(t)$ is an input vector drawn from the input data set at time *t*, and $N_{mi}(t)$ is the neighborhood kernel around the winner unit *m*. The neighborhood function in this case is the Gaussian,

$$N_{mi}(t) = \alpha(t)\exp\left[\frac{-\|\mathbf{r}_i - \mathbf{r}_m\|}{\sigma^2(t)}\right] \tag{33}$$

where $\mathbf{r}_m$ and $\mathbf{r}_i$ are the position vectors of the winning node and of the winning neighborhood nodes, respectively. The learning rate factor, $0 < \alpha(t) < 1$, decreases

monotonically with time, and $\sigma(t)$ corresponds to the width of the neighborhood function, also decreasing monotonically with time. Thus, the winning node undergoes the most change, while the neighborhood nodes furthest away from the winner undergo the least change.

2. Batch Training

The incremental process defined above can be replaced by a batch computation version which is significantly faster [44]. The batch training algorithm is also iterative, but instead of presenting a single data vector to the map at a time, the whole data set is given to the map before any adjustments are made. In each training step, the data set is partitioned such that each data vector belongs to the neighborhood set of the map unit to which it is closest, the Voronoi set [44]. The sum of the vectors in each Voronoi set are calculated as

$$\mathbf{s}_i(t) = \sum_{j=1}^{n_{Vi}} \mathbf{x}_j \tag{34}$$

where n_{Vi} is the number of samples in the Voronoi set of unit i. The new values of the weight vectors are then calculated as

$$\mathbf{w}_i(t+1) = \frac{\sum_{j=1}^{m} N_{ij}(t)\mathbf{s}_j(t)}{\sum_{j=1}^{m} n_{Vj} h_{ij}(t)} \tag{35}$$

where m is the number of map units and n_{Vj} is the number of samples falling into Voronoi set V_i.

3. Quality Measures

Although the issue of SOFM quality is not a simple one, two evaluation criteria, resolution and topology preservation, are commonly used [45]. One method to evaluate

the map resolution is to calculate the average quantization error, *Qerr*. This index is the average distance between each input data vector and its best matching unit (BMU) [46]. Topology preservation may be assessed by calculating the topological error, *Terr*, of the map. This value represents the percentage of data vectors for which its first BMU and second BMU are not adjacent units [47]. Both *Qerr* and *Terr* are used in the evaluation of the SOFM in this thesis.

B. Genetic Algorithms

Genetic algorithms (GAs) are a type of random search algorithm based on the mechanics of natural selection and natural genetics. GAs operate on pieces of information as nature operates on genes in the course of evolution. Individuals in the system are termed *strings*, and are analogous to chromosomes in biological systems. As chromosomes are composed of genes, a string in a GA is composed of *alleles*. These strings are represented by a linear sequence of letters of an alphabet, typically ones and zeros, and are allowed to mutate, crossover, and reproduce. All individuals of one generation, termed a *population*, are evaluated by a particular *fitness function* with respect to the given application domain. After a number of generations, the population consists of individuals that are well adapted in terms of the chosen fitness function [28].

Genetic algorithms have been proven to provide robust search in complex spaces [48]. Their ability to take advantage of accumulating information about an initially unknown search space to bias subsequent searches has demonstrated substantial improvement over a number of random and local search methods [28]. Since GAs are primarily a domain independent search technique, they are particularly suited for

applications where domain knowledge is difficult or impossible to provide, as in the case of feature selection for image segmentation.

1. Crossover, Mutation, and Reproduction

Although several variations of genetic algorithms exist [48], they typically share the common operations of crossover, mutation, and reproduction. In general, after a population of individuals has been initialized either randomly or according to a heuristic, the fitness of the individuals is calculated according to the given fitness function. Parents are then selected with a probability proportional to their fitness. The selected parents are mated, with possible mutation, to create the next generation. This process is repeated until an acceptable solution is found. The details explaining how genetic algorithms are successful in obtaining an optimal solution may be found in Appendix I.

The general principle for selecting parents is that of natural selection, with an individual's chance of contributing to the new generation's genetic pool being proportional to its fitness. Although there are several ways to implement the reproduction operator, one of the computationally simplest and most popular is the "Roulette wheel" method [48]. This method selects the strings in a statistical fashion, based solely upon their relative fitness values as a percentage of the total.

Once the parents have been selected, individuals for the next generation are formed using two main genetic operators, crossover and mutation. Crossover is performed in two steps. First, the selected parents are mated at random. Second, a random position in the gene structure is selected (the crossover site) for each pair of mates, and the remaining segments of the parents are swapped with crossover probability p_c. For example, if the crossover site is chosen to be between the third and fourth alleles,

parent #1	**0**	**1**	**1**	**0**	**0**	**1**	**0**
parent #2	**1**	**0**	**0**	**1**	**0**	**1**	**1**

generates

child #1	**0**	**1**	**1**	**1**	**0**	**1**	**1**
child #2	**1**	**0**	**0**	**0**	**0**	**1**	**0**

after crossover. Therefore, the features of two individuals are combined to create two similar offspring. If the GA decides not to perform crossover for a particular pair, the two selected strings are simply copied to the new population.

Mutation operates by exchanging a single allele. For each string element in each string in the mating pool, the GA determines whether or not it should perform mutation with mutation probability p_m If it does decide to perform mutation on an allele, it changes the element value to a new one, for example a one to a zero or vice versa. The mutation probability is typically a very small number, such as 0.001, to prevent fit strings from being destroyed. This operation is a way to insert new information into the population, preventing possible stagnation that might occur during the search process. In doing so, the GA is protected against the irrecoverable loss of good solution features and maintains diversity in the population.

There are various ways to implement generation replacement. In total generation replacement, every offspring becomes a part of the next generation, while every parent is discarded. Some variations of generation replacement compare the original parent individuals and the offspring simultaneously. For a population size n, only the best n of the $2n$ parents and children become part of the next generation. This method is called elitist generation replacement [49]. Although it guarantees that fit individuals are not lost from one generation to the next, it has been suggested that elitism improves local search at the expense of global perspective [48]. For this reason, total generation replacement

has been utilized in this thesis. Termination may be activated by finding an acceptable approximate solution or by observing no performance improvements over a certain number of generations.

2. Multiobjective Optimization

The fitness function discussed in the previous section assumed that there was a single objective for the genetic algorithm to optimize. In single objective optimization, the goal is to obtain the "best" design or decision, which is typically the global minimum or global maximum, depending on the nature of the problem. However, in many real-world design or decision making problems, the simultaneous optimization of several objectives is required.

When multiple objectives are to be optimized, a singular solution which is best with respect to all objectives may not exist. Instead, a set of solutions may be considered superior to the rest of the solutions in the search space considering all objectives, but which are inferior to other solutions in the space. These solutions are known as *Pareto-optimal* or *nondominated* solutions. The rest of the solutions are known as dominated solutions. Since none of the solutions in the nondominated set are absolutely better than any other, any one of them is considered acceptable. Mathematically, the problem can be written as follows [50],

$$\begin{aligned} &\text{Minimize/Maximize} && f_i(\mathbf{x}) && i = 1, 2, \ldots, N \\ &\text{Subject to} && g_j(\mathbf{x}) \le 0 && j = 1, 2, \ldots, J \\ & && h_k(\mathbf{x}) = 0 && k = 1, 2, \ldots, K \end{aligned} \tag{36}$$

The parameter $\mathbf{x}$ is an n dimensional vector have n design or decision variables. Using the following definitions [51],

1. *A vector* $\mathbf{u} = (u_1, \cdots, u_n)$ *is said to be* inferior *to* $\mathbf{v} = (v_1, \cdots, v_n)$ *iff* $\mathbf{v}$ *is partially less than* $\mathbf{u}$ ($\mathbf{v}\ p < \mathbf{u}$), *such that*, $\forall i = 1, \cdots, n, \quad v_i \leq u_i$,

2. *A vector* $\mathbf{u} = (u_1, \cdots, u_n)$ *is said to be* superior *to* $\mathbf{v} = (v_1, \cdots, v_n)$ *iff* $\mathbf{v}$ *is inferior to* $\mathbf{u}$, *and*

3. *Vectors* $\mathbf{u} = (u_1, \cdots, u_n)$ and $\mathbf{v} = (v_1, \cdots, v_n)$ *are said to be* non-inferior *to one another if* $\mathbf{v}$ *is neither inferior nor superior to* $\mathbf{u}$,

each element in the Pareto-optimal set constitutes a non-inferior solution to the multiobjective problem. To illustrate, consider two functions to be simultaneously minimized, $f_1(x)$ and $f_2(x)$. Suppose the solution set for these functions is as shown in Table II.

TABLE II
EXAMPLE OF TWO FUNCTIONS TO BE SIMULTANEOUSLY OPTIMIZED. EACH FUNCTION IS EQUALLY SIGNIFICANT WITH RESPECT TO THE FINAL SOLUTION.

Solution	$f_1(x)$	$f_2(x)$
A	0.9501	0.8381
B	0.2311	0.1096
C	0.6068	0.6813
D	0.4860	0.3795
E	0.8913	0.8318
F	0.7621	0.5028
G	0.4565	0.7095
H	0.1085	0.4289
I	0.8214	0.3046
J	0.4447	0.1897

The Pareto-optimal set includes {*B*, *H*}, as visualized in Figure 5, since they are both non-inferior solutions. While *B* performs better with respect to $f_2(x)$, *H* performs better with respect to $f_1(x)$; neither solution dominates the other. Although *J* is not inferior to *H*, it is inferior to *B*, and therefore is not included in the set.

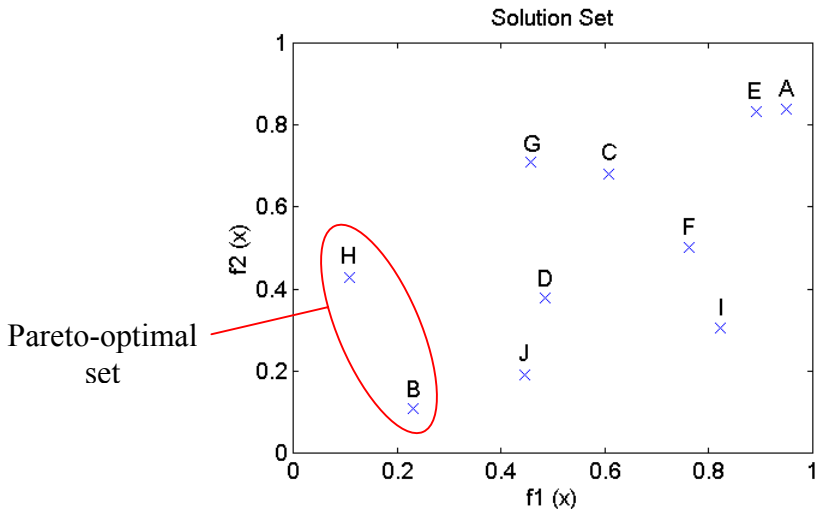


FIGURE 5 – Graphical Representation Of Multiobjective Optimization Problem. The Pareto-Optimal Set Is Circled In Red.

A number of methods have been developed to address the issue of multiobjective optimization, including the vector-valued genetic algorithm (VEGA), the nondominated sorting genetic algorithm (NSGA), and goal programming [50], [51]. One classical technique is to scalarize the vector of objectives into one objective by averaging them with a weight vector. This process is known as *objective weighting* and allows a simpler optimization algorithm to be used. Multiple objective functions are combined into one overall objective function, Z, as follows [50]:

$$Z = \sum_{i=1}^{N} w_i f_i(\mathbf{x}) \tag{37}$$

where $\mathbf{x} \in \mathbf{X}$, and $\mathbf{X}$ is the feasible region. The weights w_i are fractional numbers $0 \le w_i \le 1$, and all weights are summed to one,

$$\sum_{i=1}^{N} w_i = 1 . \tag{38}$$

With this method, the optimal solution is controlled by the weight vector **w** such that an objective may be favored by a simple weight adjustment.

A common problem with multiobjective optimization is the appearance of an *objective conflict* [50], where none of the feasible solutions allow simultaneous optimal solutions for all objectives. Thus a mathematically most favorable Pareto-optimum is that solution which offers least objective conflict. In addition to being computationally inexpensive, solutions obtained with equal weights in objective weighting are typically Pareto-optimum solutions with minimal objective conflict. For this reason, this is the multiobjective optimization routine employed in this thesis. The two objective functions are *Qerr* and *Terr*, given equal weighting between the two.

Genetic algorithms seek to maximize fitness values, but *Qerr* and *Terr* are defined as minimization problems. The following cost-to-fitness transformation is normally used [48] when this situation occurs,

$$f(x) = \begin{cases} C_{\max} - g(x) & \text{when } g(x) < C_{\max} \\ 0 & \text{otherwise} \end{cases} \tag{39}$$

where $g(x)$ is the original function and C_{max} may be taken as an input coefficient, as the largest *g* value observed, as the largest *g* value in the current population, or the largest of the last *k* generations [48]. In this work, C_{max} was taken as an input coefficient to facilitate the comparison of one generation with another across all modalities.

IV. IMPLEMENTATION

The proposed system is divided into three main procedures: (1) feature extraction, (2) feature selection, and (3) post-processing. An overview of the system is shown in Figure 6. All programming was done in the *Matlab* 5 computing environment by MathWorks, Inc.

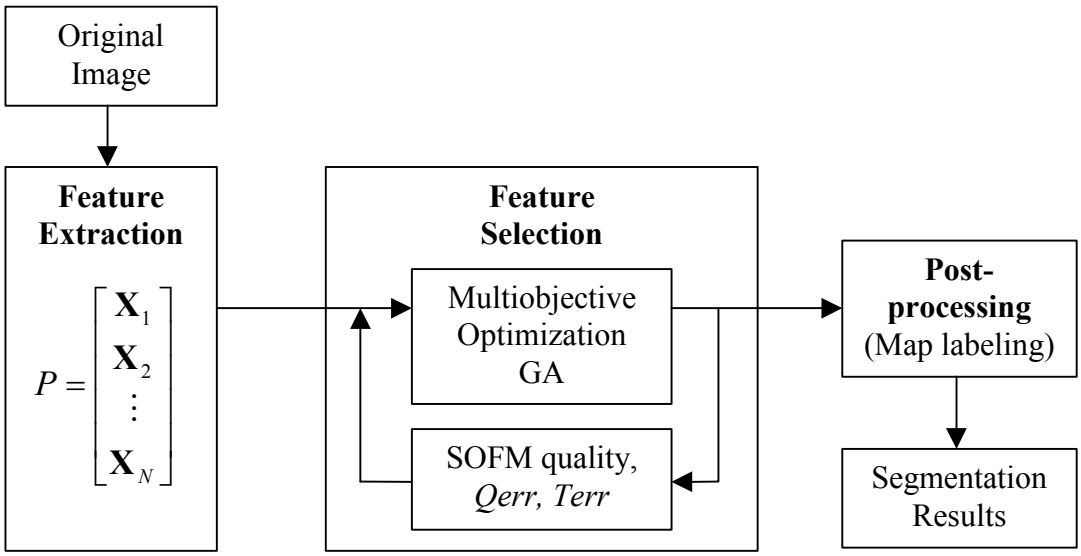


FIGURE 6 – System Overview. After The Features Are Extracted From The Original Images, The MOGA Searches For The Subset Of Features Which Result In The Best SOFM. The Best SOFM Is Then Labeled And Used To Segment Each Image.

A. System Design

In the feature extraction stage, an input image of dimension 256 x 256 is presented to the system. From this image, 23 features are calculated, forming the pattern matrix P. The dimensions of P are N x D, where $N = 65536$ is the number of pixels in the image and $D = 23$ is the number of features.

A random sample of 5% of the feature vectors, P', were then presented to the genetic algorithm. The reason for using a small percentage of the feature vectors as has also been done in [28], [52], [53] was not only to decrease the computational complexity, but it has been suggested that self-organizing feature maps are more effective in mapping a small amount of linearly independent data [53]. The validity of this method is discussed further in Section V.

It has been indicated that good GA performance requires the use of a moderate population size, a high crossover probability, and a low mutation probability [48]. For this reason, the following parameters have been used in this thesis,

1. *population sizes, Ps* = 30, 40, 50,
2. *crossover probability*, $p_c = 0.7$,
3. *mutation probability*, $p_m = 0.01$.

The features are represented as a binary string of length $N = 23$, where a 1 denotes the presence of a feature and a 0 denotes its absence. After a population has been randomly initialized, the first string extracts the appropriate features from the reduced pattern matrix and presents them to the self-organizing feature map. For example, if the first string in the population is

0 1 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1,

then features 2, 5, 6, 8, 20, and 23 of P' are the only ones used to train the SOFM.

The *SOM Toolbox for Matlab* [54] was used to generate the maps in this thesis. An implementation of the SOFM, the *Toolbox* has been widely used in many applications [44]-[46], [53]. The dimension of the maps are determined automatically. First, the number of map units are calculated according to the heuristic,

$$mapunits = 5 \cdot s^{0.54321} \tag{40}$$

where s is the number of samples. Then, the two largest eigenvalues of the training data are calculated, and the ratio of these eigenvalues define the ratio between the sidelengths of the map grid. The final values of the sidelengths are set such that their product most closely matches the desired number of map units [54]. The maps are initialized linearly and are trained using the batch algorithm described in Section III. After the SOFM has converged, the two quality measures, *Qerr* and *Terr*, are calculated and stored. Each string in the population undergoes this same procedure.

Once quality measures have been obtained for every string in the population, objective weighting in conjunction with the cost-to-fitness transformation is applied to calculate the fitness of each string as,

$$fitness = 1 - (0.5 \cdot Qerr + 0.5 \cdot Terr) \tag{41}$$

The Roulette wheel method is used to select the parents, which are then mated with crossover and mutation to produce the next generation. The process is repeated for each generation in turn.

A post-processing step to label the SOFM is necessary because of its unsupervised nature. The array neurons are labeled in response to a specific known vector from the training set [43]. In this thesis, the classification for each pixel is known in advance, so the labels assigned to each pixel are used to label the SOFM.

B. Brain Model

The simulation images used to test the system were contributed by M. N. Ahmed [33]. Thirty-five images consisting of ellipsoids of known dimension and gray levels were constructed to represent a 3D model of the head. The gray levels assigned to each

structure of the model are shown in Table III. An example of one of the images is shown in Figure 7.

TABLE III
PARAMETERS OF THE BRAIN MODEL, LISTING EACH STRUCTURE OF THE MODEL AND ITS CORRESPONDING GRAY LEVEL VALUE.

Notation	Structure	Gray level value
SKN	Skin	41
WM	White matter	81
GM	**Gray matter**	251
CC	Corpus callosum	101
LV	Left ventricle	32
RV	Right ventricle	31
LCN	Left caudate nucleus	132
RCN	Right caudate nucleus	131
LTN	Left thalamus nucleus	141
RTN	Right thalamus nucleus	143
FO	Fornix	201
BKG	Background	1

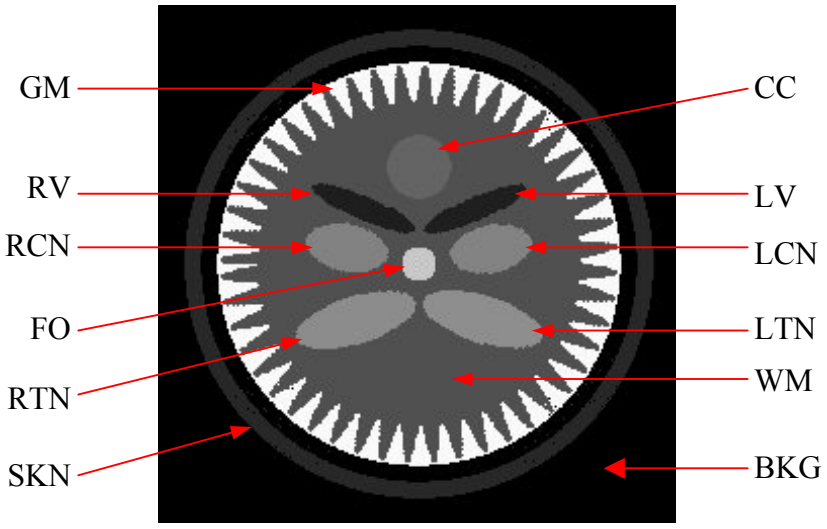


FIGURE 7 – Sample Image From The Brain Model With Pointers To Each Structure. The Model Follows The Convention Of MRI, Such That The Structure On The Right Side Of The Image Corresponds To The Structure On The Left Side Of The Subject.

C. MRI Data Set

Magnetic resonance imaging (MRI) uses magnetic energy and radio waves to create images ("slices") of the human body. During the examination, the MRI scanner creates a strong magnetic field, causing the nuclei within body tissues to align. The energy absorbed by the nuclei is then released, returning the nuclei to their initial state of equilibrium. It is this transmission of energy by the nuclei that is observed as the MRI signal.

Two physical processes return the nuclei to their initial state: the relaxation back to equilibrium of the component of the nuclear magnetization which is 1) parallel and 2) perpendicular to the magnetic field. The first process takes roughly T1 time, and the second takes T2 time. The density of protons in a tissue (PD), T1, and T2 determine the strength of the MRI signal [55].

For most soft tissues in the body, the proton density is very homogenous but may exhibit higher intensity for gray matter. T1 and T2 can be very different for different soft tissues, contributing to higher contrast in the images [55]. While T1 shows higher intensity for white matter, T2 presents higher intensity for cerebrospinal fluid.

The six MRI data sets used were obtained from *The Whole Brain Atlas* project [55] and the Computer Vision and Image Processing Laboratory at the University of Louisville [56]. Forty-four T1 and T2 images from a subject with healthy brain structures and from a subject with Alzheimer's disease constituted the four sets from [55]. The images from each subject have been registered such that the pixel at (x, y) in the T1 image represents the same structure as the pixel at (x, y) in the corresponding T2 image. An example of a T1 and T2 image of the same slice is shown in Figure 8. From

[56], 18 PD images from two subjects with multiple sclerosis comprised the remaining two sets. An example of a PD image from one subject is shown in Figure 9.

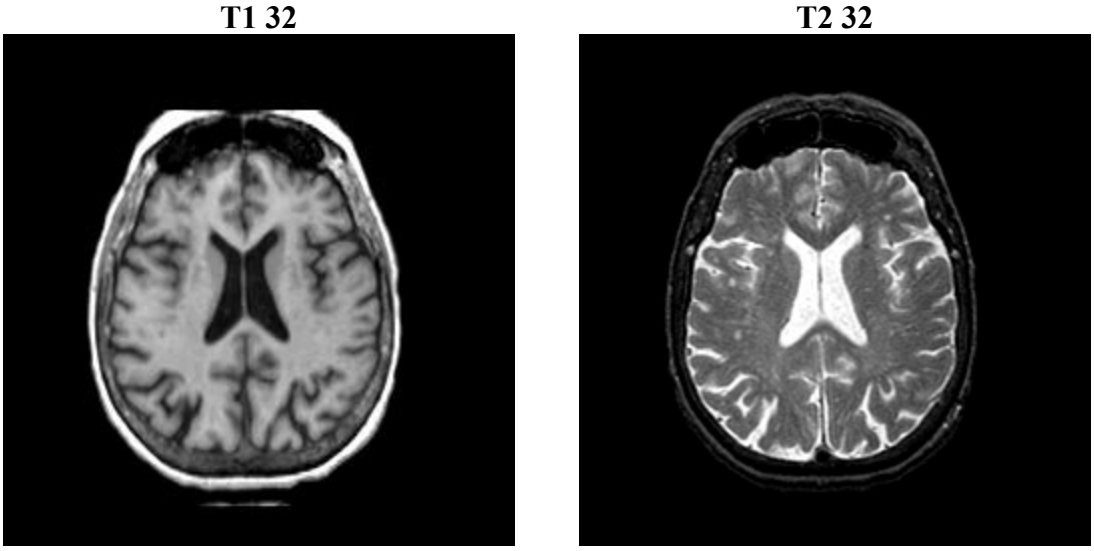


FIGURE 8 – Example Of T1 And T2 MR Images Of A Subject With Healthy Brain Structures. T1 Images Display Higher Intensity (Brightness) For White Matter, While T2 Images Show Higher Intensity For Cerebrospinal Fluid.

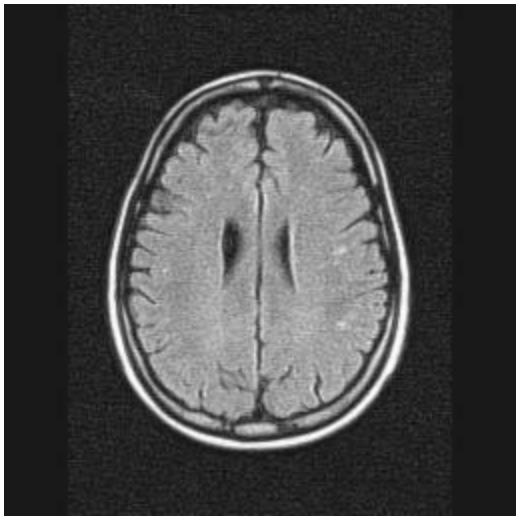


FIGURE 9 – Example Of A PD MR Image Of A Subject With Multiple Sclerosis. PD Images Are Often Lower In Contrast Than T1 Or T2 Images, Showing Slightly Higher Intensity For Gray Matter.

V. SEGMENTATION RESULTS

For each volume, population sizes of 30, 40, and 50 were tested over 30 generations, and the quality measures *Qerr* and *Terr*, the fitness values, and the generated strings were stored for each case. For all trials, only 5% of the feature vectors were used to train the maps. Varying percentages of the feature vectors in increments of 5%, up to 100%, were tried; in all cases, as the percentage of feature vectors used was increased, the quality of the SOFM decreased, which is consistent with the findings in [53]. Selecting the features randomly had a small impact on the segmentation accuracy when all other parameters were fixed. The worst case observed is shown in Table IV. For this reason, all percentages reported should be considered to have a tolerance of $\pm 0.5\%$.

TABLE IV
VARIATIONS IN SEGMENTATION ACCURACY FROM USING RANDOM SAMPLES. THE WORST DIFFERENCE OBSERVED WAS 0.45%.

	Segmentation Accuracy
Trial 1	94.38
Trial 2	94.22
Trial 3	94.27
Trial 4	94.51
Trial 5	94.67
Trial 6	94.64
Trial 7	94.54
Trial 8	94.67
Trial 9	94.45
Trial 10	94.37

A. Simulation Results

Although the whole volume was processed, the details from only six images are described to keep the documentation manageable, while illustrating the variations seen across the volume. As expected, the average fitness values did improve over time. Figure 10 shows the average fitness values for one trial of each population size on Model A. Typically, no additional improvement was observed after 18-26 generations, but the genetic algorithm was allowed to run for 30 generations to facilitate analysis.

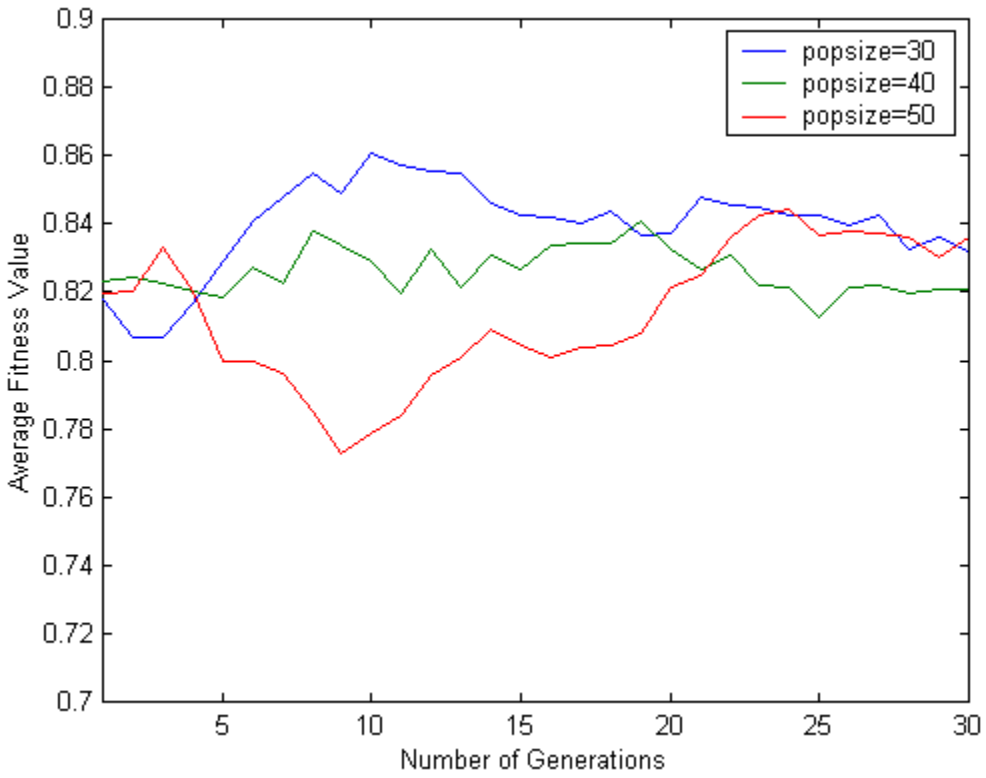


FIGURE 10 – Improvement In Average Fitness Values For Simulation Model. These Results Are Consistent With The Theory Presented In Appendix I.

The largest Pareto-optimal set observed for any one image contained four strings. Consequently, four strings from each image, consisting of the Pareto-optimal set and the next highest performers when needed, were selected to generate SOFMs for their

respective images. For each SOFM, the set of feature vectors *P* were assigned to the neuron which was its best matching unit. Each neuron was then labeled according to the majority of feature vectors with which it was associated. For example, if nine BKG feature vectors and one SKN feature vector were assigned to the same neuron, that neuron would take the label of BKG. An example of a U-matrix, which depicts the distances between neighboring model vectors as intensity values [57] and its corresponding labeled map are shown in Figure 11.

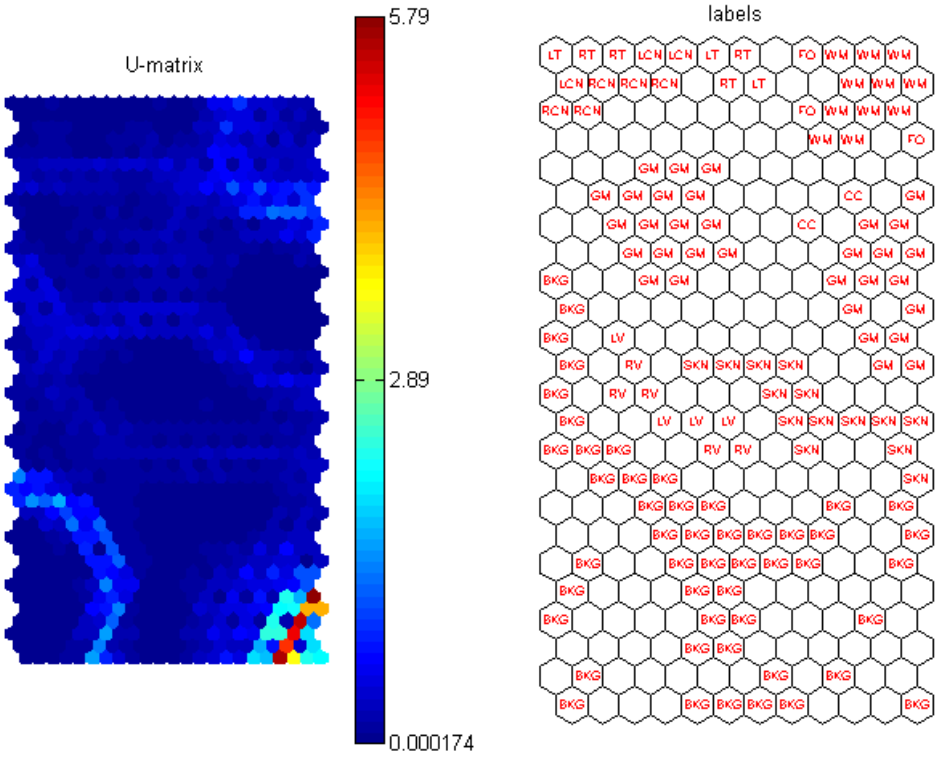


FIGURE 11 – Sample U-Matrix And Corresponding Labeled Map. The U-Matrix Depicts The Distances Between Neighboring Model Vectors As Intensity Values. The Labeled Map Shows Which Neurons Represent Which Structures.

The segmentation accuracy was defined as the percentage of feature vectors which were correctly classified, as shown in Table V, with the Pareto-optimal set for

Models A, C, E, and F highlighted in pale yellow. These models were included in Table V because they produced the strings which resulted in the highest segmentation accuracy.

TABLE V
A COMPARISON OF ERROR VALUES, FITNESS VALUES, AND SEGMENTATION ACCURACY FOR MOST FIT STRINGS ON SIMULATION MODELS A, C, E, AND F.

Features Represented in String	Qerr	Terr	**Fitness Value**	**Segmentation Accuracy**
Model A				
1, 5, 8, 18, 19	0.0330	0.0700	0.9485	94.46
2, 5, 6, 8, 20, 23	0.0580	0.0650	0.9385	85.27
5, 6, 8, 10, 19, 21, 22	0.0560	0.0950	0.9245	88.99
2, 5, 8, 18, 20, 21	0.0610	0.1370	0.9010	84.52
Model C				
2, 3, 4, 6, 18, 19, 20	0.0610	0.0570	0.9410	92.07
1, 4, 5, 6, 7, 10, 17, 19	0.0340	0.1110	0.9275	97.88
1, 6, 7, 10, 17, 19	0.0280	0.1590	0.9065	98.25
2, 3, 5, 6, 17, 23	0.0740	0.0590	0.9335	91.28
Model E				
1, 4, 5, 8, 19, 21	0.0420	0.0650	0.9465	96.71
1, 8, 18, 19, 21	0.0390	0.1020	0.9295	96.52
1, 4, 5, 10, 19, 21	0.0320	0.1240	0.9220	97.80
1, 4, 7, 8, 18, 19, 22	0.0290	0.1480	0.9115	96.05
Model F				
7, 11, 13, 17	0.0330	0.0760	0.9455	98.56
1, 4, 7, 8, 17, 18, 19, 21, 22	0.0820	0.0670	0.9255	99.20
1, 5, 12, 21	0.0290	0.2000	0.8855	99.23
1, 5, 7, 14, 17, 18, 20, 22	0.0540	0.0810	0.9325	98.42

Although for many of the trials, high fitness values did result in high classification rates, this was not always the case. For example, with Model A, fitness values as high as 0.9385 only resulted in 85.27% of the pixels being correctly classified. The image which resulted from this particular string is shown in Figure 12. Despite the fact that no direct correlation between SOFM quality and segmentation accuracy was found, 82% of the

strings for all Pareto-optimal sets did result in a segmentation accuracy between 94.01 – 99.91% for their respective images in the volume.

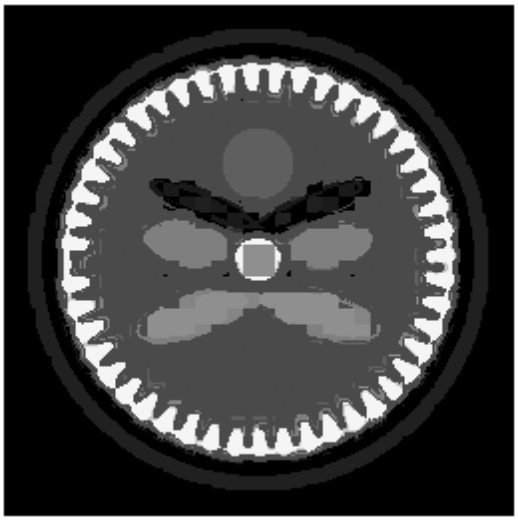


FIGURE 12 – Example Of Discrepancy Between High Fitness Value And Segmentation Accuracy Due To A Loss In Spatial Resolution.

To determine which string would result in the highest segmentation accuracy for the volume as a whole, the strings which resulted in the ten highest classification rates, were used to generate maps for each image in the volume. The segmentation results of each string on the volume are presented in Table VI, and showed string #5 to result in the highest segmentation accuracy.

The features included in this optimal subset are listed in Table VII. Using this subset of features, the segmentation accuracy for each structure in the brain model was calculated, as shown in Table VIII. The worst errors most often resulted from the SOFM incorrectly classifying a right-side structure as its corresponding left-side structure, and vice versa. This error typically occurred on the boundaries of one structure with another.

TABLE VI
SEGMENTATION ACCURACY OF TOP TEN MOST FIT STRINGS ON SIMULATION MODEL.

String Number	Features Represented in String	Simulation Model
1	1, 4, 7, 8, 17, 18, 19, 21, 22	97.65
2	1, 5, 12, 21	96.72
3	1, 5, 7, 14, 17, 18, 20, 22	96.82
4	1, 6, 7, 10, 17, 19	97.91
5	1, 4, 5, 6, 7, 10, 17, 19	98.17
6	1, 4, 5, 10, 19, 21	97.66
7	1, 5, 8, 18, 19	96.44
8	1, 8, 18, 19, 21	95.99
9	1, 4, 5, 8, 19, 21	95.43
10	1, 4, 7, 8, 18, 19, 22	97.13

TABLE VII
OPTIMAL FEATURE SUBSET FOR SIMULATION MODEL.

	Feature	Feature Class
x_1	Intensity	First-order
x_4	Mean	Texture
x_5	Contrast	Texture
x_6	Homogeneity	Texture
x_7	Entropy	Texture
x_{10}	$L_iL_{i,}$, $\sigma = 1.0$	Multiscale
x_{17}	ϕ_1	Moment Invariant
x_{19}	ϕ_3	Moment Invariant

An example of an image segmented using this feature subset is shown in Figure 13. The six original images under consideration and their corresponding segmented images are shown in Appendix II.

TABLE VIII
SEGMENTATION ACCURACY FOR EACH STRUCTURE OF
THE BRAIN MODEL, USING THE OPTIMAL FEATURE SUBSET.

Brain Structure	Simulation Model
BKG	99.41
CC	99.36
FO	96.70
WM	99.79
LCN	72.59
LT	81.81
LV	49.32
RCN	87.18
RT	91.59
RV	63.30
SKN	98.52
GM	98.99
Total	98.17

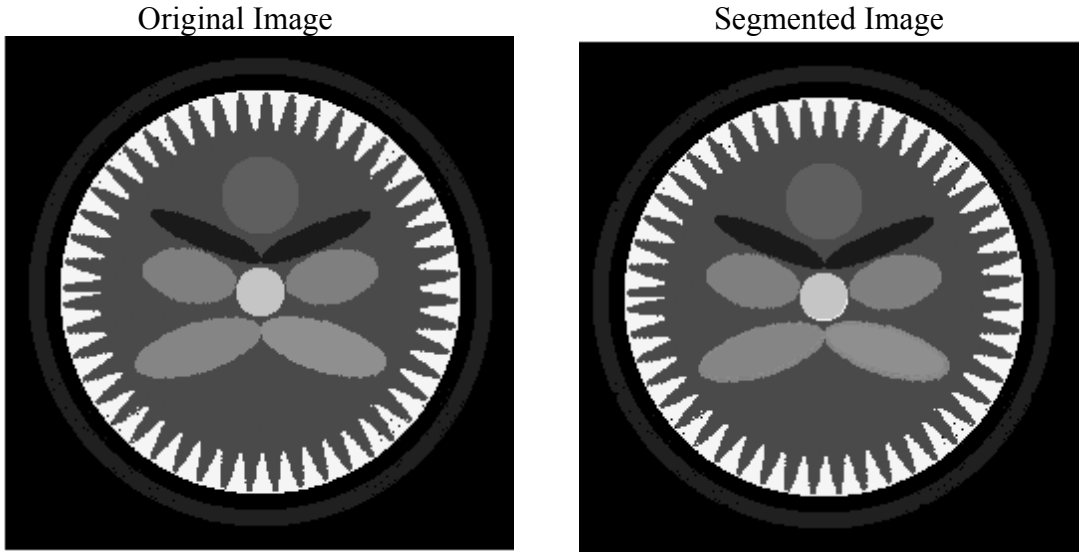


FIGURE 13 – Example Of Segmented Model Image, Compared To Original.

To test the robustness of the system, the images were degraded with Gaussian noise. The signal-to-ratio (SNR) values were taken to be 10, 20, and 30, as defined by,

$$SNR = 10 \cdot \log_{10}\left[\frac{\text{image variance}}{\text{noise variance}}\right]. \tag{42}$$

An example of the degraded images for Model A is shown in Figure 14. The segmentation accuracy for the degraded models using the optimal feature subset is presented in Table IX. The results for the volume were as anticipated; the system performed adequately for smaller additions of noise, but was relatively unacceptable for large additions.

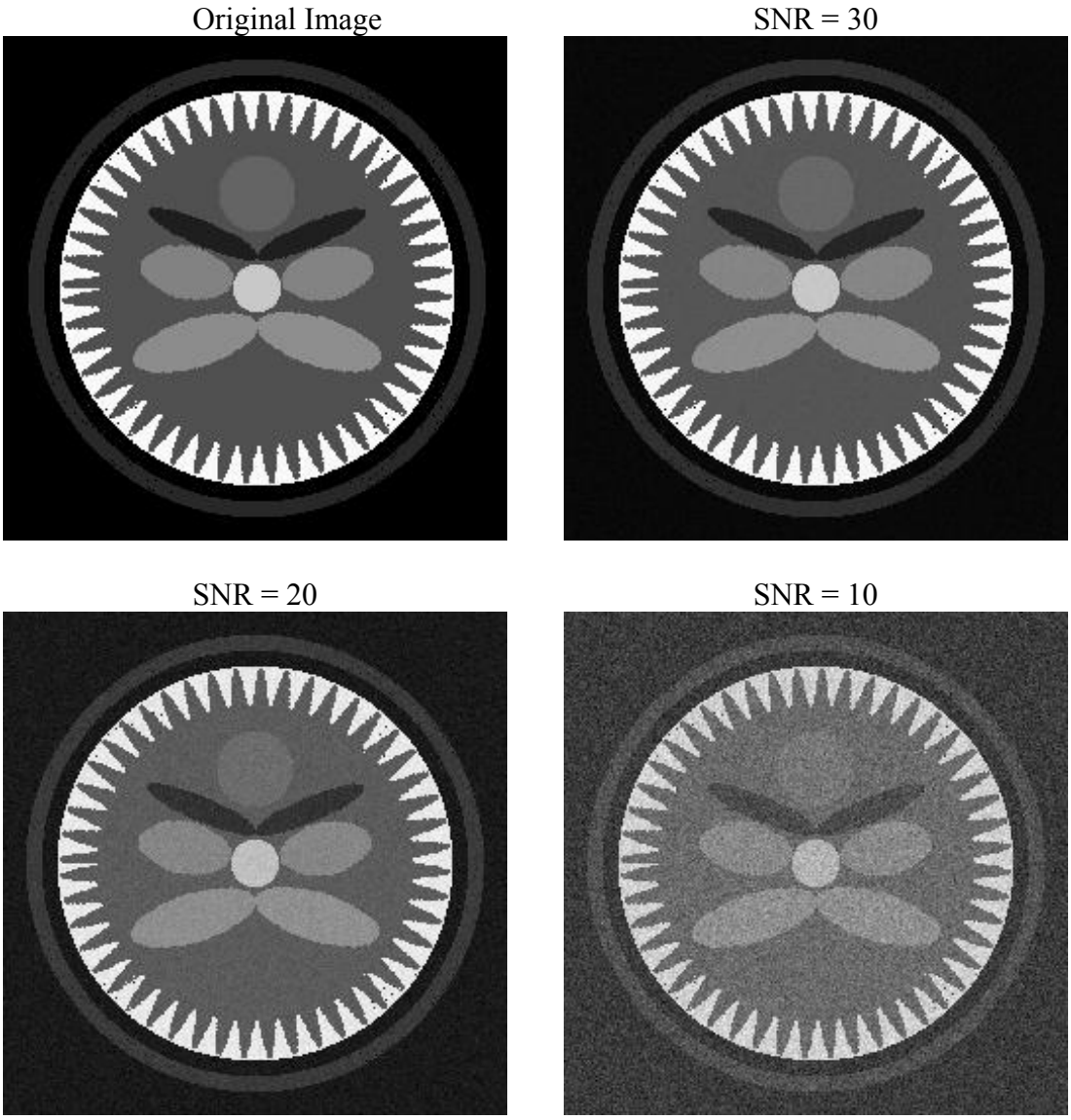


FIGURE 14 – Model Image Degraded With Increasing Levels Of Gaussian Noise.

TABLE IX
SEGMENTATION ACCURACY FOR SIMULATION MODEL
DEGRADED WITH GAUSSIAN NOISE.

Brain Structure	**Volume, Original Image**	**Volume,** SNR = 30	**Volume,** SNR = 20	**Volume,** SNR = 10
BKG	99.41	99.20	99.90	97.02
CC	99.36	99.90	74.15	22.70
FO	96.70	95.51	99.21	43.64
WM	99.79	99.57	99.62	96.17
LCN	72.59	43.80	55.53	58.58
LT	81.81	63.00	24.44	49.86
LV	49.32	40.07	22.54	92.71
RCN	87.18	58.27	75.04	26.62
RT	91.59	69.09	65.56	20.86
RV	63.30	68.35	22.29	45.77
SKN	98.52	94.31	94.87	83.29
GM	98.99	99.97	99.50	97.11
Total	98.17	96.59	93.73	87.58

B. Experimental Results

The algorithm was then tested on the six MRI data sets. As with the simulation model, the details of only six slices will be presented from each volume. The areas of interest were taken to be white matter (WM), gray matter (GM), and cerebrospinal fluid (CSF). First, the structures which did not fall into one of these three categories, such as bone and skin, were manually removed from the images. The images were then manually segmented according to their classification. An example of a T1 and T2 image cleared of the undesirable structures and the manual segmentation are shown in Figure 15. It should be noted that the lack of ground truth values means that the classification accuracy is determined by the manual segmentation alone.

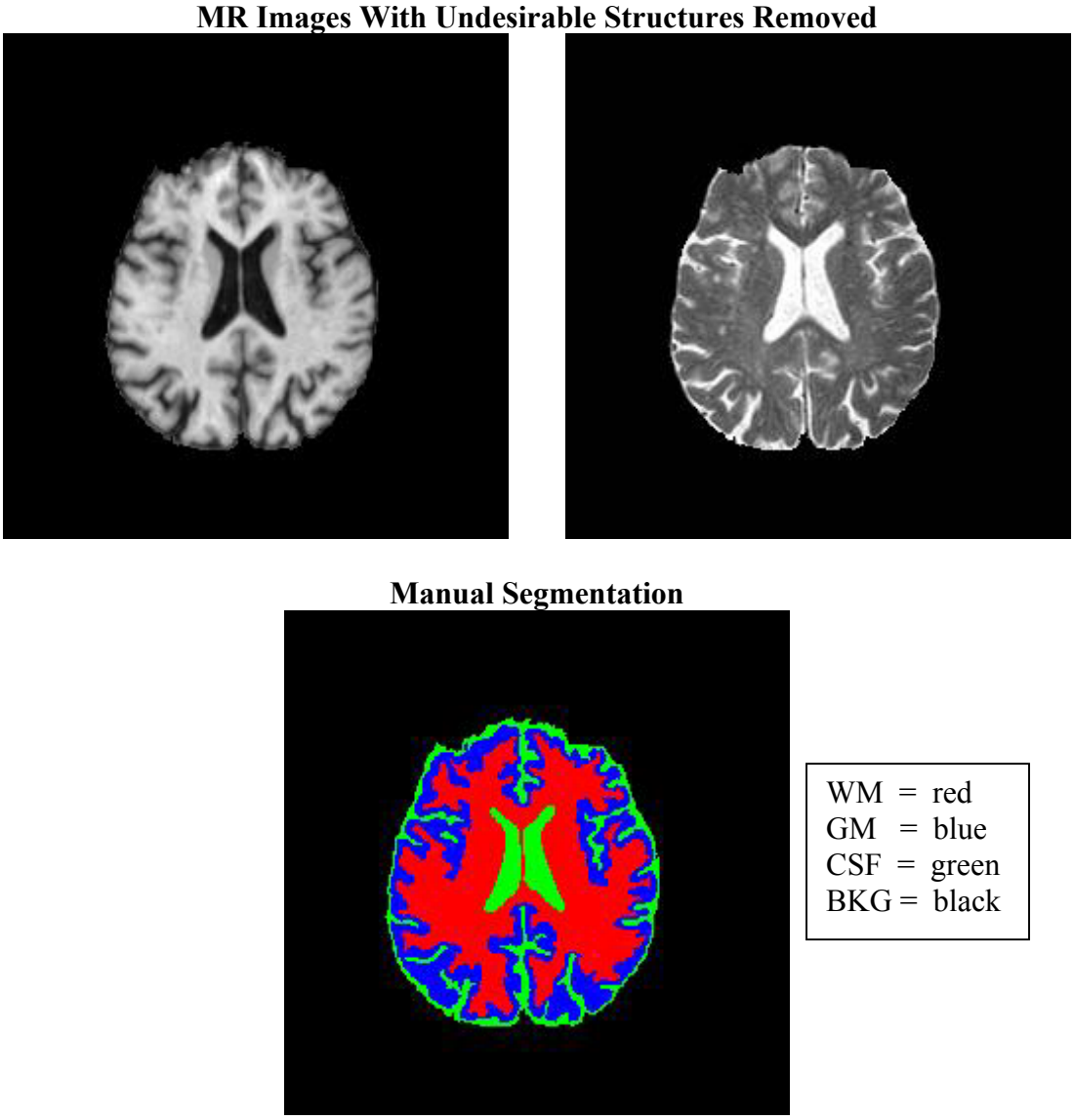


FIGURE 15 – Example Of Images Cleared Of Undesirable Structures And The Manual Segmentation Depicting White Matter, Gray Matter, Cerebrospinal Fluid, And Background.

As with the simulation model, the average fitness values of the genetic algorithm for the MRI data sets did improve over time, as seen by the example given in Figure 16. The GA did seem to converge more quickly for the MRI data sets, often within 10 – 20 generations; however, the GA was allowed for run for 30 generations as before.

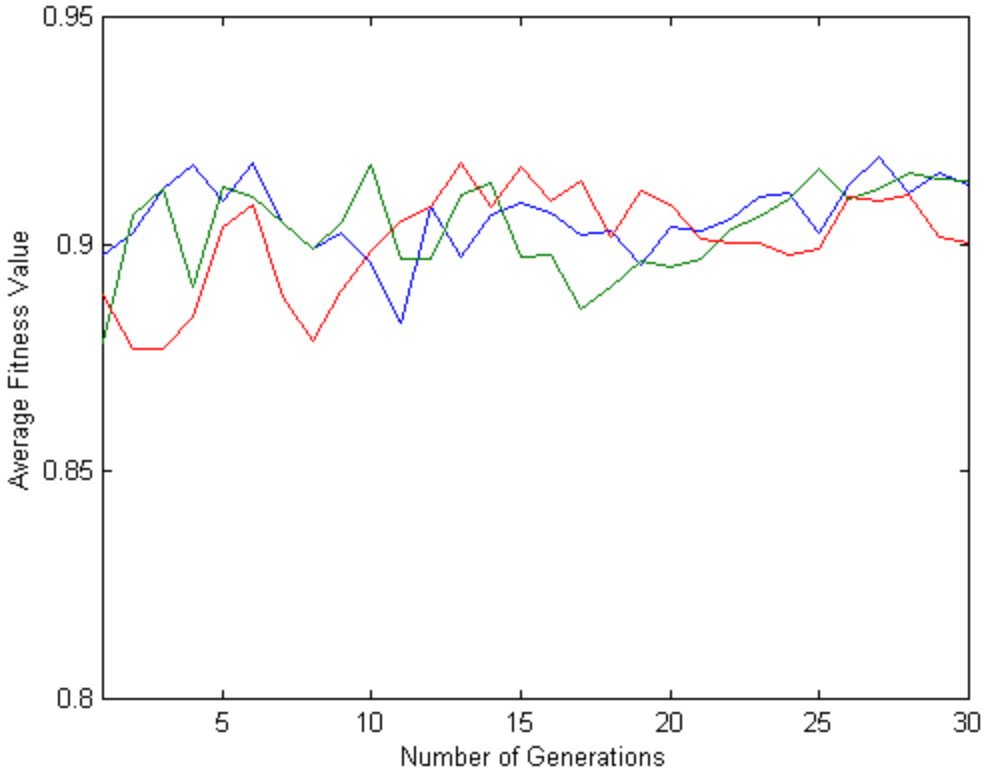


FIGURE 16 – Example Of Improvement In Average Fitness Values For MRI Data Set, Consistent With Theory Presented In Appendix I.

The procedure used to find the most fit strings for the simulation model was similarly performed on the MRI data sets. The segmentation accuracy for the T1, T2, and PD strings on their respective volumes is deferred to Appendix III. The features present in the most fit strings for each modality are listed in Table X. It is interesting to note that three of the features, x_1, x_3, and x_7, are present in the subsets for all three modalities. Furthermore, x_1 and x_7 were also present in the optimal subset found for the simulation model. In all cases, at least one feature from each of the four feature classes is present in each subset.

TABLE X
OPTIMAL FEATURE SUBSETS FOR T1, T2, AND PD MRI DATA SETS.

T1 Data Set		
	Feature	**Feature Class**
x_1	Intensity	First-order
x_2	Mean	First-order
x_3	Variance	First-order
x_5	Contrast	Texture
x_7	Entropy	Texture
x_8	Energy	Texture
x_9	$L_iL_{i,}$, $\sigma = 0.5$	Multiscale
x_{20}	ϕ_4	Moment Invariant
x_{21}	ϕ_5	Moment Invariant
T2 Data Set		
	Feature	**Feature Class**
x_1	Intensity	First-order
x_3	Variance	First-order
x_4	Mean	Texture
x_7	Entropy	Texture
x_{13}	$L_{ii,}$, $\sigma = 0.5$	Multiscale
x_{17}	ϕ_1	Moment Invariant
x_{22}	ϕ_6	Moment Invariant
x_{23}	ϕ_7	Moment Invariant
PD Data Set		
	Feature	**Feature Class**
x_1	Intensity	First-order
x_2	Mean	First-order
x_3	Variance	First-order
x_7	Entropy	Texture
x_8	Energy	Texture
x_9	$L_iL_{i,}$, $\sigma = 0.5$	Multiscale
x_{13}	$L_{ii,}$, $\sigma = 0.5$	Multiscale
x_{14}	$L_{ii,}$, $\sigma = 1.0$	Multiscale
x_{17}	ϕ_1	Moment Invariant
x_{21}	ϕ_6	Moment Invariant

Using each optimal feature subset, the segmentation accuracy for their respective data sets was calculated, and is shown in Table XI. For the T1 and T2 data sets, Set #1 refers to the subject with healthy brain structures, and Set #2 refers to the subject with Alzheimer's disease. Each of the PD data sets were obtained from subjects with multiple sclerosis. An example of the segmentation of a T1 image is shown in Figure 17, while the segmentation for six images from each of the six volumes are displayed in Appendix IV.

TABLE XI
SEGMENTATION ACCURACY FOR THE T1, T2, AND PD MRI DATA SETS.

	T1 Data Sets		**T2 Data Sets**		**PD Data Sets**	
Tissue	**Set #1**	**Set #2**	**Set #1**	**Set #2**	**Set #1**	**Set #2**
BKG	99.61	99.50	99.58	99.40	99.46	99.50
GM	84.51	80.81	58.42	57.00	81.70	68.36
CSF	77.53	74.59	73.22	53.01	80.43	81.42
WM	92.08	91.54	78.56	82.44	84.99	83.91
Total	95.94	95.31	91.72	90.54	95.51	95.36

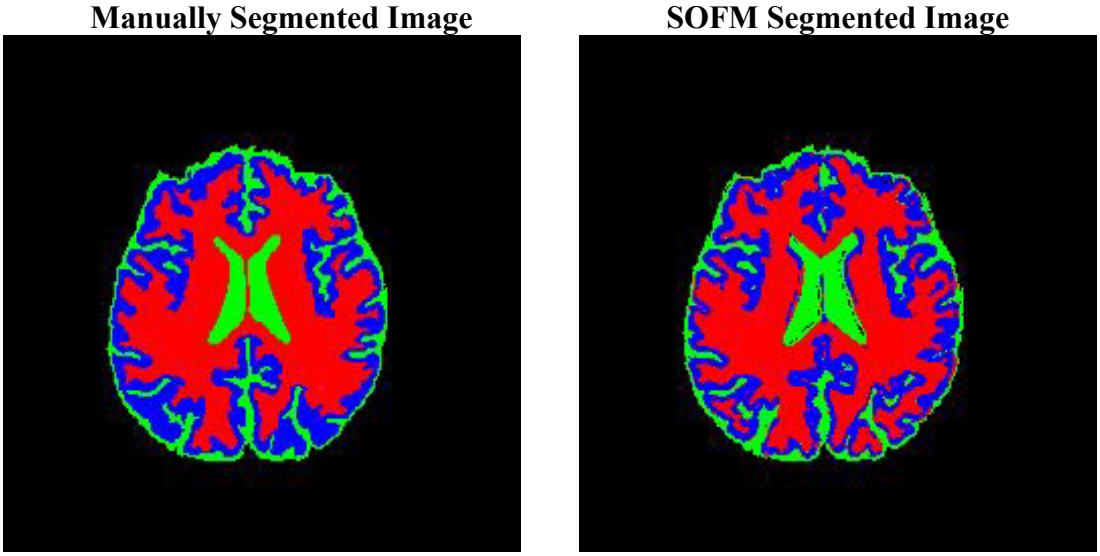


FIGURE 17 – Example Of A Manually Segmented T1 Image And Its Corresponding SOFM Segmentation Using Its Optimal Feature Subset.

In terms of overall performance, the T1 and PD data sets resulted in comparably acceptable classification rates. In considering the regions of interest, the T1 data sets were better able to classify white matter and gray matter, while the PD data sets were more suitable for correctly classifying cerebrospinal fluid. The results for the T2 data sets were less satisfactory, and were outperformed by the T1 and PD data sets in all cases. Considering that T2 images show a high contrast between cerebrospinal fluid and white and gray matter, the segmentation results for the T2 data set were somewhat disappointing. Taking into account the fact that cerebrospinal fluid, by nature, often occupies such a curvilinear and narrow space, these results are not entirely unexpected. This fact would also explain why the white matter was more accurately classified than the gray matter for all three modalities.

C. Comparison with Principal Component Analysis

To assess the quality of the feature selection algorithm implemented in this thesis, it was compared against using principal component analysis to reduce the feature set. Principal component analysis (PCA) is one of the most well known techniques to reduce the dimensionality of a data set [58]. This reduction is achieved by transforming the original data set to its principal components, ordered such that the first few variables retain most of the variation present in all of the original variables.

Assume that the input vectors are as given previously,

$$\mathbf{X} = \left[x_1, x_2, \cdots x_D\right]. \tag{43}$$

The input vectors are transformed using the relation,

$$\mathbf{Y} = A\mathbf{X} \tag{44}$$

where A is an m x d transformation matrix, $m \leq d$. The rows of A correspond to the m largest eigenvalues of the sample autocovariance matrix,

$$\Sigma = \left\langle \mathbf{X}\mathbf{X}^T \right\rangle \tag{45}$$

where $\langle\ \rangle$ represents expectation.

To compare the PCA performance with the results achieved by the genetic algorithm, the simulation model data set was reduced with m = 4, 5, ... 9. This particular range was selected based on the number of features represented in each string of Table VI to compare the two algorithms objectively. The results for the original model is shown in Table XII, and the results for the model degraded with Gaussian noise is given in Appendix V. For the original model, the genetic algorithm outperformed PCA for nearly every structure. For the models degraded with Gaussian noise, however, the PCA was more uniform in segmentation accuracy than the GA.

TABLE XII
PCA SEGMENTATION ACCURACY FOR SIMULATION MODEL.

Original Model		**Number of Principal Components**					
Brain Structure	**GA**	**4**	**5**	**6**	**7**	**8**	**9**
BKG	99.41	96.58	98.49	98.03	98.79	98.54	98.53
CC	99.36	96.86	95.41	96.48	98.66	97.48	97.82
FO	96.70	53.99	54.05	55.06	48.89	54.79	95.03
WM	99.79	95.64	94.07	95.38	97.25	96.62	97.84
LCN	72.59	55.28	20.81	34.72	67.46	0.04	21.20
LT	81.81	75.95	13.91	2.98	27.02	69.32	39.53
LV	49.32	52.08	45.67	26.31	43.87	16.43	70.32
RCN	87.18	22.03	59.70	47.36	14.14	29.28	68.00
RT	91.59	47.75	78.25	75.49	57.92	12.78	59.38
RV	63.30	41.33	35.27	59.68	52.81	64.41	12.30
SKN	98.52	96.24	95.95	95.64	94.36	94.55	96.88
GM	98.99	91.46	94.99	93.68	90.92	95.14	94.88
Total	98.17	91.43	91.37	90.97	91.85	91.40	93.76

Principal component analysis was then applied to the MRI data sets. As before, the number of principal components was selected according to the length of the strings in Appendix III. The segmentation accuracy acquired using this method for the T1, T2 and PD MRI data sets, respectively, are tabulated in Appendix VI.

For the T1 data sets, the GA outperformed PCA with the exception of cerebrospinal fluid being more accurately classified by using either eight or nine principal components. The results for the T2 data sets were more diverse. In all cases, the GA was better able to distinguish white matter, but PCA was clearly superior at detecting gray matter. The results for cerebrospinal fluid were more varied; however, the greatest accuracy was obtained using either five or eight principal components. Although the GA performed better in terms of overall segmentation accuracy than PCA for the PD data sets, the results for the individual tissues were less conclusive. In classifying gray matter, often when one method was superior in the first data set, it was inferior in the second data set. A similar effect was observed for cerebrospinal fluid, although the GA appeared to perform slightly better. With respect to white matter, PCA was the superior method, in particular when using nine principal components. These comparisons are summarized in Table XIII.

TABLE XIII
SUMMARY OF WHICH METHOD PERFORMS THE BEST FOR EACH TISSUE TYPE FOR T1, T2, AND PD IMAGES.

Tissue	T1	T2	PD
GM	GA	PCA, 4	Inconclusive
CSF	PCA, 9	PCA, 5	GA
WM	GA	GA	PCA, 9
Overall	GA	PCA, 9	GA

All tests were performed on a PC with an Intel® Celeron™ processor. The average run time for each procedure is shown in Table XIV.

TABLE XIV
PROCESSING TIMES FOR EACH PROCEDURE.

Procedure	Processing Time (sec)
Calculating the feature pattern	110.43
GA, population size = 30, one generation	81.02
GA, population size = 40, one generation	101.25
GA, population size = 50, one generation	120.09
PCA	25.61
Labeling the SOFM	19.56
Calculating segmentation accuracy	27.22

VI. CONCLUSIONS AND RECOMMENDATIONS

The segmentation algorithm did successfully find an optimal feature subset which resulted in a high segmentation accuracy for both the simulation model, the T1 MRI data sets, and the PD MRI data sets. The results for the T2 MRI data sets were only marginally acceptable with respect to correctly distinguishing white matter, gray matter, and cerebrospinal fluid.

In spite of the fact that a large number of strings generated by the genetic algorithm based on self-organizing feature map quality did result in high classification rates, no direct correlation between map quality and segmentation accuracy was found. This fact was most clearly demonstrated by Figure 12 and Table V. This appeared to be largely due to the loss of spatial resolution of the texture measures and the multiscale features. One possibility for increasing the spatial resolution of these features would be to calculate the features in a window surrounding each pixel independently, although this would significantly increase the computation time in the feature extraction stage.

It is important to note that relying upon the manual segmentation to determine classification accuracy does leave room for error. It is possible that the algorithm correctly distinguished one tissue type from another, but was classified incorrectly if the manual segmentation was inaccurate.

In all cases, the T1 and PD data sets were more useful in distinguishing tissue types than the T2 data sets. Even noting the greater ability of principal component analysis over the genetic algorithm at detecting gray matter, and in some cases

cerebrospinal fluid, for the T2 data set, the segmentation accuracy achieved for the T1 and PD data sets was superior in every instance. For the T1 data set, using the genetic algorithm to obtain an optimal feature subset did prove to result in higher classification rates than using principal component analysis, pointing to the validity of this method.

It should also be noted that this algorithm would be inappropriate for any type of real-time processing application. Depending on the size of the population used, each run of the GA typically took between 30-50 minutes, and the trained SOFMs required approximately 27.22 seconds to segment and label each image. However, for the purpose of diagnosing the presence of Alzheimer's disease before the onset of physical symptoms or to study the cause of the degeneration of the brain, these run times might be adequate.

REFERENCES

[1] W. M. Wells III, W. E. L. Grimson, R. Kikinis, and F. A. Jolesz, "Adaptive segmentation of MRI data," *IEEE Transactions on Medical Imaging*, vol. 15, no. 4, pp. 429-442, Aug. 1996.

[2] Y. Wang and T. Adali, "Quantification and segmentation of brain tissues from MR images: A probabilistic neural network approach," *IEEE Transactions on Image Processing*, vol. 7, no. 8, pp. 1165-1180, 1998.

[3] J. G. Webster, Ed., *Medical Instrumentation: Application and Design*. New York: John Wiley & Sons, Inc., 1998, pp. 551-555.

[4] A. D. Kulkarni, *Artificial Neural Networks for Image Understanding*. New York: Van Nostrand Reinhold, 1994.

[5] T. Kapur, W. E. L. Grimson, W. M. Wells III, and R. Kikinis, "Segmentation of brain tissue from magnetic resonance images," *Medical Image Analysis*, vol. 1, no. 2, pp. 109-127, 1996.

[6] M. Özkan, B. M. Dawant, and R. J. Maciunas, "Neural-network-based segmentation of multi-modal medical images: A comparative and prospective study," *IEEE Transactions on Medical Imaging*, vol. 12, no. 3, pp. 534-544, Sept. 1993.

[7] M. Kamber, R. Shinghal, D. L. Collins, G. S. Francis, and A. C. Evans, "Model-based 3-D segmentation of multiple sclerosis lesions in magnetic resonance brain images," *IEEE Transactions on Medical Imaging*, vol. 14, no. 3, pp. 442-453, Sept. 1995.

[8] D. Horn, E. Ruppin, M. Usher, and M. Herrmann, "Neural network modeling of memory deterioration in Alzheimer's disease," *Neural Computation*, vol. 5, pp. 736-749, 1993.

[9] D. T. Villareal and J. C. Morris, "The diagnosis of Alzheimer's disease," *Alzheimer's Disease Review*, vol. 3, pp. 142-152, 1998.

[10] J. Martin, A. Pentland, S. Sclaroff, and R. Kikinis, "Characterization of neuropathological shape deformations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 2, pp. 97-112, Feb. 1998.

[11] C. Lürig, R. Grosso, and T. Ertl, "Combining wavelet transform and graph theory for feature extraction and visualization," in *Proc. Eighth Eurographics Workshop on Visualization in Scientific Computing*, 1997, pp. 137-144.

[12] W. M. Wells III, P. Viola, H. Atsumi, S. Nakajima, and R. Kikinis, "Multi-modal volume registration by maximization of mutual information," *Medical Image Analysis*, vol. 1, no. 1, pp. 35-51, 1996.

[13] J. C. Rajapakse, J. N. Giedd, and J. L. Rapoport, "Statistical approach to segmentation of single-channel cerebral MR images," *IEEE Transactions on Medical Imaging*, vol. 16, no. 2, pp. 176-186, April 1997.

[14] D. N. Kennedy, P. A. Filipek, and V. S. Caviness, Jr., "Anatomic segmentation and volumetric calculations in nuclear magnetic resonance imaging," *IEEE Transactions on Medical Imaging*, vol. 8, no. 1, pp. 1-7, 1989.

[15] S.C. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 9, pp. 884-900, Sept. 1996.

[16] K. O. Lim and A. Pfefferbaum, "Segmentation of MR brain images into cerebrospinal fluid spaces, white and gray matter," *Journal of Computer Assisted Tomography*, vol. 13, no. 4, pp. 588-593, July/Aug. 1989.

[17] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, 1973.

[18] E. E. Kerre and M. Nachtegael, Eds., *Fuzzy Techniques in Image Processing*. Physica-Verlag, Heidelberg, 2000.

[19] C. Li, D. B. Goldgof, and L. O. Hall, "Knowledge-based classification and tissue labeling of MR images of human brain," *IEEE Transactions on Medical Imaging*, vol. 12, no. 4, Dec. 1993.

[20] N. A. Mohamed, M. N. Ahmed and A. A. Farag, "Modified fuzzy c-mean in medical image segmentation," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 3429-3432.

[21] W. C. Lin, E. C. K. Tsao, and C. T. Chen, "Constraint satisfaction neural networks for image segmentation," *Pattern Recognition*, vol. 25, no. 7, pp. 679-693, 1992.

[22] S. C. Amartur, D. Piraino, and Y. Takefuji, "Optimization neural networks for the segmentation of magnetic resonance images," *IEEE Transactions on Medical Imaging*, vol. 11, no. 2, June 1992.

[23] X. Yao and Y. Liu, "A new evolutionary system for evolving artificial neural networks," *IEEE Transactions on Neural Networks*, vol. 8, no. 3, pp. 694-713, May 1997.

[24] M. N. Ahmed and A. A. Farag, "Two-stage neural network for volume segmentation of medical images," *Pattern Recognition Letters*, vol. 18, pp. 1143-1151, 1997.

[25] S. Haring, M. A. Viergever, and J. N. Kok, "Kohonen networks for multiscale image segmentation," *Image and Vision Computing*, vol. 6, no. 12, pp. 339-344, 1994.

[26] M. Kudo and J. Sklansky, "Comparison of algorithms that select features for pattern classifiers," *Pattern Recognition*, vol. 33, pp. 25-41, 2000.

[27] H. Vafaie and K. De Jong, "Genetic algorithms as a tool for feature selection in machine learning", in *Proceedings of the Fourth International Conference on Tools with Artificial Intelligence*, 1992, pp. 200-204.

[28] H. Vafaie and K. De Jong, "Robust feature selection algorithms," in *Proceedings of the Fifth Conference on Tools for Artificial Intelligence*, 1993, pp. 356-363.

[29] R. Setiono, and H. Liu, "Neural-network feature selector," *IEEE Transactions on Neural Networks*, vol. 8, no. 3, pp. 654-659, 1997.

[30] S. M. Yamany, K. Khiani, and A. A. Farag, "Application of neural networks and genetic algorithms in the classification of endothelial cells," *Pattern Recognition Letters*, vol. 18, pp. 1205-1210, Nov. 1997.

[31] D. Ruck, S. Rogers, and M. Kabrisky, "Feature selection using a multilayer perceptron," *Neural Network Computation*, vol. 20, pp. 40-48, 1990.

[32] J. Yang and V. Honavar, "Feature subset selection using a genetic algorithm," *IEEE Transactions on Intelligent Systems*, vol. 13, no. 2, pp. 44-49, 1998.

[33] M. N. Ahmed, "Novel image segmentation and registration algorithms for the study of brain structure and function," Ph.D. dissertation, University of Louisville, Louisville, KY, USA, 1997.

[34] J. C. Russ, *The Image Processing Handbook*, Third Ed. North Carolina: CRC Press, 1999.

[35] J. R. Parker, *Algorithms for Image Processing and Computer Vision*. New York: John Wiley & Sons, Inc., 1997.

[36] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, Nov. 1973.

[37] J. Portilla and E. P. Simoncelli, "Texture representation and synthesis using correlation of complex wavelet coefficient magnitudes", *Prof. IEEE workshop on Statistical and Computational Theories of Vision*, 1999.

[38] Ch. Bouman and B. Liu, "Multiple resolution segmentation of textured images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 2, pp. 99-113, 1991.

[39] A. P. Witkin, "Scale-space filtering," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 1983, pp. 1019-1022.

[40] J. J. Koenderink, "The structure of images," *Biological Cybernetics*, vol. 50, no. 5, pp. 363-370, 1984.

[41] L. M. J. Florack, B. M. ter Haar Romeny, J. J. Koenderink, and M. A. Viergever, "Scale and the differential structure of images", *Image and Vision Computing*, vol. 10, pp. 376-388, July/Aug. 1992.

[42] J. B. A. Maintz, P. A. van den Elsen, and M. A. Viergever, "Comparison of edge-based and ridge-based registration of CT and MR brain images," *Medical Image Analysis*, vol. 1, no. 2, pp. 151-161, 1996.

[43] J. Zurada, *Introduction to Artificial Neural Systems*. Minnesota: West Publishing Company, 1992.

[44] T. Kohonen, S. Kaski, K. Lagus, J. Salojarvi, J. Honkela, V. Paatero, and A. Saarela, "Self organization of a massive document collection," *IEEE Transactions on Neural Networks*, vol. 11, no. 3, pp. 574-585, May 2000.

[45] E. Alhoniemi, J. Hollmn, O. Simula, and J. Vesanto, "Process monitoring and modeling using the self-organizing map," *Integrated Computer Aided Engineering*, vol. 6, no. 1, pp. 3-14, 1999.

[46] S. Kaski and K. Lagus, "Comparing self-organizing maps," in *Proceedings of International Conference on Neural Networks*, 1997, pp. 809-814.

[47] H.-U. Bauer and K. Pawelzik, "Quantifying the neighborhood preservation of self-organizing feature maps", *IEEE Transactions on Neural Networks*, vol. 3, no. 4, pp. 570-579, 1992.

[48] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Massachusetts: Addison-Wesley Publishing Co., Inc., 1989.

[49] J. Hallinan and P. Jackway, "Simultaneous evolution of feature subset and neural classifier on high-dimensional data," in *Conference on Digital Image Computing and Applications*, 1999.

[50] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms", *Evolutionary Computation*, vol. 2, no. 3, pp. 221-248, 1995.

[51] C. M. Fonseca and P. J. Fleming, P. J., "Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization." in *Proceedings of the Fifth International Conference on Genetic Algorithms*, 1993, pp. 416-423.

[52] N. W. Campbell, B. T. Thomas and T. Troscianko, "Segmentation of natural images using self-organizing feature maps", in *British Machine Vision Conference*, pp. 223-232, 1996.

[53] A. Zizzari, U. Seiffert, B. Michaelis, G. Gademann, and S. Swiderski, "Detection of tumor in digital images of the brain," in *Proceedings of the International Conference on Signal Processing, Pattern Recognition and Applications*, 2001, pp. 132-137.

[54] E. Alhoniemi, J. Himberg, J. Parhankangas, and J. Vesanta, "SOM Toolbox for Matlab", 1997. [Online]. Available: http://www.cis.hut.fi/projects/somtoolbox/

[55] K. A. Johnson and J. A. Becker, "The Whole Brain Atlas," 1999. [Online]. Available: http://www.med.harvard.edu/AANLIB/home.html

[56] J. Nett, R. Falk, A. A. Farag, and S. Ildstad, "Volume Registration", 2001. [Online]. Available: http://www.cvip.uofl.edu/mi/MS_MRI/registration/

[57] S. Kaski, J. Nikkila, and T. Kohonen, "Methods for interpreting a self-organized map in data analysis," in *Proceedings of the Sixth European Symposium on Artificial Neural Networks,* 1998.

[58] T. Jolliffe, *Principal Component Analysis*. New York: Springer-Verlag, 1986.

APPENDIX I

The search heuristics of a genetic algorithm are based upon Holland's *Schema Theorem* [48]. Without loss of generality, only the binary representation will be considered here. A schema H is defined as a template for describing a subset of strings with similar sections. The template consists of 0's, 1's, and "don't care" symbols, #'s. For example, the schema 0#0 matches two strings, 000 and 010. Thus, the total number of schema present in a binary string of length L is 3^L. The order of a schema $O(H)$ is equal to the number of fixed positions (the number of 0's and 1's) in the schema. The defining length $d(H)$ is the distance between the first and last specific string position. Considering the schema 1#00#0#,

$$O(1\#00\#0\#) = 4 \text{, and} \tag{46}$$

$$d(1\#00\#0\#) = 5 \tag{47}$$

(the first specific string position is 1, the last specific string position is 6, so $d(H) = 6 - 1 = 5$).

Assume at time t there are m instances of a schema H in the population $A(t)$; these m instances are denoted $m(H,t)$. If a string A_i is selected with probability,

$$p_i = \frac{f_i}{\sum f_i} \tag{48}$$

during reproduction, then for a nonoverlapping population of size n, where $f(H)$ is the average fitness of the strings representing schema H at time t, the number of representatives of the schema H expected at time $t + 1$ is,

$$m(H,t+1) = m(H,t) \cdot n \cdot \frac{f(H)}{\sum f_i}. \tag{49}$$

Since the average fitness of the population $A(t)$ may be denoted,

$$\bar{f} = \frac{\sum f_i}{n} \tag{50}$$

then,

$$m(H,t+1) = m(H,t)\frac{f(H)}{\bar{f}}. \tag{51}$$

Thus, schemata with fitness values higher than the population average will receive an increasing number of samples in the next generation [48].

A schema H survives crossover with probability,

$$p_{sc} = 1 - \frac{d(H)}{(L-1)} \tag{52}$$

since the schema is likely to be disrupted whenever the crossover site falls within the defining length from the $L - 1$ possible choices. Thus, if crossover is performed with probability p_c, then the probability of survival may be written as,

$$p_{sc} \geq 1 - p_c \frac{d(H)}{(l-1)} \ . \tag{53}$$

The schema survival probability when considering mutation with probability p_m , is

$$p_{sm} = (1 - p_m)^{O(H)}. \tag{54}$$

For values of $p_m << 1$, the survival probability may be approximated [48] by

$$p_{sm} = 1 - O(H) \cdot p_m \tag{55}$$

With the combined effect of reproduction, crossover, and mutation, a schema H is expected to receive

$$m(H,t+1) \geq m(H,t) \cdot \frac{f(H)}{\bar{f}} \cdot \left[1 - p_c \frac{d(H)}{(L-1)} - O(H) \cdot p_m \right] \tag{56}$$

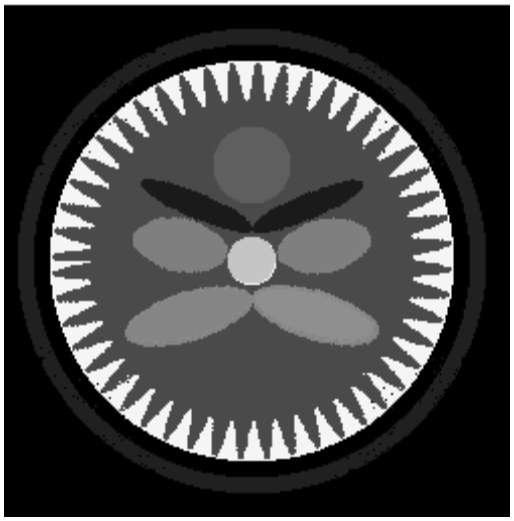
copies in the next generation. As such, particularly good schemata will propagate in future generations, thereby increasing the average fitness over time. In this way, the genetic algorithm is a global technique which shifts attention to the productive regions in the search space.

APPENDIX II

Model A

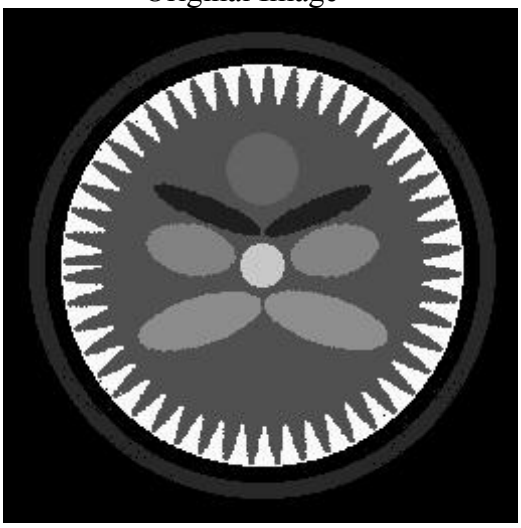
Original Image

Segmented Image



Model B

Original Image

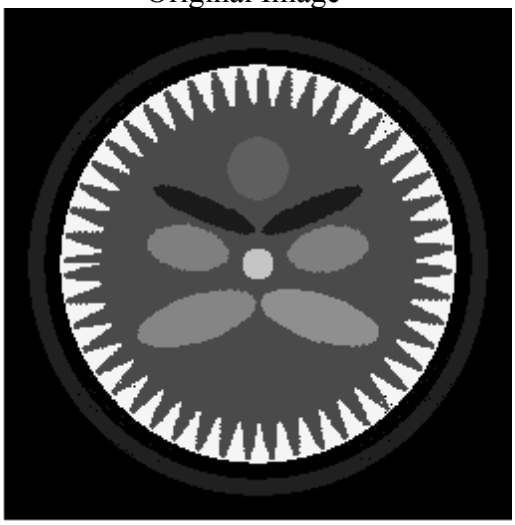


Segmented Image

FIGURE 18 – Segmented Image Compared To Original Image For Models A And B.

Model C

Original Image



Segmented Image

Model D

Original Image

Segmented Image

FIGURE 19 – Segmented Image Compared To Original Image For Models C And D.

Model E

Original Image

Segmented Image

Model F

Original Image

Segmented Image

FIGURE 20 – Segmented Image Compared To Original Image For Models E And F.

APPENDIX III

TABLE XV
SEGMENTATION ACCURACY OF TOP TEN MOST FIT STRINGS FOR T1 IMAGES. STRING #7 PERFORMS THE BEST ACROSS ALL IMAGES.

String Number	Features Represented in String	Subject With Healthy Structures	Subject With Alzheimer's Disease
1	3, 5, 6, 7, 9, 21	94.47	94.26
2	3, 5, 8, 10, 17, 18, 20, 21, 22, 23	94.05	93.89
3	3, 6, 7, 9, 13, 21	94.91	94.51
4	3, 4, 5, 6, 8, 13, 14, 17, 19, 23	95.08	94.35
5	1, 3, 5, 8, 14, 20, 21, 23	94.83	94.84
6	1, 3, 4, 6, 14, 20, 21, 23	94.81	94.50
7	1, 2, 3, 5, 7, 8, 9, 20, 21	95.94	95.31
8	1, 3, 4, 5, 6, 14, 23	95.06	95.05
9	3, 6, 7, 8, 9, 14, 20, 21, 23	94.45	94.45
10	1, 3, 4, 5, 6, 8, 14, 21, 23	95.33	94.93

TABLE XVI
SEGMENTATION ACCURACY OF TOP TEN MOST FIT STRINGS FOR T2 IMAGES. STRING #10 PERFORMS THE BEST ACROSS ALL IMAGES.

String Number	Features Represented in String	Subject With Healthy Structures	Subject With Alzheimer's Disease
1	1, 4, 7, 13	89.84	89.35
2	2, 3, 4, 5, 6, 17	91.07	90.06
3	3, 8, 9, 13, 17, 20, 23	90.71	89.96
4	3, 8, 9, 13, 18, 20, 23	90.86	90.13
5	1, 4, 7, 10, 13, 17	90.57	88.86
6	1, 4, 5, 6, 7, 8, 10, 13, 17	90.30	90.15
7	1, 4, 6, 7, 8, 10, 13, 17	90.33	90.01
8	2, 3, 4, 5, 13, 19	90.67	89.65
9	3, 6, 7, 10, 13, 19	90.94	90.09
10	1, 3, 4, 7, 13, 17, 22, 23	91.72	90.54

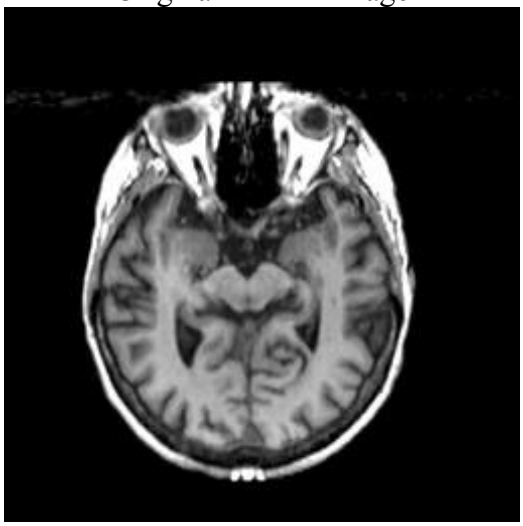
TABLE XVII

SEGMENTATION ACCURACY OF TOP TEN MOST FIT STRINGS FOR PD IMAGES. STRING #3 PERFORMS THE BEST ACROSS ALL IMAGES.

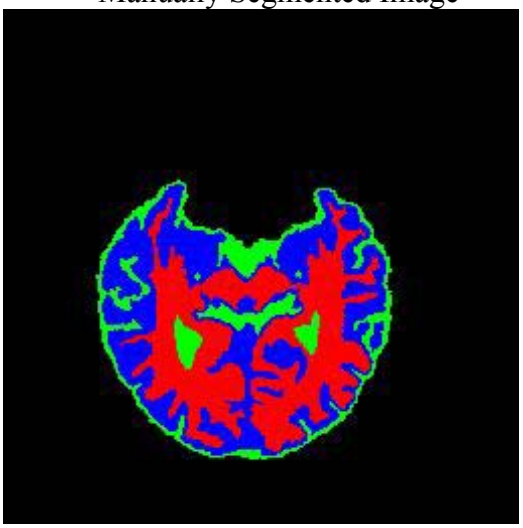
String Number	Features Represented in String	Set #1	Set #2
1	2, 6, 10, 12, 13, 14, 19, 21, 22, 23	94.32	94.81
2	4, 6, 8, 9, 13, 14, 17, 19	93.56	92.72
3	1, 2, 3, 7, 8, 9, 13, 14, 17, 21	95.51	95.36
4	1, 2, 3, 14, 18, 19, 20, 23	94.01	94.88
5	1, 3, 18, 21, 22, 23	93.84	93.53
6	4, 5, 10, 14, 17, 18, 22	95.28	94.33
7	2, 3, 4, 7, 8, 9, 18, 20, 23	94.43	95.01
8	1, 3, 14, 18, 19, 20, 21, 22, 23	94.55	94.35
9	4, 6, 7, 8, 10, 14, 17, 20	95.12	94.24
10	4, 7, 8, 10, 13, 14, 20, 23	94.17	93.48

APPENDIX IV

Original T1 MR Image



Manually Segmented Image



SOFM Segmented Image

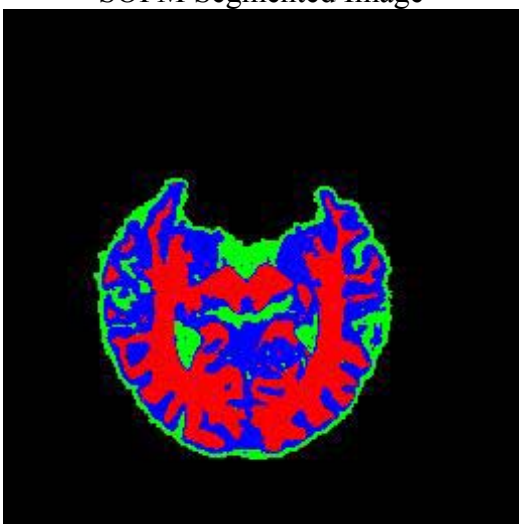
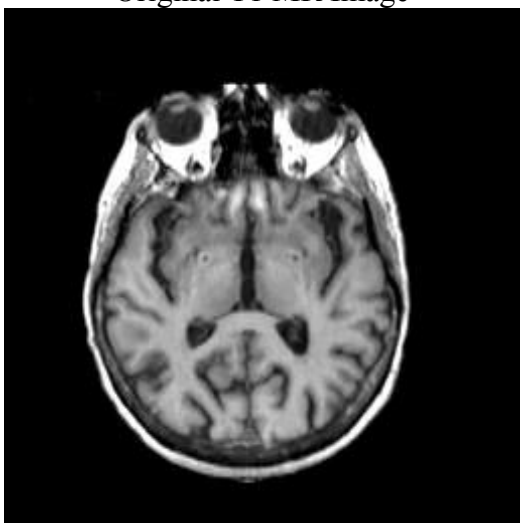
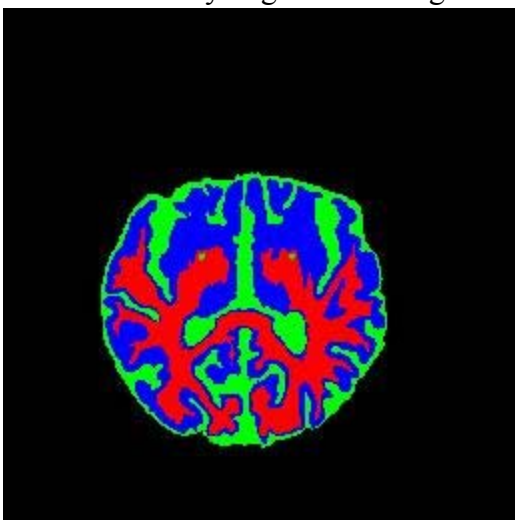


FIGURE 21 – T1 Segmentation Results For Subject With Healthy Brain Structures, Slice 23.

Original T1 MR Image



Manually Segmented Image



SOFM Segmented Image

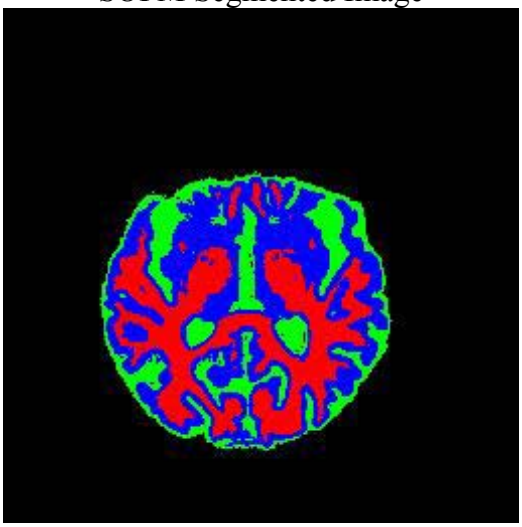
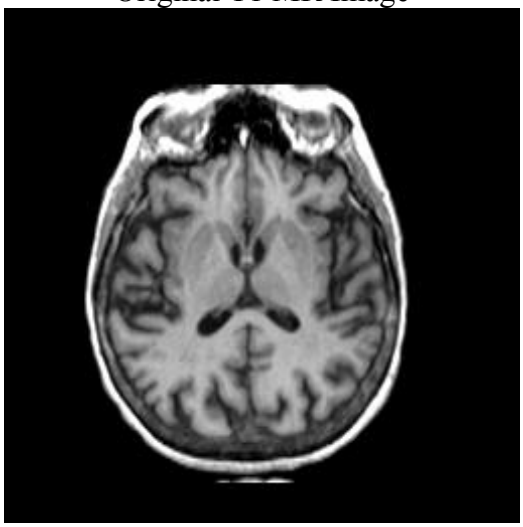
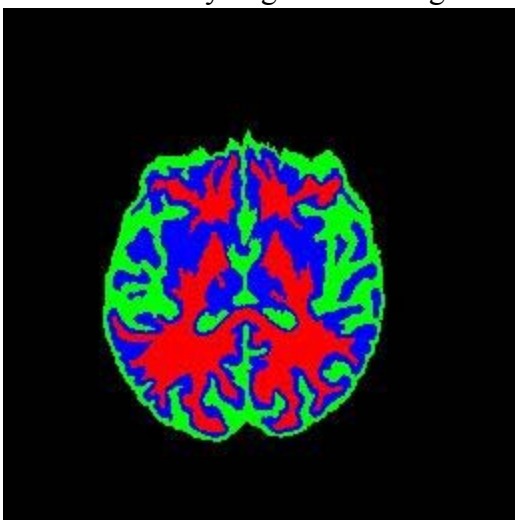


FIGURE 22 – T1 Segmentation Results For Subject With Healthy Brain Structures, Slice 26.

Original T1 MR Image



Manually Segmented Image



SOFM Segmented Image

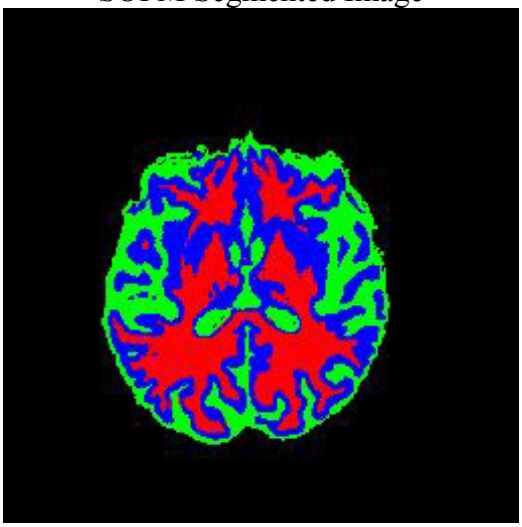
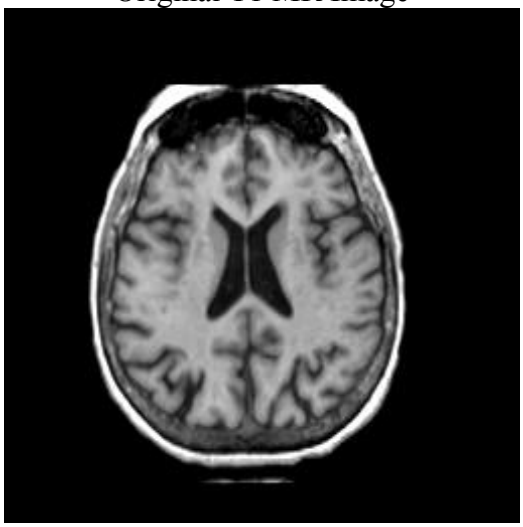
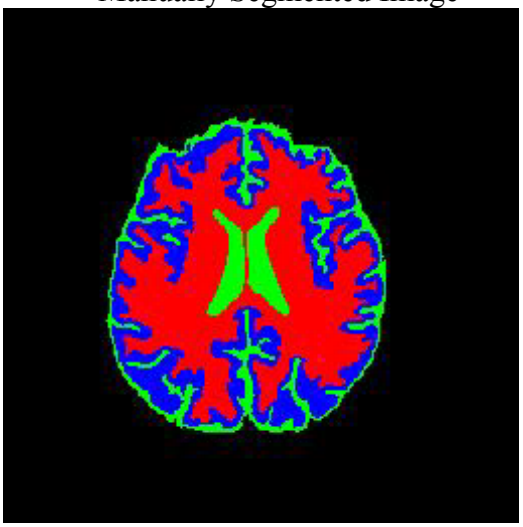


FIGURE 23 – T1 Segmentation Results For Subject With Healthy Brain Structures, Slice 29.

Original T1 MR Image



Manually Segmented Image



SOFM Segmented Image

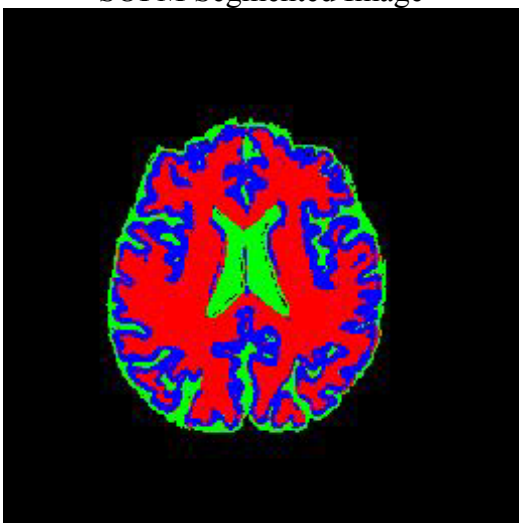
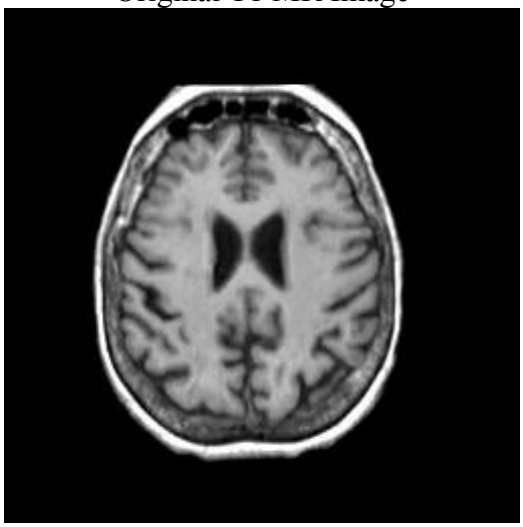
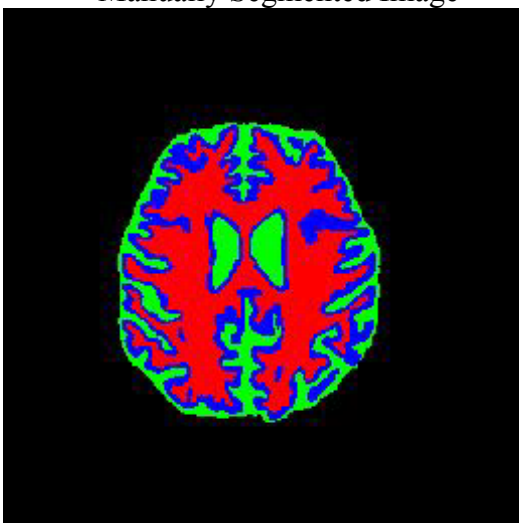


FIGURE 24 – T1 Segmentation Results For Subject With Healthy Brain Structures, Slice 32.

Original T1 MR Image



Manually Segmented Image



SOFM Segmented Image

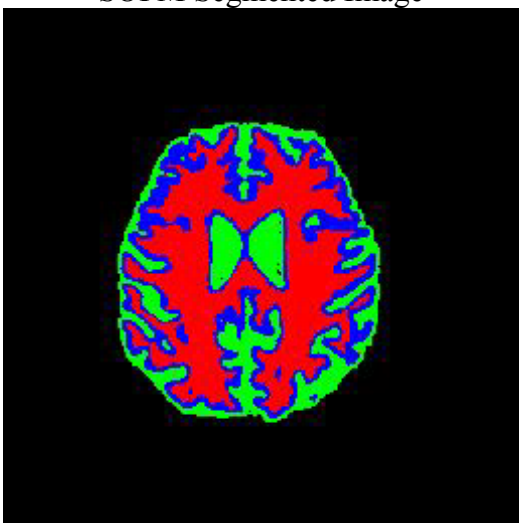
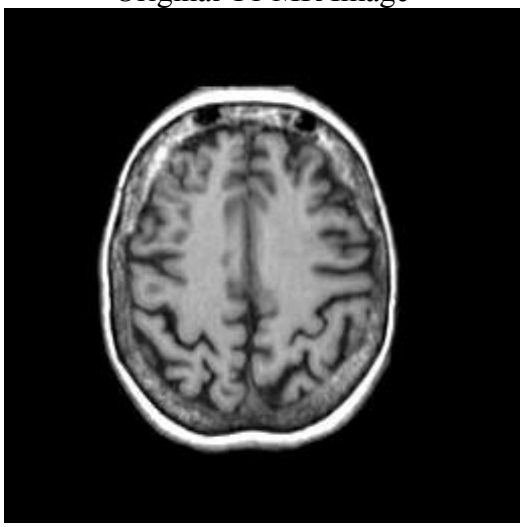
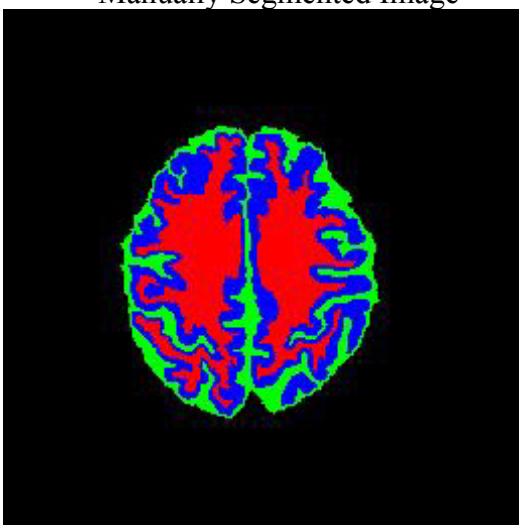


FIGURE 25 – T1 Segmentation Results For Subject With Healthy Brain Structures, Slice 35.

Original T1 MR Image



Manually Segmented Image



SOFM Segmented Image

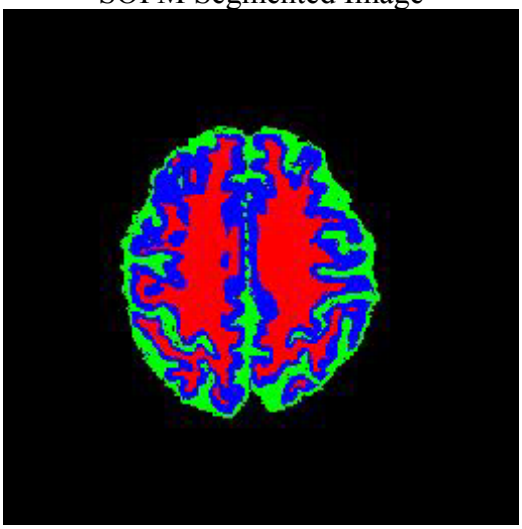
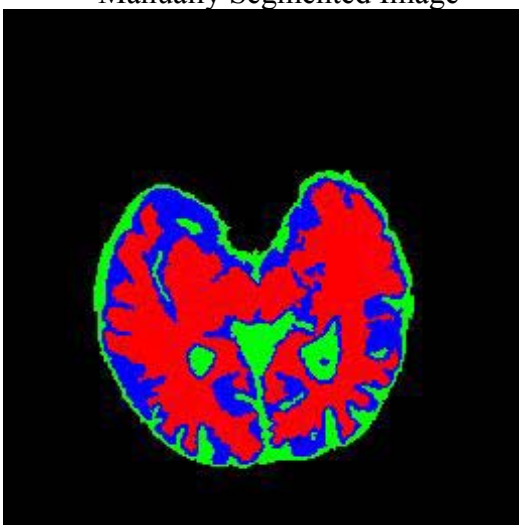


FIGURE 26 – T1 Segmentation Results For Subject With Healthy Brain Structures, Slice 38.

Original T1 MR Image

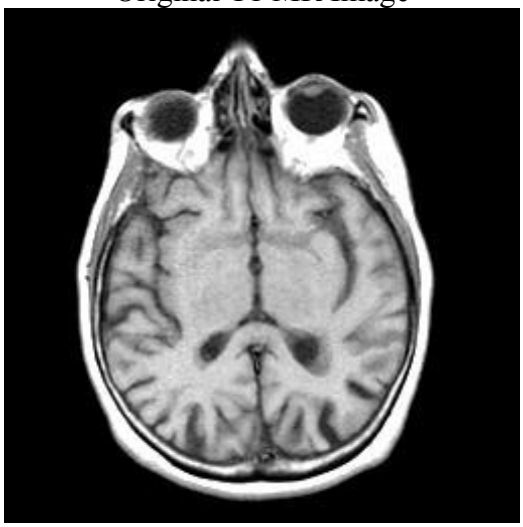
Manually Segmented Image



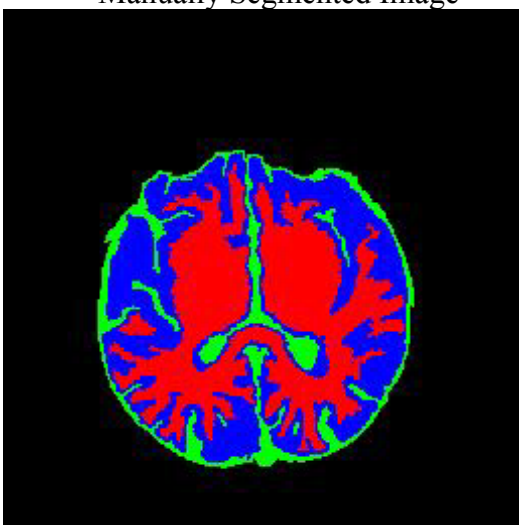
SOFM Segmented Image

FIGURE 27 – T1 Segmentation Results For Subject With Alzheimer's Disease, Slice 23.

Original T1 MR Image



Manually Segmented Image



SOFM Segmented Image

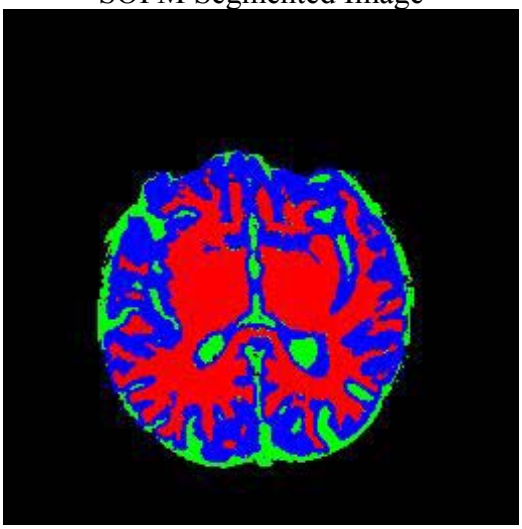
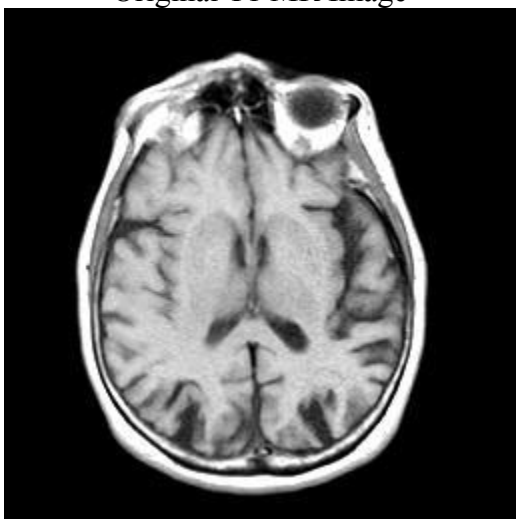
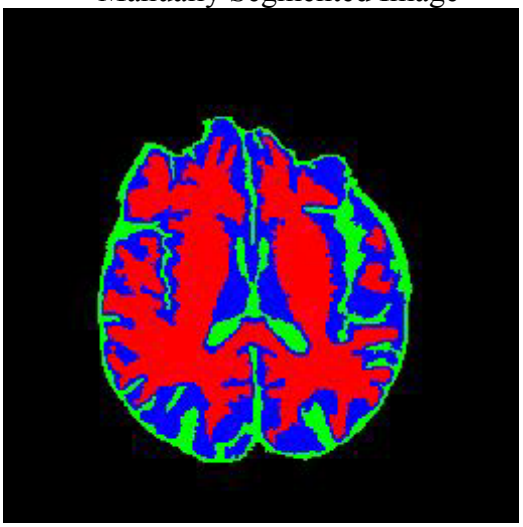


FIGURE 28 – T1 Segmentation Results For Subject With Alzheimer's Disease, Slice 26.

Original T1 MR Image



Manually Segmented Image



SOFM Segmented Image

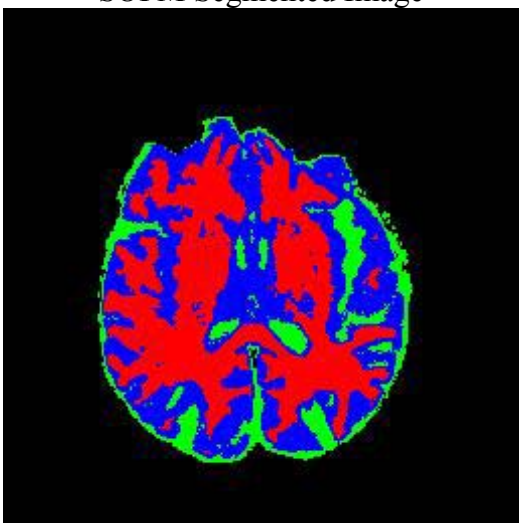
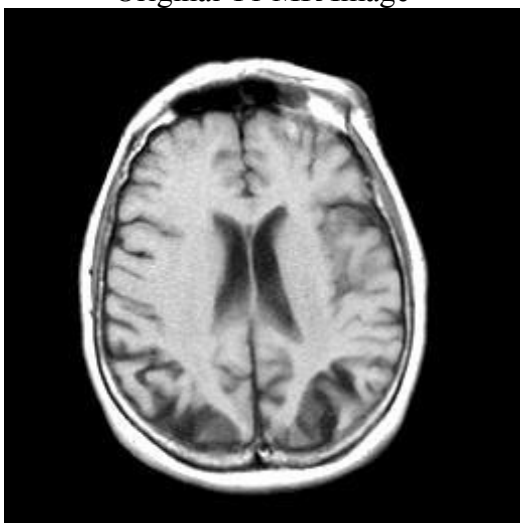
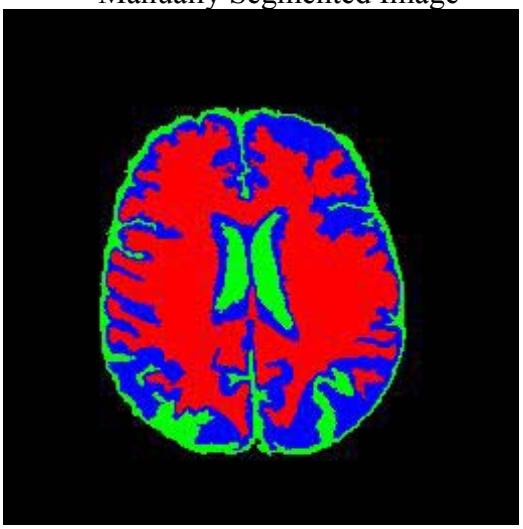


FIGURE 29 – T1 Segmentation Results For Subject With Alzheimer's Disease, Slice 29.

Original T1 MR Image



Manually Segmented Image



SOFM Segmented Image

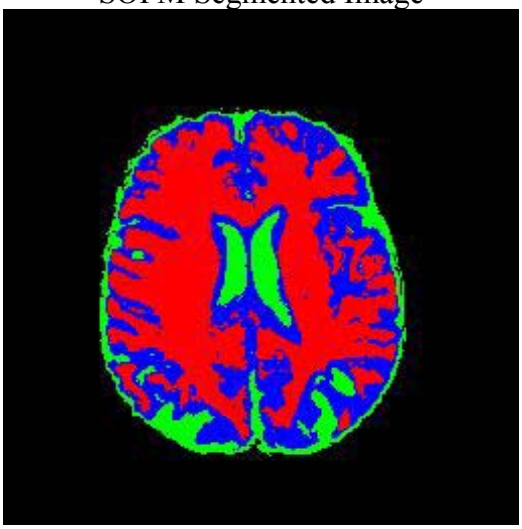
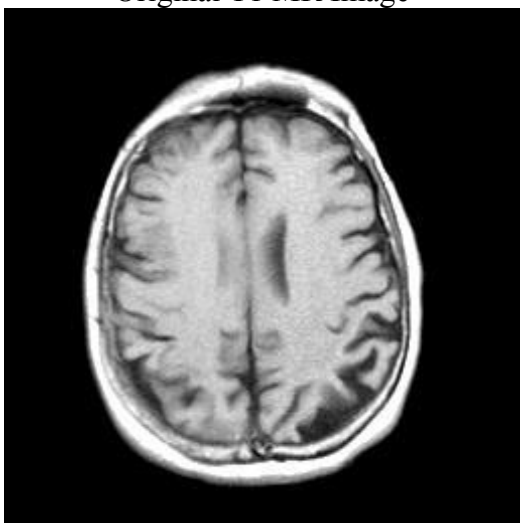
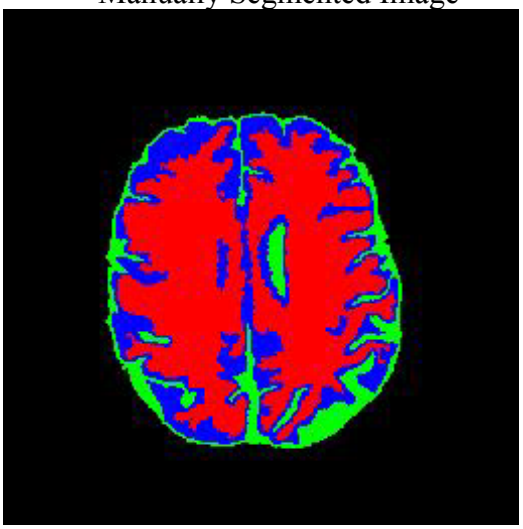


FIGURE 30 – T1 Segmentation Results For Subject With Alzheimer's Disease, Slice 32.

Original T1 MR Image



Manually Segmented Image



SOFM Segmented Image

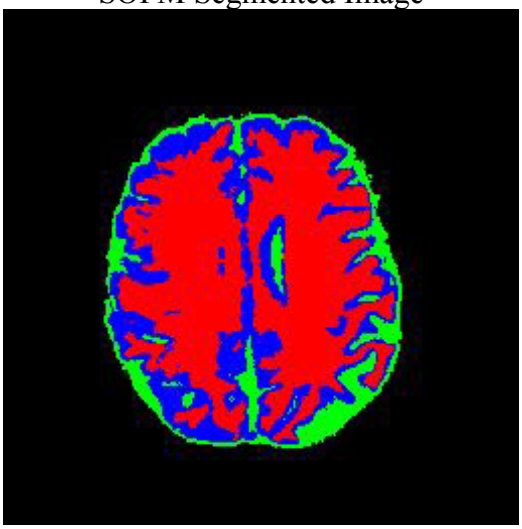
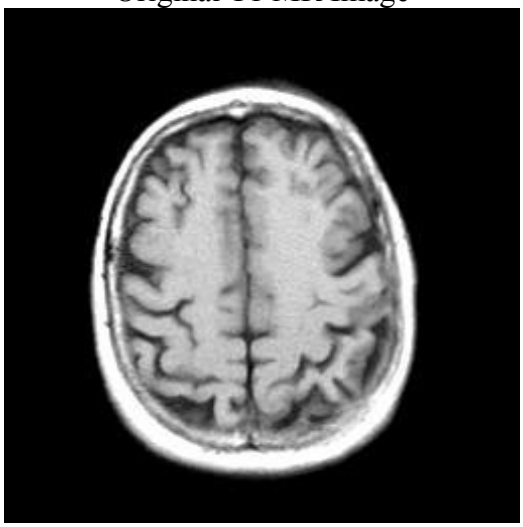
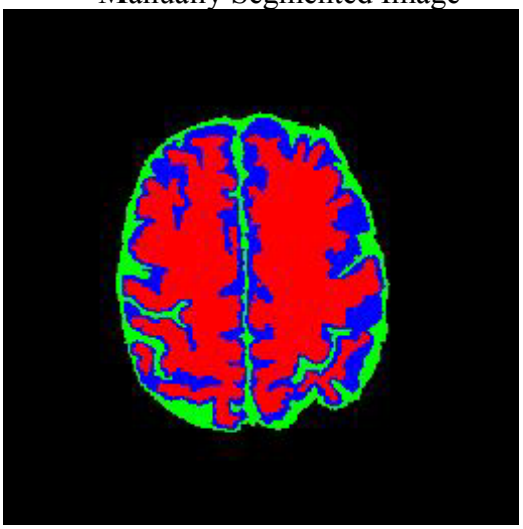


FIGURE 31 – T1 Segmentation Results For Subject With Alzheimer's Disease, Slice 35.

Original T1 MR Image



Manually Segmented Image



SOFM Segmented Image

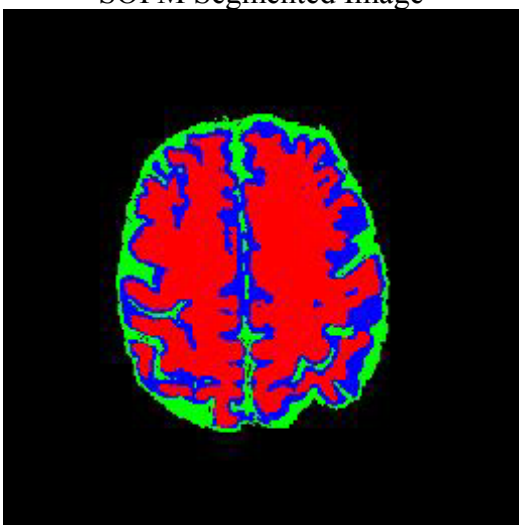
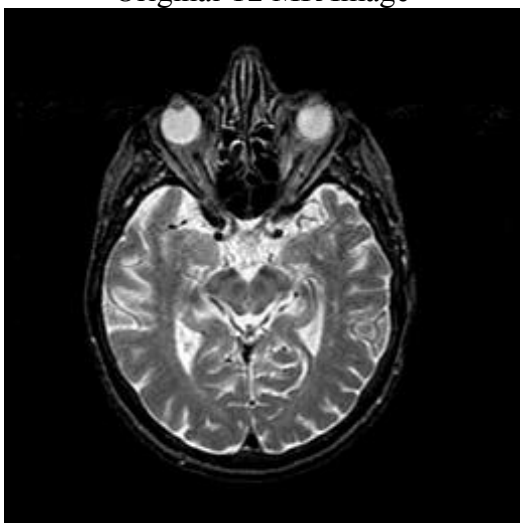
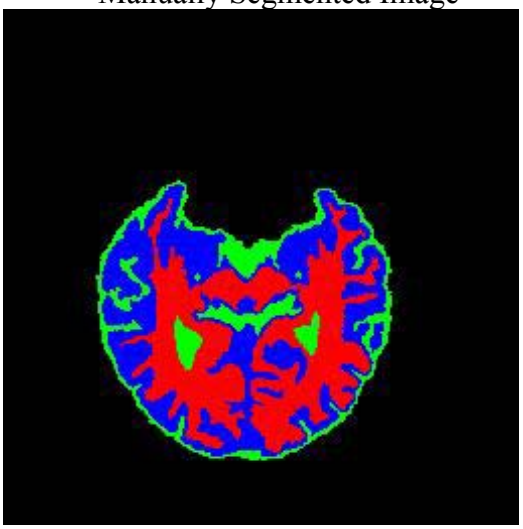


FIGURE 32 – T1 Segmentation Results For Subject With Alzheimer's Disease, Slice 38.

Original T2 MR Image



Manually Segmented Image



SOFM Segmented Image

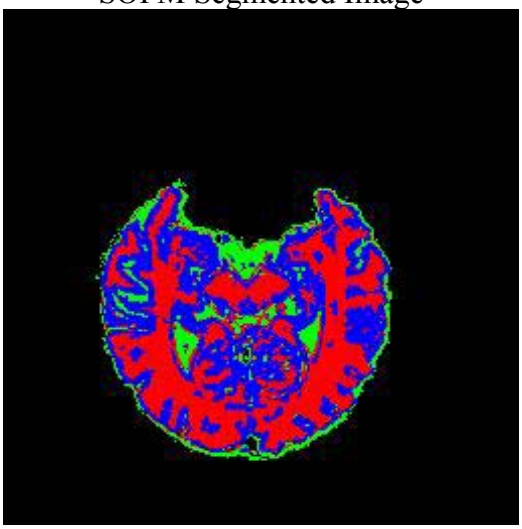
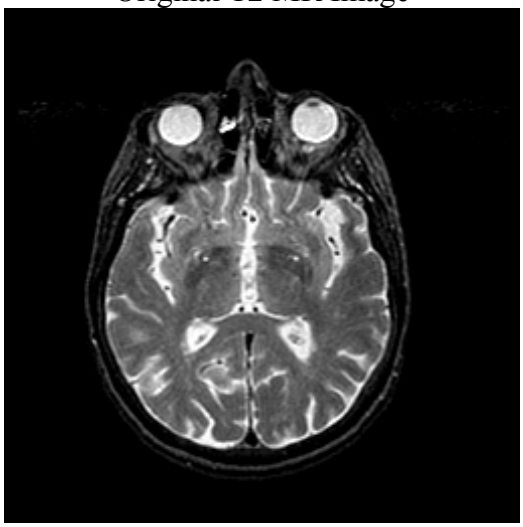
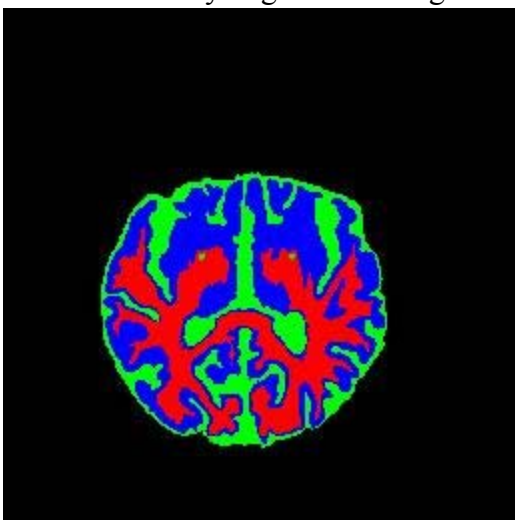


FIGURE 33 – T2 Segmentation Results For Subject With Healthy Brain Structures, Slice 23.

Original T2 MR Image



Manually Segmented Image



SOFM Segmented Image

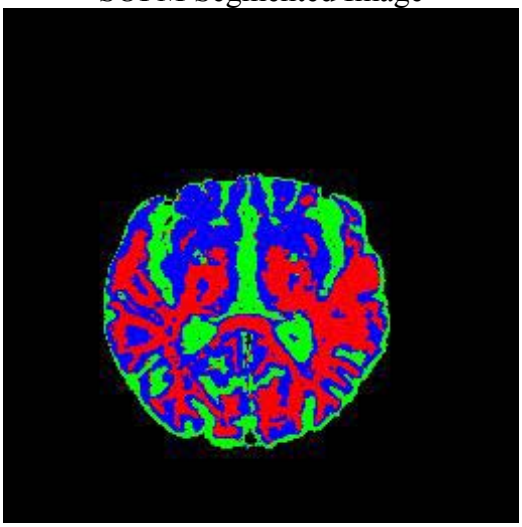
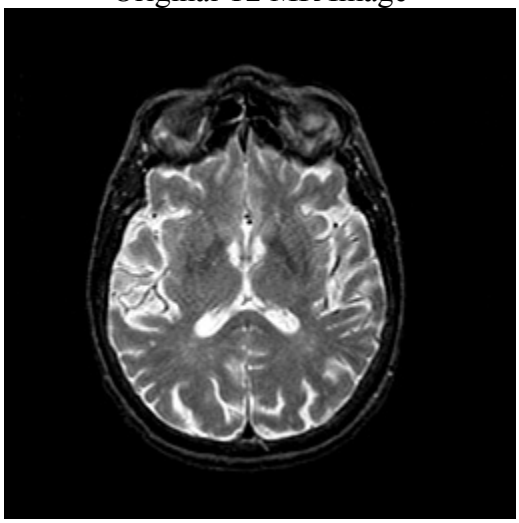
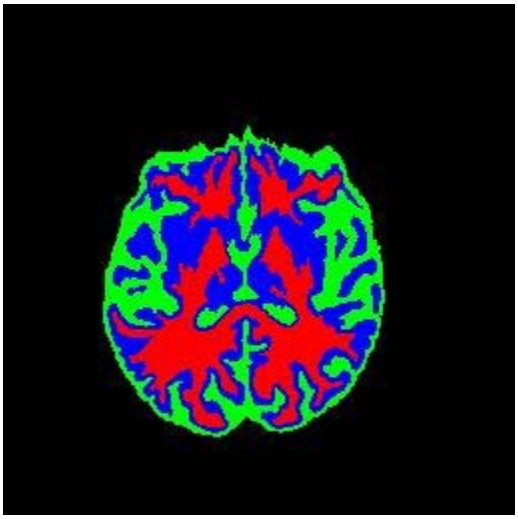


FIGURE 34 – T2 Segmentation Results For Subject With Healthy Brain Structures, Slice 26.

Original T2 MR Image



Manually Segmented Image



SOFM Segmented Image

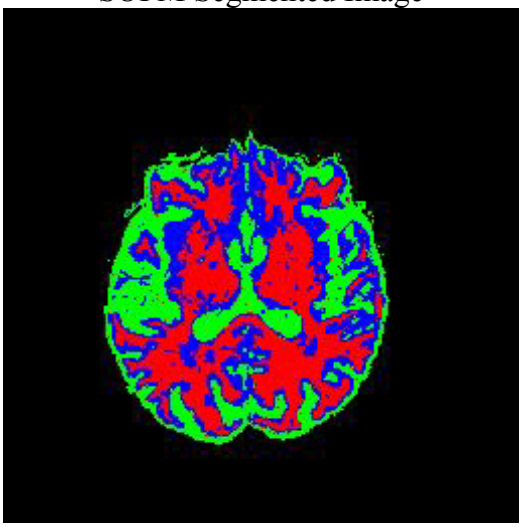
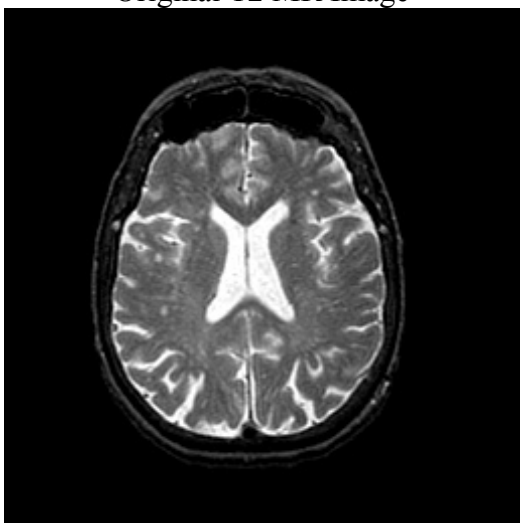
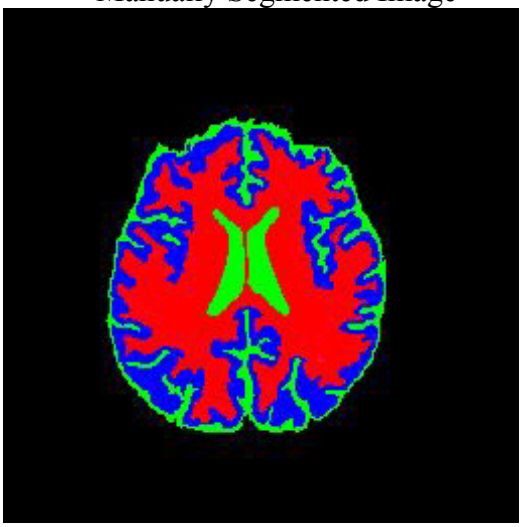


FIGURE 35 – T2 Segmentation Results For Subject With Healthy Brain Structures, Slice 29.

Original T2 MR Image



Manually Segmented Image



SOFM Segmented Image

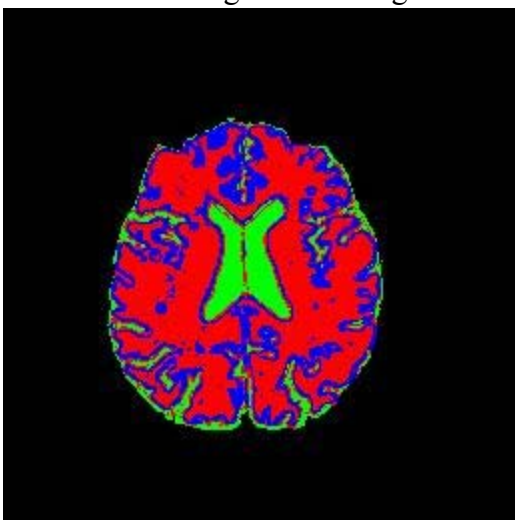
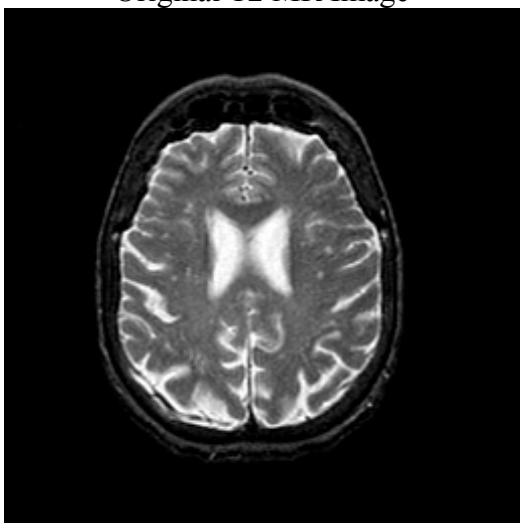
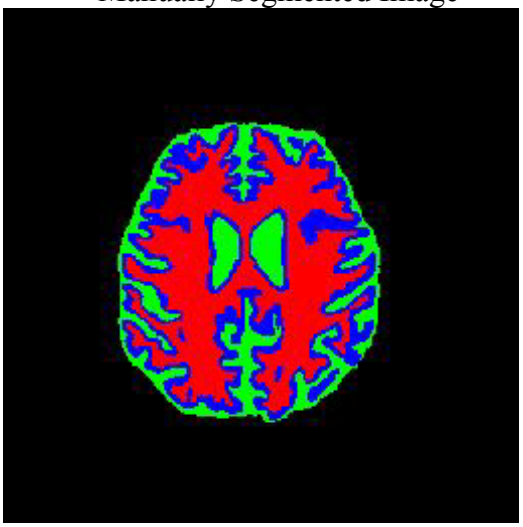


FIGURE 36 – T2 Segmentation Results For Subject With Healthy Brain Structures, Slice 32.

Original T2 MR Image



Manually Segmented Image



SOFM Segmented Image

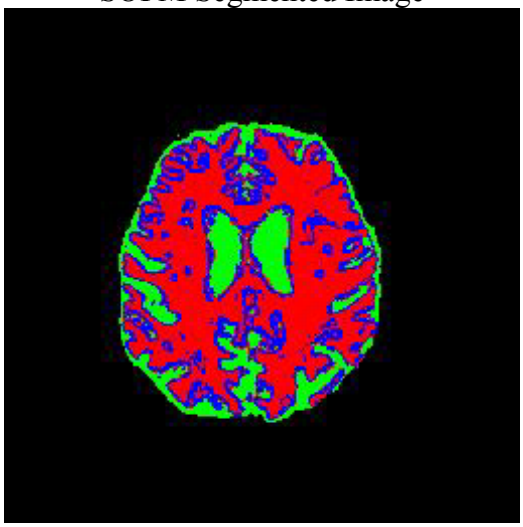
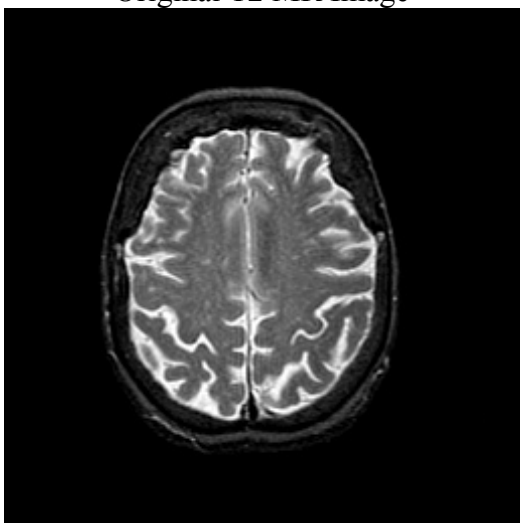
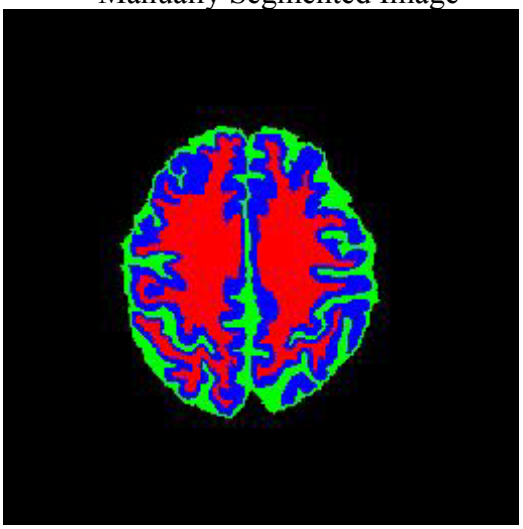


FIGURE 37 – T2 Segmentation Results For Subject With Healthy Brain Structures, Slice 35.

Original T2 MR Image



Manually Segmented Image



SOFM Segmented Image

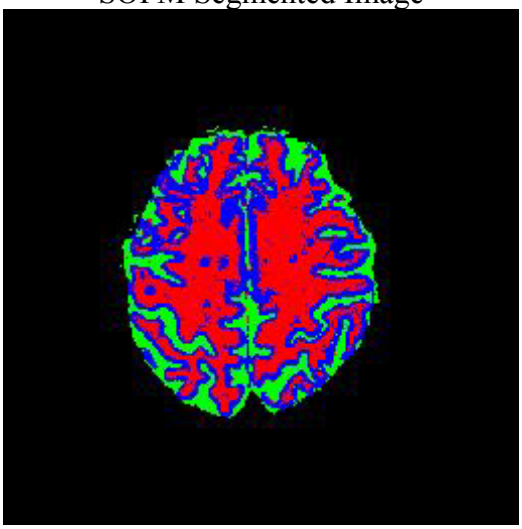
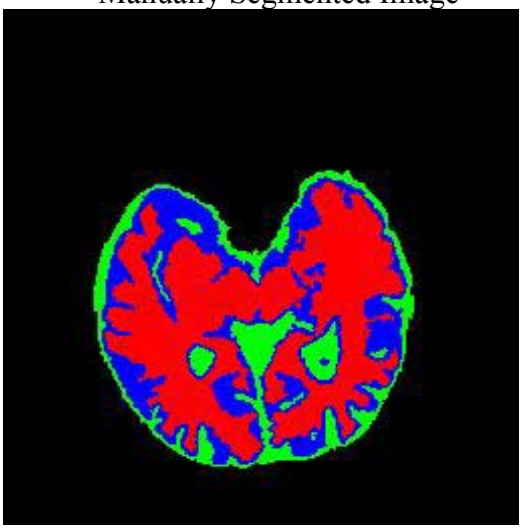


FIGURE 38 – T2 Segmentation Results For Subject With Healthy Brain Structures, Slice 38.

Original T2 MR Image

Manually Segmented Image



SOFM Segmented Image

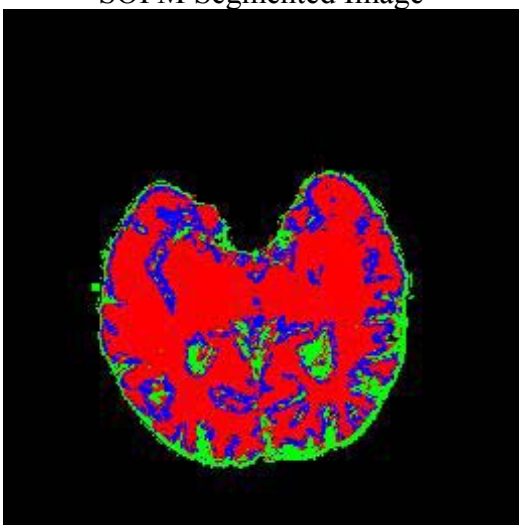
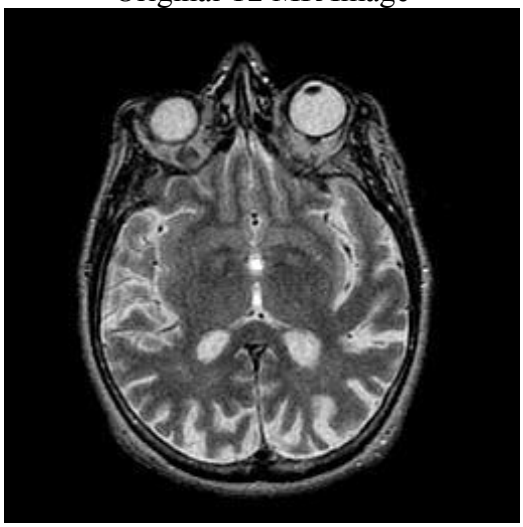
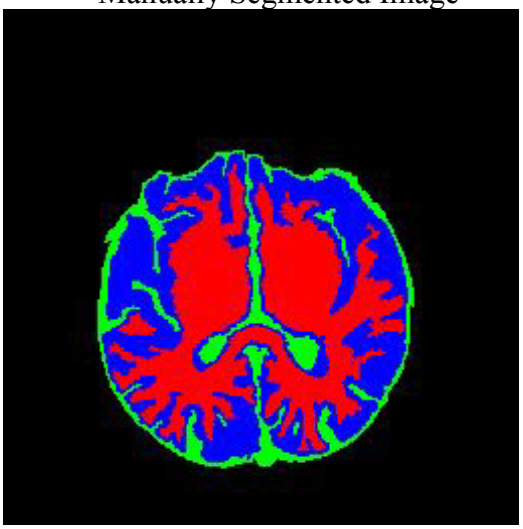


FIGURE 39 – T2 Segmentation Results For Subject With Alzheimer's Disease, Slice 23.

Original T2 MR Image



Manually Segmented Image



SOFM Segmented Image

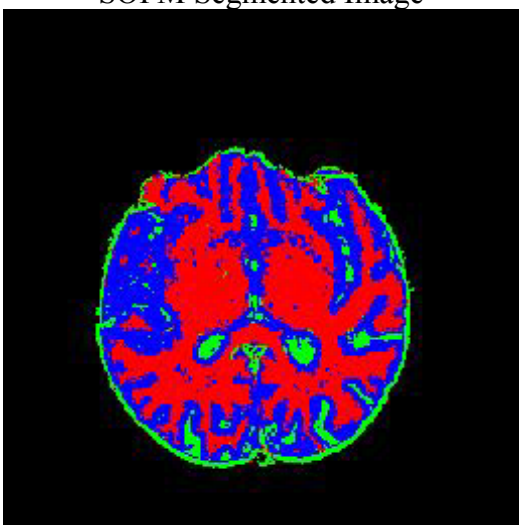
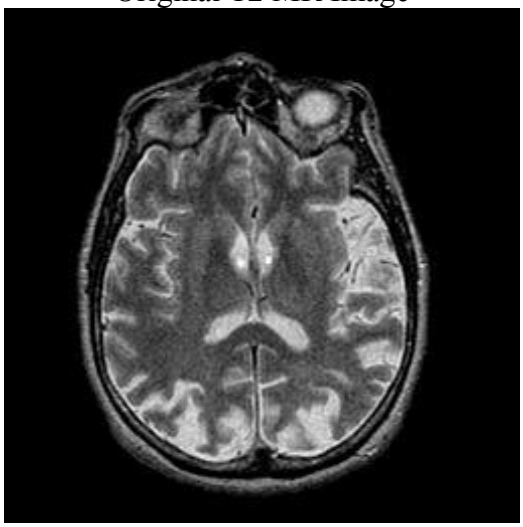
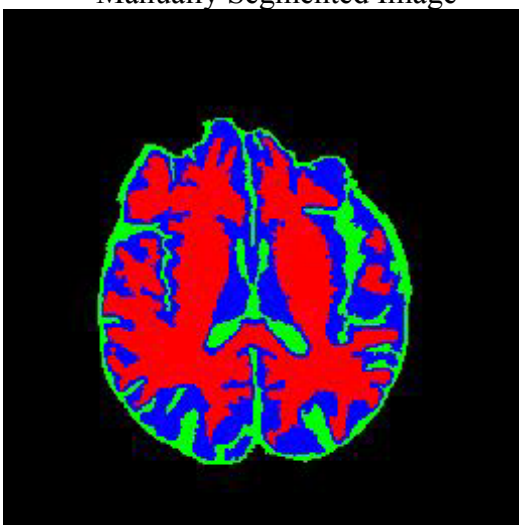


FIGURE 40 – T2 Segmentation Results For Subject With Alzheimer's Disease, Slice 26.

Original T2 MR Image



Manually Segmented Image



SOFM Segmented Image

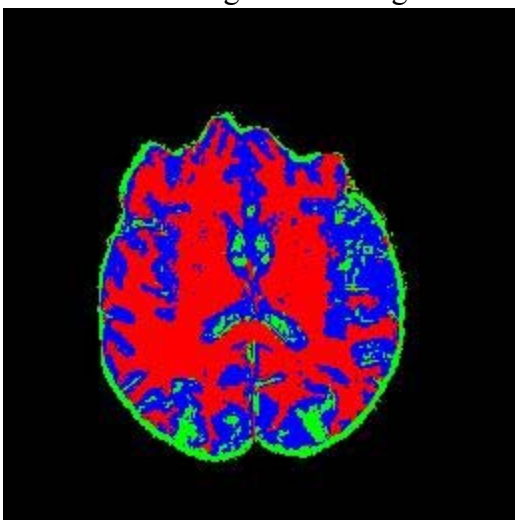
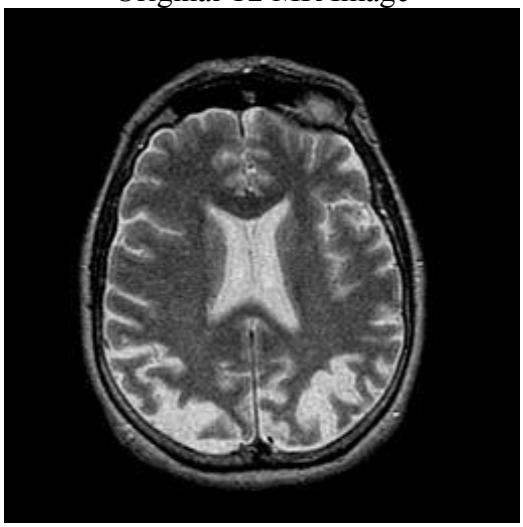
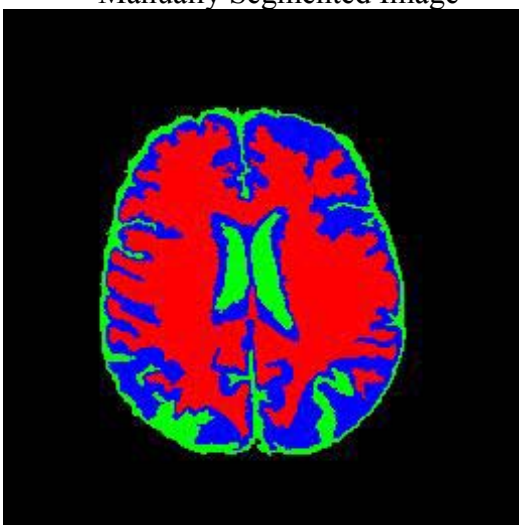


FIGURE 41 – T2 Segmentation Results For Subject With Alzheimer's Disease, Slice 29.

Original T2 MR Image



Manually Segmented Image



SOFM Segmented Image

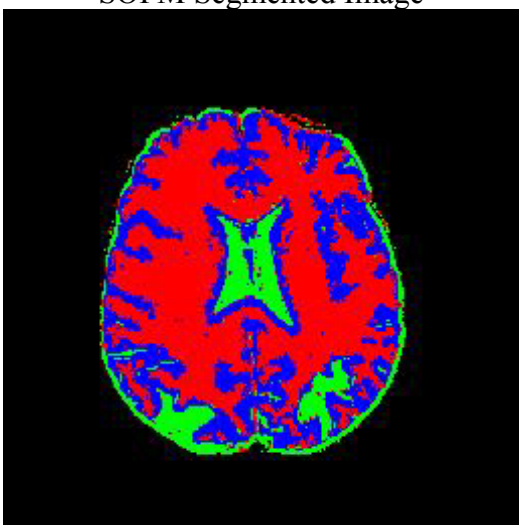
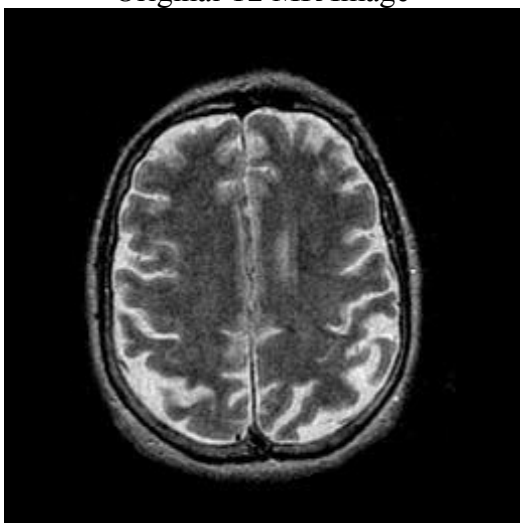
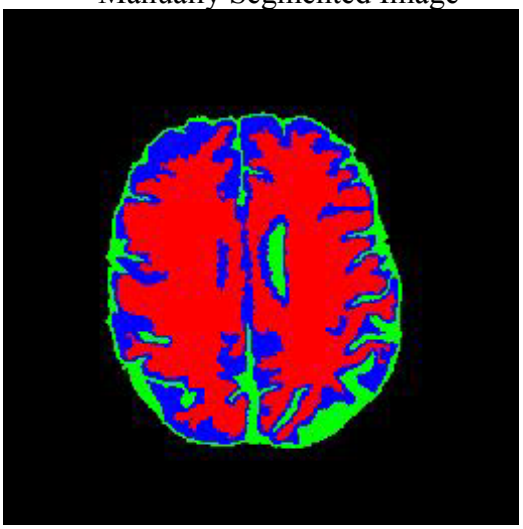


FIGURE 42 – T2 Segmentation Results For Subject With Alzheimer's Disease, Slice 32.

Original T2 MR Image



Manually Segmented Image



SOFM Segmented Image

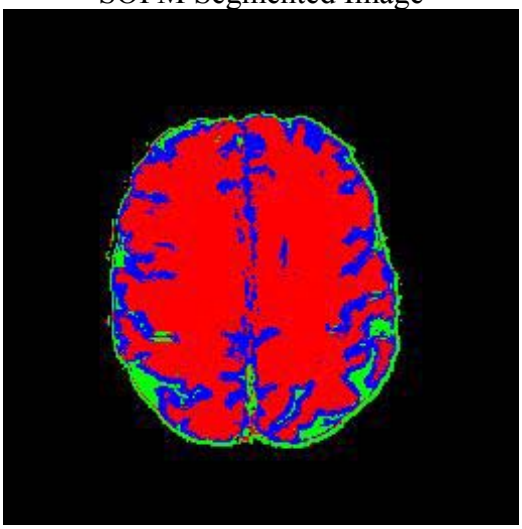
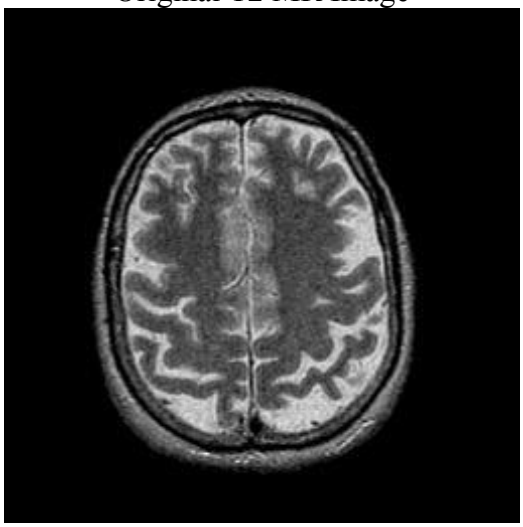
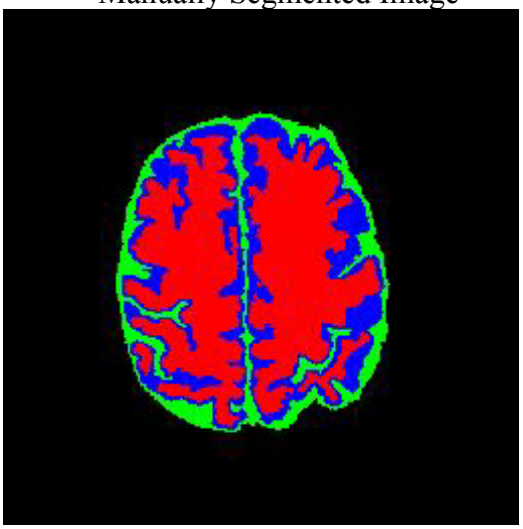


FIGURE 43 – T2 Segmentation Results For Subject With Alzheimer's Disease, Slice 35.

Original T2 MR Image



Manually Segmented Image



SOFM Segmented Image

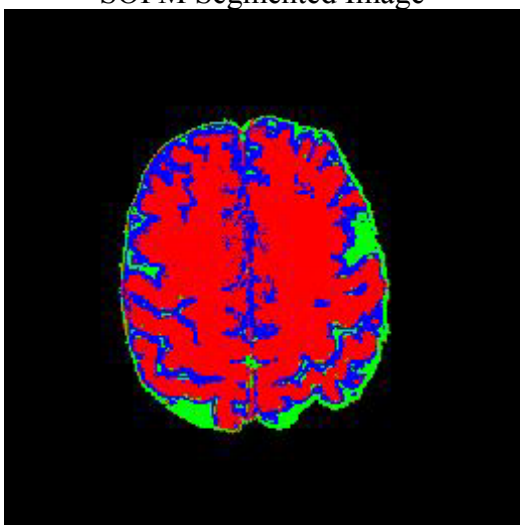
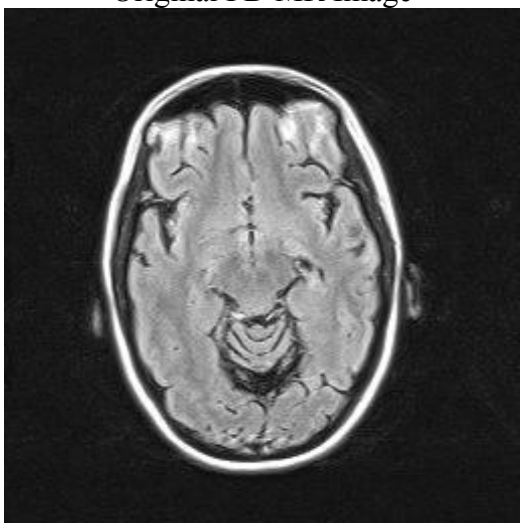
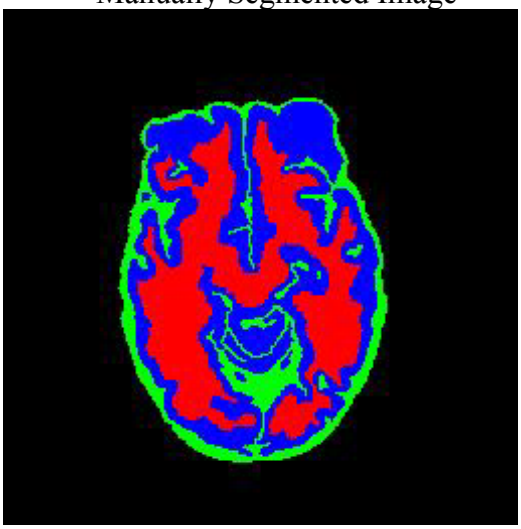


FIGURE 44 – T2 Segmentation Results For Subject With Alzheimer's Disease, Slice 38.

Original PD MR Image



Manually Segmented Image



SOFM Segmented Image

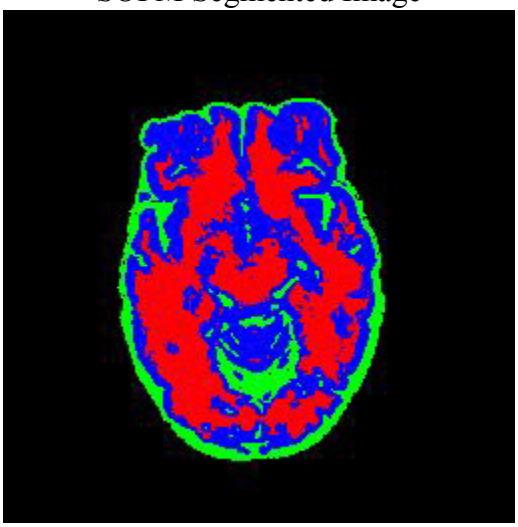
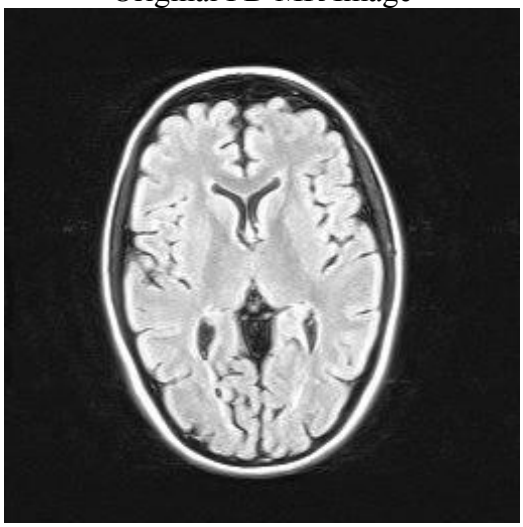
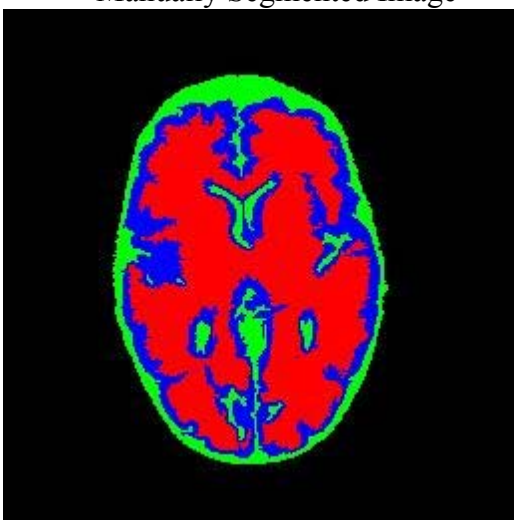


FIGURE 45 – PD Segmentation Results For Subject #1 With Multiple Sclerosis, Slice 8.

Original PD MR Image



Manually Segmented Image



SOFM Segmented Image

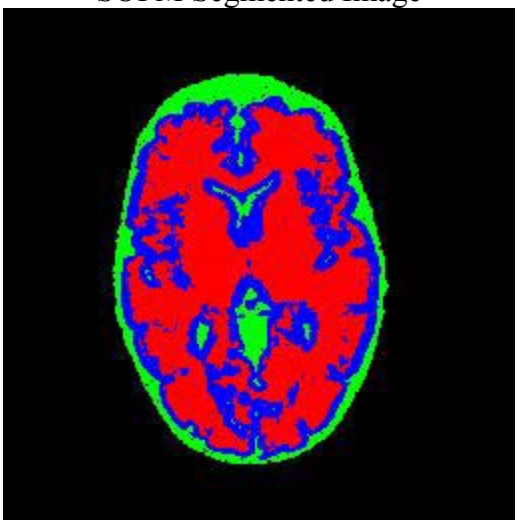
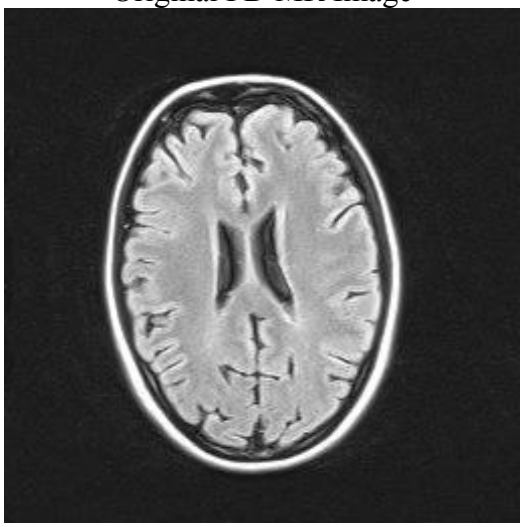
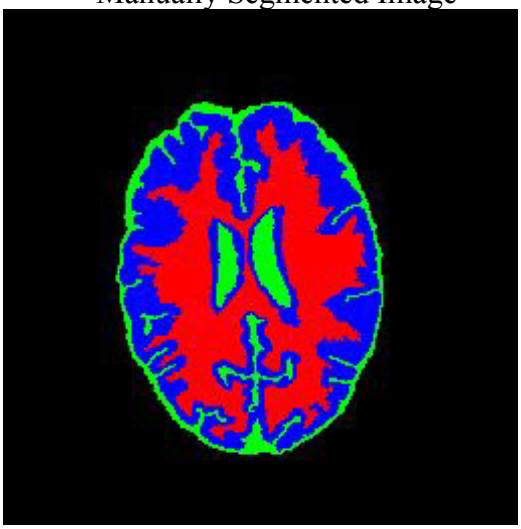


FIGURE 46 – PD Segmentation Results For Subject #1 With Multiple Sclerosis, Slice 10.

Original PD MR Image



Manually Segmented Image



SOFM Segmented Image

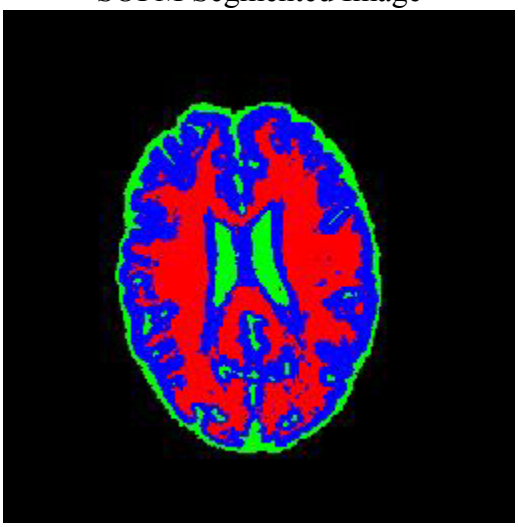
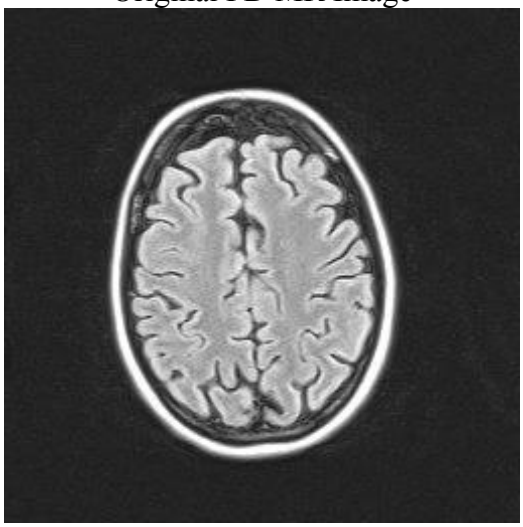
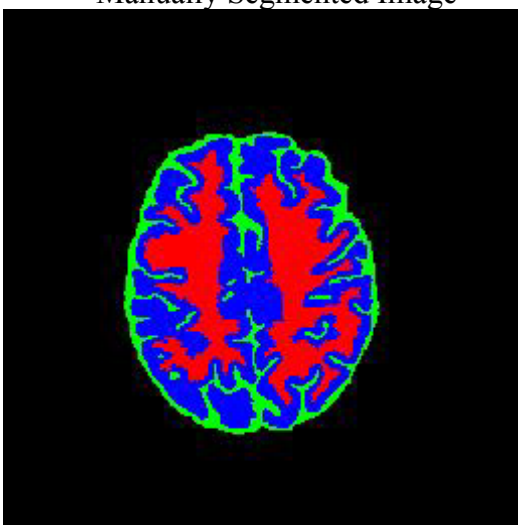


FIGURE 47 – PD Segmentation Results For Subject #1 With Multiple Sclerosis, Slice 12.

Original PD MR Image



Manually Segmented Image



SOFM Segmented Image

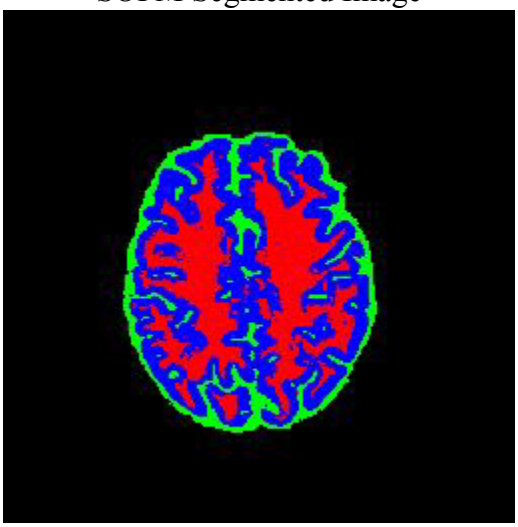
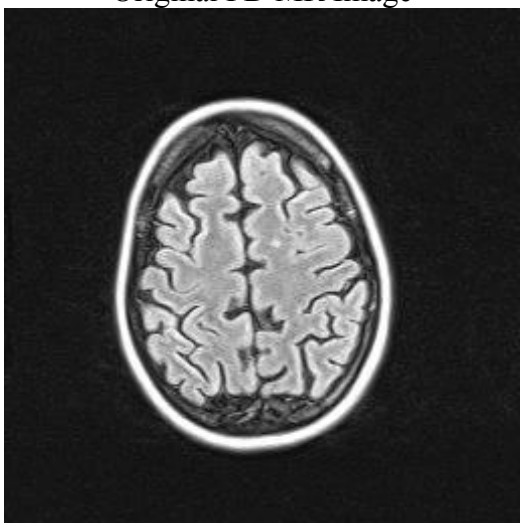
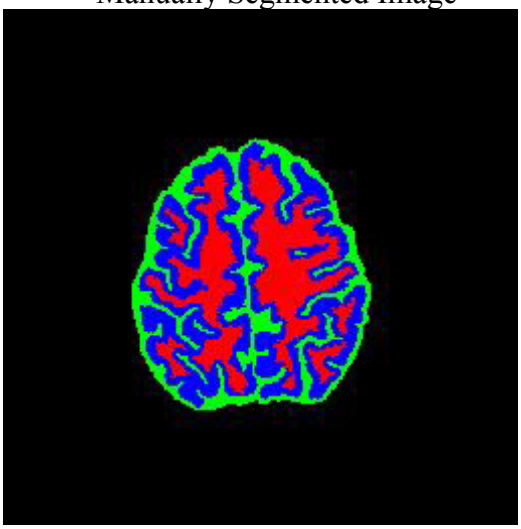


FIGURE 48 – PD Segmentation Results For Subject #1 With Multiple Sclerosis, Slice 14.

Original PD MR Image



Manually Segmented Image



SOFM Segmented Image

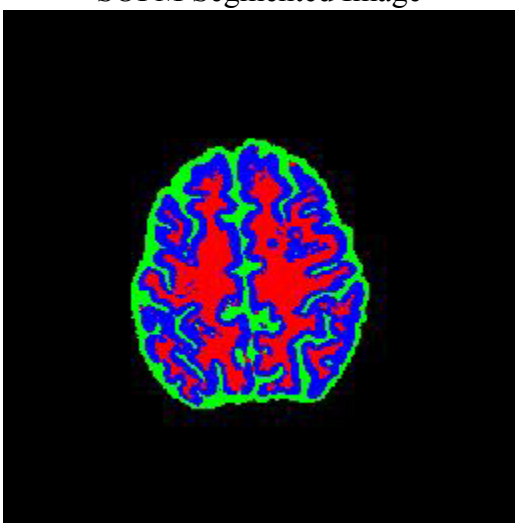
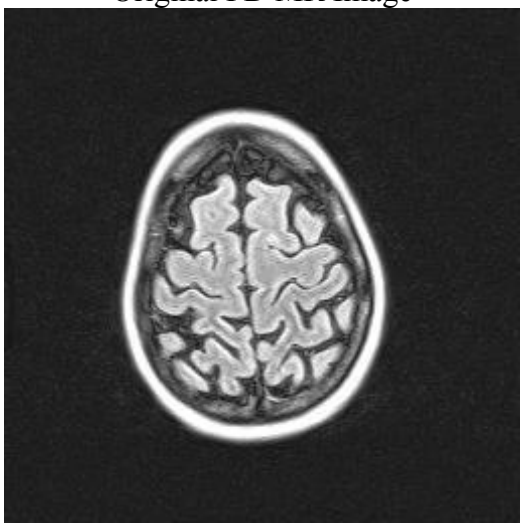
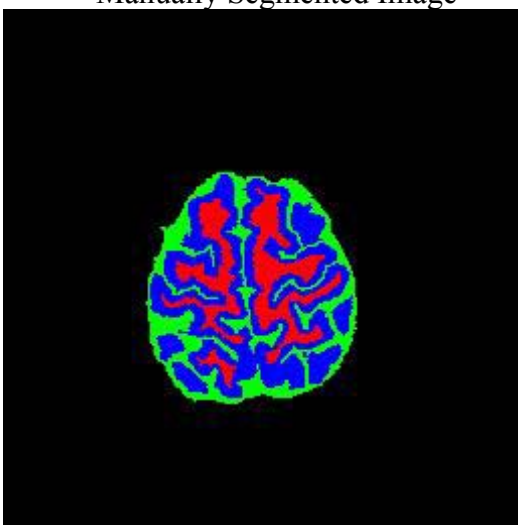


FIGURE 49 – PD Segmentation Results For Subject #1 With Multiple Sclerosis, Slice 15.

Original PD MR Image



Manually Segmented Image



SOFM Segmented Image

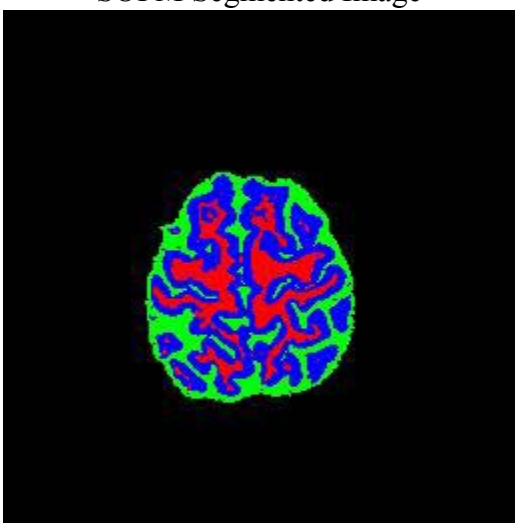
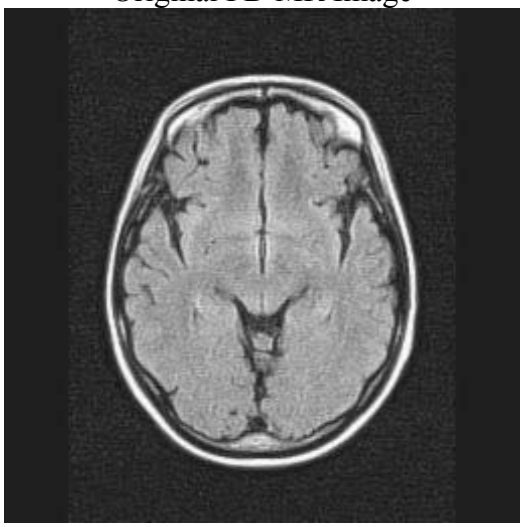
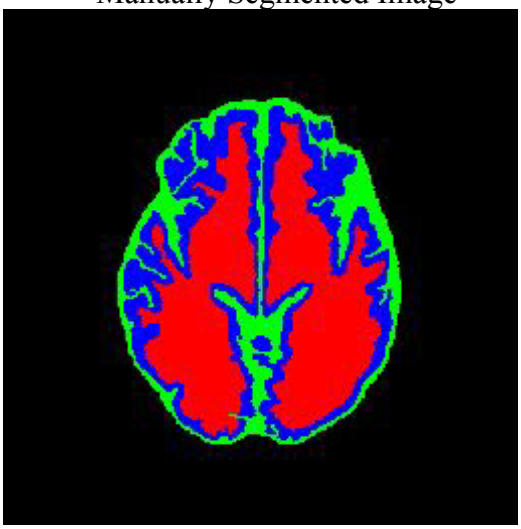


FIGURE 50 – PD Segmentation Results For Subject #1 With Multiple Sclerosis, Slice 16.

Original PD MR Image



Manually Segmented Image



SOFM Segmented Image

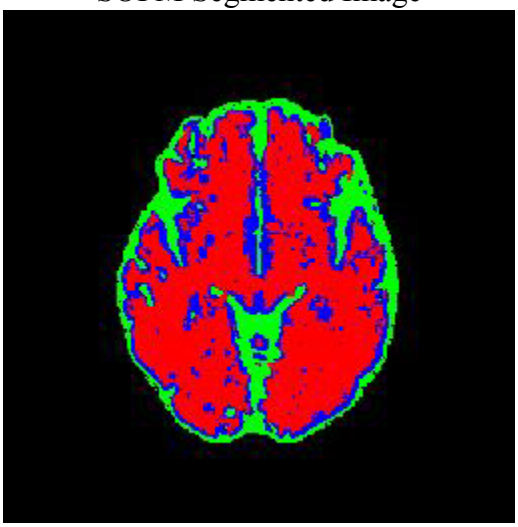
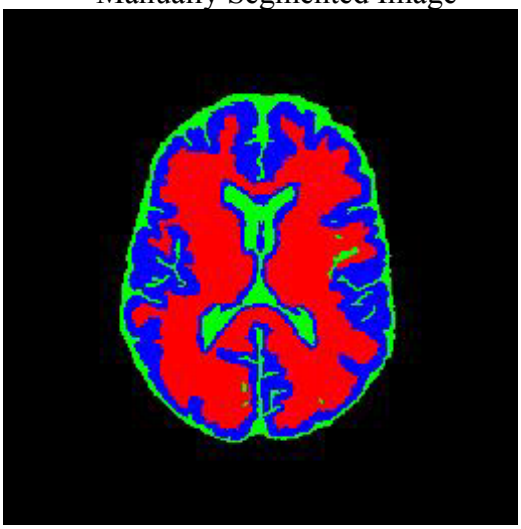


FIGURE 51 – PD Segmentation Results For Subject #2 With Multiple Sclerosis, Slice 8.

Original PD MR Image

Manually Segmented Image



SOFM Segmented Image

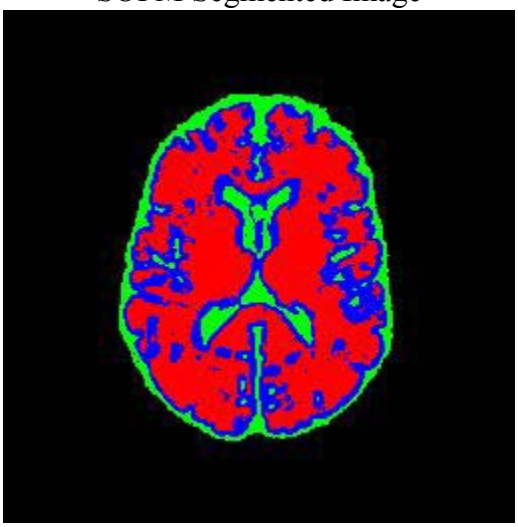
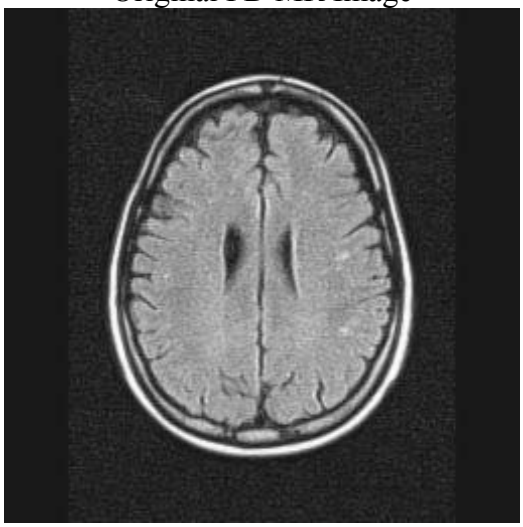
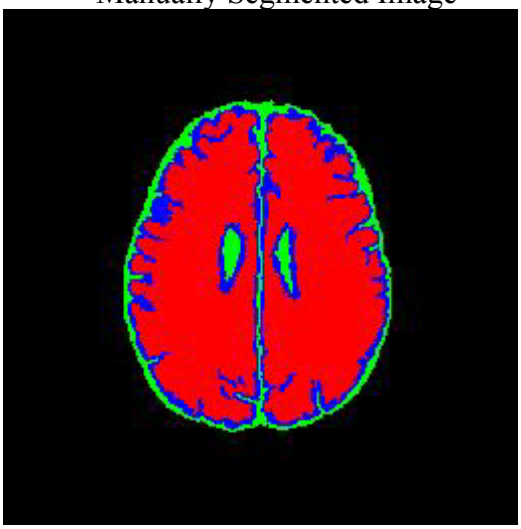


FIGURE 52 – PD Segmentation Results For Subject #2 With Multiple Sclerosis, Slice 10.

Original PD MR Image



Manually Segmented Image



SOFM Segmented Image

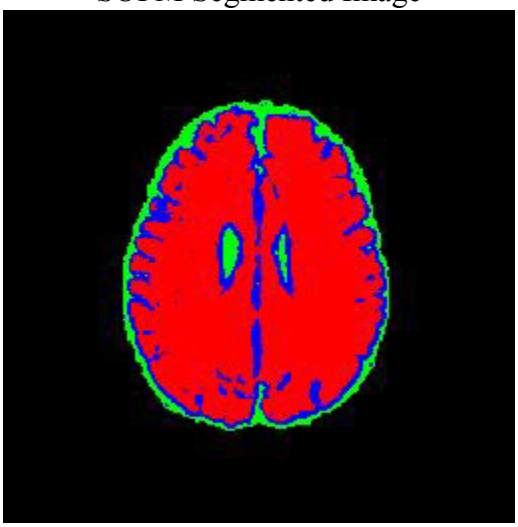
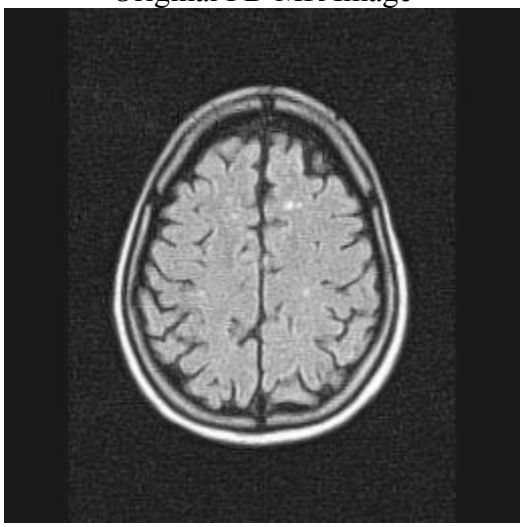
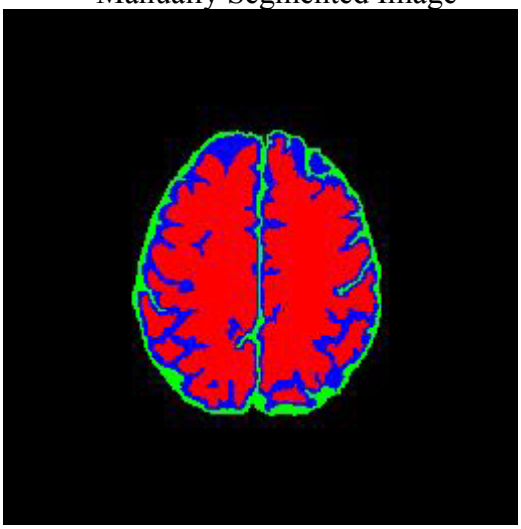


FIGURE 53 – PD Segmentation Results For Subject #2 With Multiple Sclerosis, Slice 12.

Original PD MR Image



Manually Segmented Image



SOFM Segmented Image

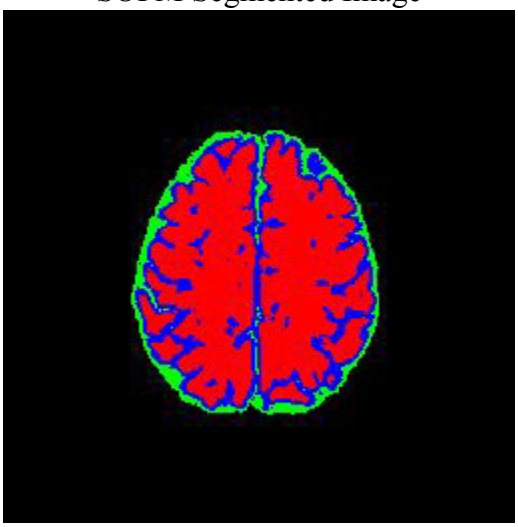
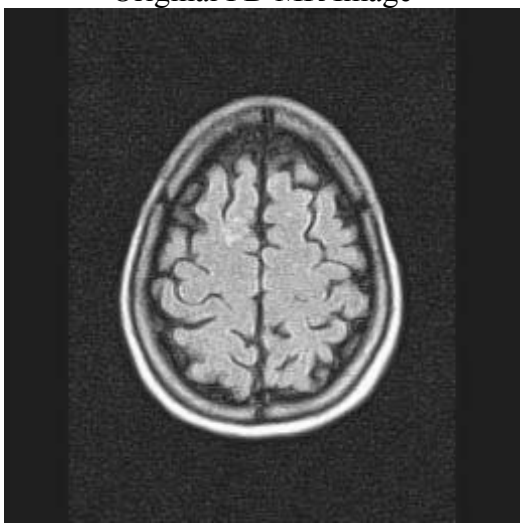
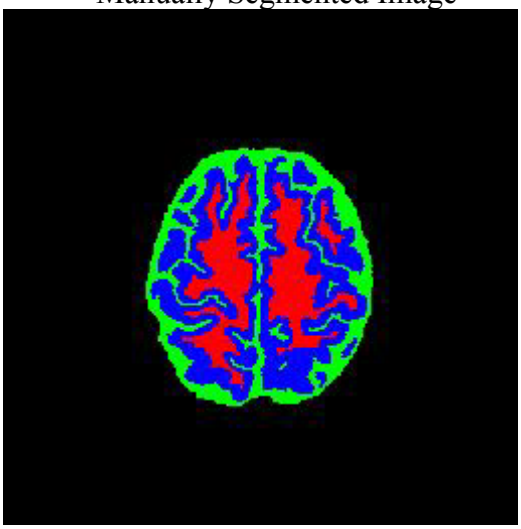


FIGURE 54 – PD Segmentation Results For Subject #2 With Multiple Sclerosis, Slice 14.

Original PD MR Image



Manually Segmented Image



SOFM Segmented Image

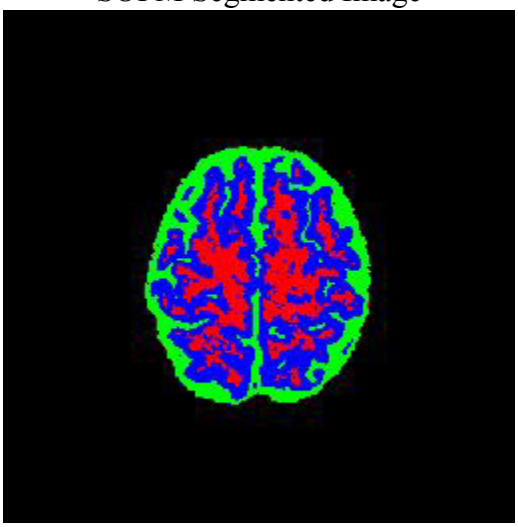
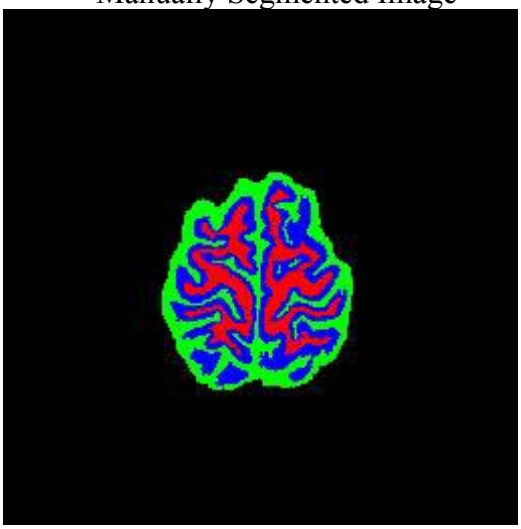


FIGURE 55 – PD Segmentation Results For Subject #2 With Multiple Sclerosis, Slice 15.

Original PD MR Image

Manually Segmented Image



SOFM Segmented Image

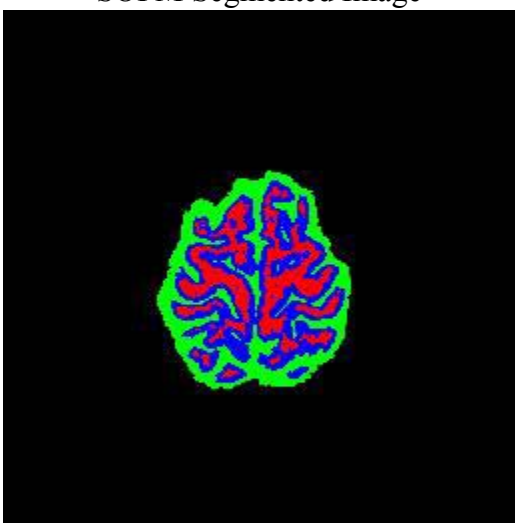


FIGURE 56 – PD Segmentation Results For Subject #2 With Multiple Sclerosis, Slice 16.

APPENDIX V

TABLE XVIII
SEGMENTATION ACCURACY OF ORIGINAL MODEL STRUCTURES, COMPARING GA WITH PCA.

Original Model		Number of Principal Components					
Brain Structure	**GA**	**4**	**5**	**6**	**7**	**8**	**9**
BKG	99.41	96.58	98.49	98.03	98.79	98.54	98.53
CC	99.36	96.86	95.41	96.48	98.66	97.48	97.82
FO	96.70	53.99	54.05	55.06	48.89	54.79	95.03
WM	99.79	95.64	94.07	95.38	97.25	96.62	97.84
LCN	72.59	55.28	20.81	34.72	67.46	10.04	21.20
LT	81.81	75.95	13.91	2.98	27.02	69.32	39.53
LV	49.32	52.08	45.67	26.31	43.87	16.43	70.32
RCN	87.18	22.03	59.70	47.36	14.14	29.28	68.00
RT	91.59	47.75	78.25	75.49	57.92	12.78	59.38
RV	63.30	41.33	35.27	59.68	52.81	64.41	12.30
SKN	98.52	96.24	95.95	95.64	94.36	94.55	96.88
GM	98.99	91.46	94.99	93.68	90.92	95.14	94.88
Total	98.17	91.43	91.37	90.97	91.85	91.40	93.76

TABLE XIX
SEGMENTATION ACCURACY OF MODEL STRUCTURES DEGRADED WITH GAUSSIAN NOISE, SNR = 30, COMPARING GA WITH PCA.

Model , SNR = 30		Number of Principal Components					
Brain Structure	**GA**	**4**	**5**	**6**	**7**	**8**	**9**
BKG	99.20	96.81	97.37	97.98	98.60	98.08	99.25
CC	99.90	99.31	96.99	98.74	98.25	98.10	98.04
FO	95.51	51.62	41.99	54.81	55.14	72.45	56.60
WM	99.57	96.36	94.89	96.32	94.92	95.99	95.98
LCN	43.80	31.60	68.31	48.13	64.59	29.65	51.45
LT	63.00	75.06	14.78	39.08	68.92	49.69	30.24
LV	40.07	46.09	56.89	41.02	24.43	37.82	66.06
RCN	58.27	81.60	46.03	64.86	43.15	64.83	67.32
RT	69.09	28.38	65.94	45.08	17.95	46.56	62.86
RV	68.35	52.30	41.01	41.55	69.30	42.50	26.11
SKN	94.31	92.98	93.72	92.78	93.18	94.70	94.87
GM	99.97	91.56	92.93	93.87	92.78	90.85	92.49
Total	96.59	91.84	90.97	91.72	92.50	91.69	92.02

TABLE XX

SEGMENTATION ACCURACY OF MODEL STRUCTURES DEGRADED WITH GAUSSIAN NOISE, SNR = 20, COMPARING GA WITH PCA.

Model, SNR = 20		**Number of Principal Components**					
Brain Structure	**GA**	**4**	**5**	**6**	**7**	**8**	**9**
BKG	99.90	93.76	95.41	92.37	96.16	93.11	93.36
CC	74.15	94.14	95.74	93.90	97.22	94.61	96.81
FO	99.21	47.02	37.84	51.82	52.03	69.31	50.56
WM	99.62	92.76	93.06	93.41	90.10	94.81	92.36
LCN	55.53	18.19	44.70	45.76	59.71	28.58	47.38
LT	24.44	70.08	9.818	38.77	62.87	45.24	30.06
LV	22.54	42.23	53.95	36.12	21.78	33.25	64.92
RCN	75.04	76.69	40.62	60.90	37.79	58.65	60.78
RT	65.56	23.54	62.09	42.10	11.94	44.19	59.16
RV	22.29	50.35	39.19	40.23	63.78	18.09	24.39
SKN	94.87	90.39	86.87	87.95	92.34	90.18	91.14
GM	99.50	86.72	89.93	93.76	90.45	87.13	88.00
Total	93.73	86.89	88.52	90.68	90.45	88.31	90.10

TABLE XXI

SEGMENTATION ACCURACY OF MODEL STRUCTURES DEGRADED WITH GAUSSIAN NOISE, SNR = 10, COMPARING GA WITH PCA.

Model, SNR = 10		**Number of Principal Components**					
Brain Structure	**GA**	**4**	**5**	**6**	**7**	**8**	**9**
BKG	97.02	90.00	80.90	73.63	89.67	95.27	75.47
CC	22.70	65.99	82.23	44.67	75.32	82.77	59.19
FO	43.64	33.43	36.34	32.67	40.89	59.52	41.66
WM	96.17	73.11	94.21	88.05	82.37	73.81	85.24
LCN	58.58	5.21	26.51	33.84	28.54	16.33	2.95
LT	49.86	67.68	11.13	26.22	48.78	44.86	21.34
LV	92.71	0.13	9.83	19.57	9.02	15.39	17.89
RCN	26.62	57.66	24.25	52.27	36.66	48.32	51.19
RT	20.86	24.47	46.48	35.35	9.80	43.06	50.53
RV	45.77	0.06	30.77	28.42	4.22	35.46	2.26
SKN	83.29	86.98	82.36	71.05	74.65	83.27	79.27

GM	97.11	76.50	85.50	76.55	72.44	74.84	79.11
Total	87.58	75.61	77.32	80.15	71.68	79.67	75.57

APPENDIX VI

TABLE XXII

PCA SEGMENTATION ACCURACY FOR THE T1 MRI DATA SETS OF THE SUBJECT WITH HEALTHY STRUCTURES AND THE SUBJECT WITH ALZHEIMER'S DISEASE.

Subject With Healthy Structures						
T1 Volume		**Number of Principal Components**				
Tissue	**GA**	**6**	**7**	**8**	**9**	**10**
BKG	99.61	99.54	99.08	99.62	99.56	99.50
GM	84.51	75.32	83.29	75.38	74.34	77.89
CSF	77.53	71.56	72.67	77.60	78.93	76.73
WM	92.08	90.01	83.34	89.33	89.35	89.46
Total	95.94	94.60	94.68	95.01	94.93	95.13
Subject With Alzheimer's Disease						
T1 Volume		**Number of Principal Components**				
Tissue	**GA**	**6**	**7**	**8**	**9**	**10**
BKG	99.50	99.52	98.75	99.24	99.31	99.11
GM	80.81	73.33	80.09	74.02	73.30	76.69
CSF	74.59	70.44	72.30	76.28	77.06	75.85
WM	91.54	88.41	81.67	88.90	88.59	87.64
Total	95.31	94.57	93.99	93.70	94.87	93.76

TABLE XXIII

PCA SEGMENTATION ACCURACY FOR THE T2 MRI DATA SETS OF THE SUBJECT WITH HEALTHY STRUCTURES AND THE SUBJECT WITH ALZHEIMER'S DISEASE.

Subject With Healthy Structures							
T2 Volume		**Number of Principal Components**					
Tissue	**GA**	**4**	**5**	**6**	**7**	**8**	**9**
BKG	99.58	99.18	99.27	99.40	99.55	98.95	99.63
GM	58.42	69.52	61.69	66.15	68.00	63.03	69.41
CSF	73.22	59.42	78.42	68.66	68.17	76.37	74.36
WM	78.56	66.88	71.82	75.49	75.10	77.59	70.46
Total	91.72	91.35	92.16	92.35	92.60	92.32	92.87
Subject With Alzheimer's Disease							
T2 Volume		**Number of Principal Components**					
Tissue	**GA**	**4**	**5**	**6**	**7**	**8**	**9**
BKG	99.40	99.16	99.07	98.95	99.12	98.46	99.51
GM	57.00	68.82	61.00	66.26	67.30	62.78	68.22
CSF	53.01	58.68	78.24	67.17	68.11	75.07	73.36
WM	82.44	66.42	70.87	74.40	73.83	77.64	69.69
Total	90.54	90.40	92.19	91.38	92.59	91.03	92.77

TABLE XXIV

PCA SEGMENTATION ACCURACY FOR THE PD MRI DATA SETS OF SUBJECT #1 AND SUBJECT #2 WITH MULTIPLE SCLEROSIS.

Subject #1 With Multiple Sclerosis						
PD Volume		**Number of Principal Components**				
Tissue	**GA**	**6**	**7**	**8**	**9**	**10**
BKG	99.46	99.01	98.92	99.17	99.45	99.40
GM	81.70	77.81	77.33	82.52	80.19	80.68
CSF	80.43	77.78	80.45	77.28	80.30	80.18
WM	84.99	83.96	84.62	83.89	86.46	83.20
Total	95.51	93.60	93.90	94.27	94.65	94.28
Subject #2 With Multiple Sclerosis						
PD Volume		**Number of Principal Components**				
Tissue	**GA**	**6**	**7**	**8**	**9**	**10**
BKG	99.50	98.88	99.46	99.37	99.28	99.48
GM	68.36	61.84	71.90	67.07	70.31	65.94
CSF	81.42	73.38	74.91	68.74	76.37	75.00
WM	83.91	93.74	88.76	89.77	88.54	92.39
Total	95.36	94.50	95.07	94.68	94.92	95.17

H. Erin Rickard
931 Cherokee Rd. #4
Louisville, KY 40204
(502) 456-6572

EDUCATION:

M.Eng. in Electrical Engineering, University of Louisville, December 2001
Concentration: Medical image processing
Thesis: "Feature Selection For Self-Organizing Feature Map Neural Networks With Applications in Medical Image Segmentation"
Advisor: Dr. Aly A. Farag

B.S. in Electrical Engineering, University of Louisville, May 1999
Graduated with Highest Honors

AWARDS & HONORS:

Graduate Fellowship, National Science Foundation (NSF), 1999
Recipient of NSF Research Experience for Undergraduates (REU) grant, 1998
Dean's Scholar, University of Louisville, 1998
Membership in Eta Kappa Nu Electrical Engineering Honor Society
Membership in Golden Key National Honor Society

EXPERIENCE:

Research Assistant, Computer Vision and Image Processing (CVIP) Lab, 1998-2001

- Created a feature selection algorithm for medical image segmentation. Applied a multiobjective optimization genetic algorithm to search among candidate features to find an optimal subset, using a self-organizing feature map as the classifier. Tested the algorithm on a 3-D simulation model of the human brain, as well as several T1 and T2 MRI data sets with promising results.

- Implemented a Shape-From-Shading algorithm in the 3-D reconstruction of the jaw from a sequence of calibrated intra-oral video images.

- Obtained a solid background for medical imaging by taking courses such as Intro to Biomedical Engineering, Digital Image Processing, Computer Vision, Pattern Recognition, Artificial Neural Systems, and Artificial Intelligence, and by attending seminars.